HEC MONTRÉAL
École affiliée à l'Université de Montréal

Forecasting Retail Sales Using Google Trends and Machine Learning

par
Feras Al-Basha

Mémoire présentée en vue de l'obtention du grade de MSc en Global Supply Chain Management

Février 2021

Résumé

Force est de constater que personne ne peut nier l'impact et l'importante significative qu'a démontré le commerce électronique, durant les moments difficiles et sans précédent, telles que celle vécue pendant la pandémie historique de la COVID-19. Bien que la transition vers le commerce électronique ne soit pas encore faisable ni évidente pour les consommateurs, particulièrement dans quelques pays en développement où l'accès au commerce en ligne est limité, dans des économies plus développées, les consommateurs ont su bien profiter du commerce en ligne pour répondre à leurs besoins et à leurs désirs, vu la disponibilité de celui-ci. Cette croissance et cette multiplication impressionnantes des entreprises en ligne ont mené à un changement structurel dans le monde du commerce de détail, notamment en présentant des opportunités et des défis dans la prédiction de la demande, permettant ainsi d'offrir les bons produits, aux bons prix, et une distribution aux bons endroits et dans des délais efficaces.

L'objectif principal de ce mémoire est de proposer une structure méthodologique permettant d'incorporer des données externes, extraites particulièrement de Google Trends, dans l'exercice de prévision de ventes au détail, en tirant parti des techniques modernes de l'apprentissage automatique (machine learning). Afin d'étudier la robustesse de Google Trends dans la prédiction des ventes, nous aurons recours à l'ensemble de données public brésilien du commerce électronique extraites du marché Olist, ainsi que les données de Breakfast at the Frat de l'entreprise de science de données Dunnhumby. Ces données serviront à mener une expérience quantitative dans laquelle nous comparons les performances prédictives sur les prévisions de ventes des modèles suivants : a) le modèle de moyenne mobile autorégressif intégré (SARIMA), b) l'outil Facebook Prophet (FBProphet), c) l'algorithme Extreme Gradient Boosting (XGBoost), et d) l'architecture de réseau neuronal artificiel La mémoire à courte durée (Long Short-Term Memory)(LSTM). La performance de ces modèles de prévision est comparée à un modèle naïf. Le code source de l'expérience est mis à la disposition du public et pourrait être adapté dans de futurs projets. Par ailleurs, à l'aide d'un simulateur, nous évaluons les implications de la performance de la gestion des stocks des erreurs de prévision employées

dans le processus de gestion des stocks. Les résultats suggèrent qu'il n'y a pas de différences statistiquement significatives entre les prédictions faites par un modèle qui utilise uniquement un "dataset" de données réelles, et un modèle qui utilise des données réelles ainsi que les données fictives de Google Trends. Néanmoins, nous constatons que les prévisions sont plus précises lorsque les données réel et celles de Google Trends sont combinées pour prédire la demande de certains produits de vente au détail disponibles dans les données réel. Ainsi, afin de mesurer l'exactitude des prévisions, diverses mesures de performance ont été utilisées. Finalement, les résultats impliquent que les modèles qui produisent des prévisions de demande plus proches de la moyenne et avec une erreur de prévision plus faible, ont un impact positif sur les performances d'inventaire.

Le domaine voit un développement continue dans la recherche de nouveaux algorithmes de prévision basés sur l'apprentissage automatique. Par conséquent, les études comparatives permettent de comprendre les progrès réalisés avec les nouvelles approches par rapport aux précédentes. De plus, elles servent à tester les anciennes approches qui ont réussi à démontrer des expériences précédentes, et à les appliquer sur des nouvelles données et des nouveaux scénarios.

Mots clés : Prévision des séries temporelles, Analyse de la chaîne logistique, Gestion des stocks, Machine Learning, Google Trends, Vent au détail.

Méthodes de recherche : Expérience comparative quantitative, Validation croisée des séries temporelles, Simulation.

Abstract

The historical Covid-19 pandemic has demonstrated the impact and essential significance that e-commerce has on the life of individuals during unprecedented times. Although, not all consumers are able to easily transition to e-commerce shopping due to multiple reasons, in particular in developing economies, many shoppers in advanced economies have relied on digital purchases for their needs and desires. Hence, the significant growth in online business has led to a structural change in the retail industry, presenting novel challenges and opportunities in demand forecasting to provide the right product, at the right place, in the right time, for the right price. The primary objective of this thesis is to propose a methodological framework to incorporate external data, in particular from Google Trends, in retail sales forecasting by leveraging modern machine learning techniques.

In order to investigate the predictive power of Google Trends we use the Brazilian e-commerce as well as the Breakfast at the Frat public datasets, to conduct a quantitative experiment in which we compare the predictive performance on sales forecasts of the following models: a) the Seasonal Autoregressive Integrated Moving Average (SARIMA) model, b) The Facebook Prophet tool (FBProphet), c) The Extreme Gradient Boosting algorithm (XGBoost), and d) a recurrent neural network with long short-term memory (LSTM). To measure forecasting accuracy, various performance metrics are used, and the performance of all forecasting models is benchmarked against a naïve model. The source code of the experiment is made available to the public and can be adapted in future projects. In addition, performance implications of the forecasting errors in the inventory management process are evaluated using a simulation. Findings suggest that there is no statistically significant difference in the predictions made by a model that uses only real-world data as data input and a model that includes real-world data and Google Trends as data input. Nevertheless, forecasting accuracy improves when real-world data and Google Trends are combined to predict the sales of some retail products available in the real-world data. Generally, results imply that models making sales predictions that are closer to the mean and with lower forecast error, have a positive impact on inventory

performance. The field continues to expand with research on new machine learning driven forecasting algorithms. Therefore, comparative studies provide an understanding of the progress being made with new approaches relative to previous ones and serve in testing out old approaches that succeeded in previous experiments, on new datasets and scenarios.

Keywords: Time-series Forecasting, Supply Chain Analytics, Inventory Management, Machine Learning, Google Trends, Retail.

Research methods: Quantitative Comparative Experiment, Time-Series Cross-Validation, Simulation.

Table of Contents

List of Tables and Figures

Acknowledgements

I would like to express my sincere gratitude to my thesis directors Dr. Yossiri Adulyasak and Dr. Laurent Charlin for their time, outstanding guidance, insightful reviews and valuable advice. Without their commitment to my success this work would not have been possible. I am grateful for and thank HEC Montréal, in particular the department of logistics and operations management for the marvellous opportunity that has contributed to my academic and professional growth. I would like to thank all my friends and colleagues at the workplace for their support and motivation. Lastly, I thank my parents and siblings for their unconditional faith and trust throughout this journey.

Chapter 1

1.1 Introduction

The historical Covid-19 pandemic has demonstrated the impact and essential significance that e-commerce has on everyone's lives during unprecedented times. Governments around the world have imposed mandatory restrictions and enforced temporary shutdown of stores and restaurants to limit the spread of the virus among citizens (BBC News, 2020; The Canadian Press, 2020). In response, digital purchases have spurred, and global retail e-commerce sales are projected to reach 6.54 billion USD by 2022, an increase from 1.33 billion USD in 2014 (eMarketer, 2019). Although not all consumers are able to easily transition to e-commerce shopping due to multiple reasons and in particular in developing economies, many shoppers in advanced economies have relied on digital purchases for their needs and desires (Euromonitor International, 2020). Hence, the significant growth in online business has led to a structural change in the retail industry, presenting novel challenges and opportunities in demand forecasting to provide the right product, at the right place, in the right time, for the right price.

The internet search capability is a key activity that leads to a purchase of goods or services across different retail channels. From November 2018 to November 2019, internet search traffic initiated 65 percent of global e-commerce sessions with 33 percent of the traffic attributed to organic search and 32 percent to paid search (Wolfgang Digital, 2020). Notably, Google dominates the worldwide market share of all search engines since it entered the market in 1997 and now comprises of 86.02 percent of total searches as of April 2020 (StatCounter, 2020). Furthermore, studies have emerged that focus on including internet search data in demand forecasting. For example, Google Trends search index data has been shown to increasingly influence and impact business outcomes across a variety of industries, ranging from predicting private spending, UK cinema admissions, Zika epidemic, tourism influx, and oil consumption (Hand and Judge; 2012; Teng et. al; 2017; Önder, 2017; Woo and Owen, 2018; Yu et al. 2019). This thesis seeks to add to the literature by investigating the predictive power of Google Trends in retail sales forecasting using modern machine learning techniques. The predictive power of Google Trends is

explored by considering general search terms that can be used in forecasting the sales of multiple products across categories and other terms related to the activities of supply chain partners participating in the fulfilment of the end-product. This setup builds on previous studies, to exemplify, Boone et. al (2018) that use Google Trends data in sales forecasts generated at an individual product level for an online speciality food retailer and select search terms that are directly related to the brand name or description of the items sold. Additionally, an inventory management simulation is conducted utilizing the sales predictions generated from the forecasting models, to interpret supply chain cost implications.

Fisher and Raman have been studying data-driven analytical forecasting approaches to retailing since the mid-1990s (Fisher et. al, 2014). In '*The New Science of Retailing*' (2010) they survey retail companies to track their practice in forecasting, supply chain efficiency, inventory planning, and data management (Fisher et. al, 2014, Fisher and Raman, 2018). The authors observed that a majority of retailers treated demand forecasting as a right-brain function that is based off intuition and experience as opposed to a systematic use of data (Fisher and Raman, 2018). However, the sources of available data that can be used in producing forecasts have emerged from point-of-sale (POS) and loyalty cards to in-store video and social media exchanges, among others as well. In the early days of analytical forecasting, effectively analyzing historical sales data has been proven useful in improving decisions. In order to make more informed decisions and demand forecasts, retailers are looking into opportunities to mix art with science by leveraging novel techniques available through the increasing influx of data in terms of volume, variety, velocity and veracity (Hofmann and Rutschmann, 2018). Global spending by retailers on Artificial Intelligence (AI) services is expected to reach 12 billion USD by 2023, which is an increase from an estimated 3.6 billion USD in 2019, with expert demand forecasting systems taking a considerable share of the investment ("Juniper research highlights," 2019). Prominently, time-series forecasting has previously received significant attention from academia and the industries and will continue to be important in the future.

This thesis intends to add to the time-series forecasting body of knowledge by conducting a comparative quantitative experiment that is applied on the Brazilian E-commerce Public Dataset by Olist[1] and the Breakfast at the Frat[2] public dataset by dunnhumby in order to explore the predictive power of Google Trends in forecasting retail sales. The field continues to expand with research on new machine learning driven forecasting algorithms and therefore comparative studies provide an understanding on the progress being made with new approaches relative to older methods. In addition, comparative studies serve in testing out methods and approaches that succeeded in previous experiments on new datasets and scenarios.

1.2 Contribution

The contribution of this thesis are as follows:

- A methodological framework to incorporate external data in retail sales forecasting and in particular Google Trends by leveraging modern machine learning techniques.

- Empirical Comparison: Using the Brazilian E-commerce Public Dataset by Olist and the Breakfast at the Frat dataset by dunnhumby to compare the predictive performance on sales forecasts of the following models: a) Seasonal Autoregressive Integrated Moving Average (SARIMA), b) Facebook Prophet (FBProphet), c) Extreme Gradient Boosting (XGBoost), and d) a recurrent neural network with long short-term memory (LSTM).

- Inventory Management Simulation: the forecasting results of the different models considered are used to simulate a periodic inventory control policy.

- The source code[3] of the experiment is made available to the public and can be adapted in future projects.

[1] https://www.kaggle.com/olistbr/brazilian-ecommerce
[2] https://www.dunnhumby.com/source-files/
[3] https://github.com/FerasBasha/Forecasting-Retail-Sales-Using-Google-Trends-and-Machine-Learning

1.3 Outline

This thesis has 5 parts. In Chapter two we conduct a literature survey and a review of related work. Chapter three will elaborate on the data collection and integration process which is followed by Chapter four where the methodology and experiment design are addressed. In chapter five, the results of the experiment are presented and in Chapter 6 the limitations, conclusion and an outlook for future endeavors is shared.

Chapter 2

2.1 Literature Review

Typical retail supply chains involve the collaboration and coordination of manufacturers, wholesalers, warehouses, distribution centres, physical and virtual stores as part of the end-to-end process that makes the final product available to end-consumers. Multiple studies have loomed investigating supply chain management strategy and organizational performance, leading to the development of the Supply Chain Operations Reference (SCOR) model endorsed by the Supply Chain Council and APICS (Huan et. al, 2004). The SCOR model provides a unified framework in supply-chain management practices and processes that result in top organizational performance (Lockamy III and McCormack, 2004). Supply chain decisions in the SCOR model are broadly grouped in to four key decision areas referred to as plan, source, make and deliver. According to Souza (2014) and as presented in Table A1 (see Appendix), demand forecasting is an activity that influences all SCOR decision areas when planning is considered for strategic, tactical, and operational purposes.

To illustrate supply chain decisions at an operational level, consider the sales forecast made by a clothing brand in anticipation of next month's store sales. The forecasted store sales are translated into a demand forecast at the distributor level and a bill of materials at the manufacturer level. The forecasts are then used in planning and the execution of activities related to production planning, inventory management, pricing, and transportation management among others. To extend the example, a strategic decision facing the clothing brand would be where to locate a new production facility. In this case, aggregate sales forecasts, and growth trajectory for the next three to five years may serve as in input in the facility location decision. A review of research on forecasting retail demand from a strategic, tactical, and operational level is provided by Fildes, Ma and Kolassa (2019). The focus of this thesis is on operational supply chain decisions a retailer faces and in particular the activities of sales forecasting and inventory management. Emerging machine learning algorithms used in retail sales forecasting are explored and compared with time-series methods.

Machine learning is a science that is rooted in the fields of statistics, data mining, computer science, engineering and other disciples that study the modeling of data for predictive and inference purposes (Hastie, Tibshirani and Friedman, 2009). According to Mitchell (1997) a machine learning algorithm learns from experience *E* with respect to a task *T* as the experience *E* improves in performance that is measured with *P* (Goodfellow et. al, 2016). To illustrate, consider the task of predicting the sales price of second-hand cars sold. Features of second-hand cars believed to influence the sale price such as kilometers driven, vehicle condition and manufacturer among others are modelled using a function to learn the relationship between inputs and outputs. Mathematically, this can be represented with a model $y_i = f(x_i, W)$ where $y_i \in \mathbb{R}$ is the price of car *i* sold, $x_i \in \mathbb{R}^n$ is the feature input vector corresponding to car i and $W \in \mathbb{R}^n$ is a vector of parameters. Accordingly, function *f* in this example can take the form of a regression algorithm. Depending on how the model is fed input data, which may include inputs with labeled outputs $\{(x_i, y_i)\}_{i=1}^{n}$ or inputs without labels on the output $\{x_i\}_{i=1}^{n}$, machine learning tasks may be broadly categorized into supervised and unsupervised learning. In the example of a regression algorithm utilized to predict the price of second-hand cars, the loss of the model can be measured with the squared error between model predictions and true values. Hence, the set of weights or parameters of the selected machine learning model that yield the lowest loss are optimized by a machine learning algorithm.

Under supervised learning, information on the outcome variable guides the learning process leading towards the generation of future predictions. On the other hand, in unsupervised learning tasks, input data does not contain labeled outputs (Goodfellow et. al, 2016). For example, a clustering algorithm can be used to segment customers based on purchasing behaviour. Correspondingly, in unsupervised learning, data is utilized with the intent to infer the underlying structure and distribution of the "input" data (Murphy, 2012; Goodfellow et. al, 2016). The focus of this thesis is on supervised learning algorithms used in retail sales forecasting and in the following sections a survey of emerging methods

used in previous studies is provided. First, an overview of common time-series methods, as well asl machine learning driven techniques used in retail forecasting are provided. Thereafter, approaches previously used to integrate Google Trends data in forecasts and predictions tasks are examined. Lastly, forecast errors, common performance metrics used and the implication that sales forecasting accuracy has on inventory management is inspected.

2.1.1 Time-series Forecasting

Time-series is obtained by collecting data over time where typically past observations are correlated to succeeding ones and the sequence of data points carries importance. Mathematics is used to model the behaviour of the time-series data. Time-series models focus on identifying changes in patterns over time within the data as well as the examination of association relationships between the dependant and independent variable (Winters, 1960; Park et al. 1991; Holt, 2004; Chu and Zhang, 2003; Thomassey, 2013; Ljung et al. 2014). In a world with no uncertainty, a model that captures all the characteristics of time-series data and makes accurate predictions on the future without error is referred to as a deterministic model (Box, Jenkins and Reinsel, 2008). In practice, retail sales contain a high degree of uncertainty due to multiple factors ranging from changes in weather, economic conditions, and consumer preferences in addition to the occurrence of promotions and calendar events (Thomassey, 2013). Accordingly, stochastic models are used to account for uncertainty in the real-world by calculating the probability of future values within defined limits (Box, Jenkins and Reinsel, 2008).

The most common time-series methods used in retail demand forecasting are: moving average, weighted average, exponential smoothening, ARIMA and SARIMA (Alon et. al, 2001; Souza, 2014; Fildes, Ma and Kolassa 2019). Additionally, common time-series models can be implemented with spreadsheet software used by many organizations like Microsoft Excel, making the usability of time-series methods an attractive factor in practice. De Gooijer and Hyndman (2006) review publications on time-series forecasting spanning more than two decades of research, beginning in 1982, in journals managed by the International Institute of Forecasters, among others. The authors classify reviewed

papers according to the time-series method used highlighting that each time-series technique may relatively be more suitable in identifying the behavior of historical data depending on the presence of seasonality, trend, and cyclical behavior found in a given dataset. A trend occurs as a result of a long-term positive or negative change in the time-series data (Wu et al. 2007), due to for example a change in disposal income that likely leads to more spending on retail shopping month-over-month. Whereas, seasonality is a positive or negative change recurring on a periodic basis as time progresses (Liu et. al, 2016). A basic example of seasonality is the tendency of ice-cream sales increasing on hot summer days or when more foot traffic is observed on weekends in the case of a supermarket.

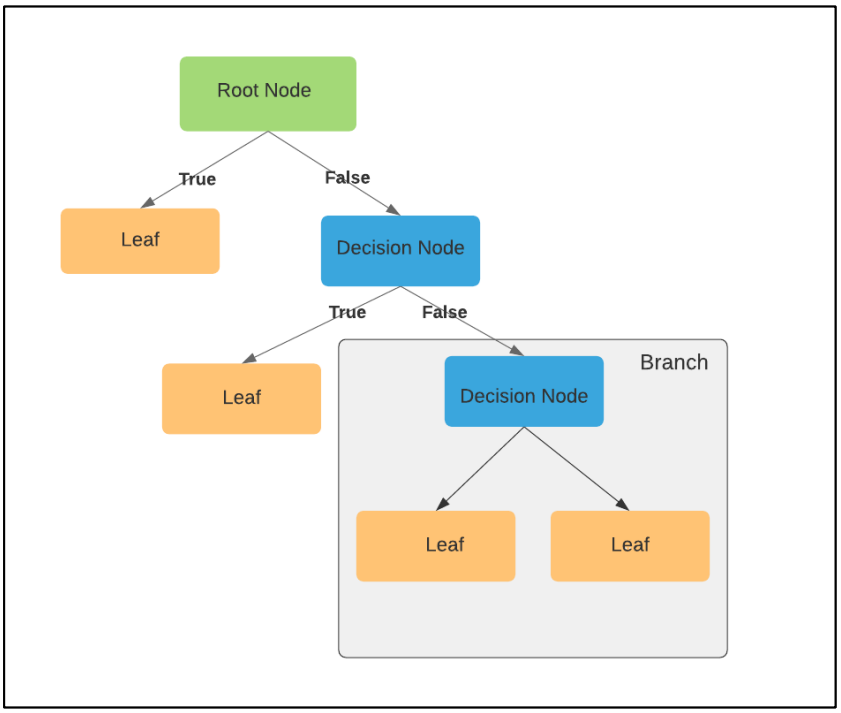
The works of Hyndman et al. (2002) and Taylor (2003) propose a taxonomy that helps practitioners in using exponential smoothing methods according to the patterns of trend and seasonality that can be linear or non-linear. For example, time-series data with no trend or seasonality the simple exponential smoothing method may be applicable whereas Holt-Winters' multiplicative method is more suited for non-linear time-series data with additive trend and multiplicative seasonality (De Gooijer and Hyndman, 2006). However, common time-series models are not as effective in modelling non-linear time-series relationships. To exemplify, basic single and multi linear regression models assume that the dependent variable is a real value and is assumed to follow a normal distribution. Furthermore, despite the flexibility of the ARIMA method in modelling deterministic characteristics of time-series using linear autoregression and stochastic properties using the moving-average process and trend (Gocheva-Ilieva et al. 2019), ARIMA tends to be weak in capturing non-linear time-series dependencies (Hwarng, 2001). Other challenges in fitting common time-series models for retail sales forecasting may be due to the bullwhip effect, the fact that there is limited amount of historical data as new products are launched/listed or the ability to model the impact of price changes of one product on others (Garcia et. al, 2005; Bayraktar et. al, 2008; Ferreira et. al, 2016). Meanwhile, machine learning-based approaches gained popularity in retail forecasting, for multiple reasons including their strength in modeling non-linear time-series relationships (Ahmed, et al. 2010; Cadavid, Lamouri and Grabot, 2018; Bandara et al. 2019; Ohrimuk et al. 2020).

2.1.2 Decision Tree Models

Decision trees are used in supervised machine learning tasks and can be applied to classification (categorical target variable) and regression (continuous target variable) prediction problems (Hastie, Tibshirani and Friedman, 2009). Regression trees seek to partition data into regions and fit a model into each space (Hastie, Tibshirani and Friedman, 2009; Krzywinski and Altman, 2017; Fallah and Ahrens, 2018). To demonstrate the process of building a decision tree, suppose that a sports merchandise retailer with a limited advertising budget seeks to utilize a classification algorithm that predicts if an individual will convert into a customer or not. The data science team at the organization decides to train a decision tree using data on existing customers, including descriptive features and a label for each customer record indicating if the customer was converted or not. Once the decision tree is trained, a prediction on weather a new customer will convert or not is made by using a criterion to evaluate the features of new customers. Each feature within the dataset is modeled onto a node on a tree with a leaf representing the possible outcome. Hence, the intuition is to divide the training dataset into smaller datasets based on the features such that each sub dataset falls under one labeled outcome. In this example, assuming that the geographic location of a customer is a predictor for conversion, a split could be done by asking a true or false question like “is the customer located in the Quebec region of Canada.” Correspondingly, starting at the root node the entire dataset is partitioned into subsets by performing splits on the customer region predictor variable.

Figure 1, depicts a basic representation of a decision tree consisting of a root node, decision nodes and leaf nodes.

Figure 1: Decision Tree (adapted from Chauhan, 2020)



Overall, the process of building decision trees can be grouped into three key activities known as splitting, pruning and tree selection. The decision on which feature to use in splitting the dataset is based on mathematical criteria such as entropy (measure of randomness in the processed features) or the information gain resulting from a split (Hastie, Tibshirani and Friedman, 2009). In classification problems the Gini index is used to calculate the probability of a feature classified incorrectly when selected randomly whereas in regression problems, typically the residual sum of squares is utilized to determine the importance of a feature when a split is made (Hastie, Tibshirani and Friedman, 2009). The calculations for splits is made on every attribute and the attribute that has the highest importance measured by the highest value for information gain is placed at the root node. However, a challenge associated with training decision trees is overfitting the data in particular when the dataset is relatively small (Hastie, Tibshirani and Friedman, 2009; Gocheva-Ilieva et al. 2019).

Theoretically, a decision tree could potentially overfit if the depth of the tree is too high. Accordingly, pruning refers to the process of prev, senting overfitting in decision trees by shortening branches into a leaf (Hastie, Tibshirani and Friedman, 2009). Typically, pruning is achieved by dividing the entire training dataset into a subset of training and validation sets. The decision tree is trained on the subset of training data and pruned for the highest accuracy on the validation set. Thereafter, the tree that yields the lowest cross-validated error is the final tree selected for making predictions on new values.

Advantages of using decision trees is their ability to model numerical and categorical data alike, to model non-linear relationships, to determine the importance of predictor variables and to provide means to interpret how the model arrived at its results (Krzywinski and Altman, 2017; Gocheva-Ilieva et al. 2019). Accordingly, decision trees have been used in sales forecasting due to the ease of use and interpretability by users (Gür et al. 2009; Loureiro et. al, 2018). In contrast, small changes in the dataset may yield to different splits and results (Hastie, Tibshirani and Friedman, 2009). In addition, when the independent variables and observations used are not significant, decision trees tend to yield unstable performance (Gocheva-Ilieva et al. 2019).

The classification and regression tree (CART) method is a supervised learning algorithm that can be used in building decision trees for time-series forecasting purposes (Krzywinski and Altman, 2017). The split criteria and determination of the root node in CART regression trees is achieved by least squares to minimize the residual of sum of squares between the actual data and the average calculated in each leaf (Gocheva-Ilieva et al. 2019). In practice, the CART method is implemented by defining control settings like for instance the minimum number of observations to be included in a node and the splitting method, among others. Other methods for building regression trees are the C4.5 tree algorithm (Oujdi, Belbachir and Boufares, 2019) inspired by the ID3 algorithm initially proposed by Quinlan (1986), and the random forest algorithm among others (Castillo et al. 2017).

Moreover, a decision tree-based approach gaining popularity in sales forecasting is XGBoost, developed by Tianqi Chen in 2016 (Chen and Guestrin, 2016). Gradient

boosting is a technique typically used with decision-tree algorithms like CART to build a model that makes predictions based on an ensemble of weak models (for example trees) (Friedman, 2002). Each learner within XGBoost is represented by a decision tree and the final model is the sum of all trees. Gradient boosting of trees is an approach that is not new however what makes XGBoost attractive is the computational efficiency achieved by boosting trees in parallel as opposed to sequentially (Friedman, 2001; Chen and Guestrin, 2016). This then allows to build tree-boosting systems at scale. Since the XGBoost model has been made available for public use, it has been the winning algorithm in numerous data-science competitions, highlighting its gain in popularity for regression, classification, and ranking problems (Volkovs et al. 2017; Baraniak 2018; Xia et al. 2020).

Since the proposal of the XGBoost method in 2016, there has been a rise in the application of XGBoost on sales forecasting as reported by Behera and Nain (2019) who conduct a comparative study on Big Mart's sales and found XGBoost to produce better performance relative to the existing models as measured by mean absolute error (MAE) and root mean squared error (RMSE). Krishna, Aich and Hegde (2018) compare machine learning algorithms for retail sales forecasting at a store level. Their results suggest that XGBoost is the best performing algorithm with the lowest RMSE and the highest value for the coefficient of determination, R^2 score. Wu, Patil and Gunaseelan (2018) compare different machine learning algorithms in the prediction of retail sales during Black Friday and report superior performance of XGBoost over other regression and deep learning-based models as measured by the MSE. Further elaboration on how XGBoost is used in a retail sales time-series forecasting context is provided in the methodology section of this thesis.

2.1.3 Deep Learning Models

Deep learning is a branch of machine learning that uses multilayered perceptrons in solving real-world problems (Goodfellow et. al, 2016). The foundations of deep learning are rooted in artificial neural networks (ANNs) that draw inspiration from biological nervous systems. The functional unit of an ANN is the perceptron (Rosenblatt, 1958). Figure 2, depicts a perceptron, where input $x_i \in \mathbb{R}^n$ is fed into the neuron that generates output $y_i \in \mathbb{R}$ based on an activation function *f*. Bias in the modelling process is accounted for with Θ and $w_i \in \mathbb{R}^n$ is the weight assigned to input x_i. The perceptron proposed by Rosenblatt (1958) is typically applied to binary classification problems and could be thought of a mechanism that weighs evidence to generate an output. Incidentally, more versatile learning methods, known as deep learning, are formed when perceptrons are structured in multiple layers utilizing ANN architectures (Goodfellow et. al, 2016).

Figure 2: Perceptron

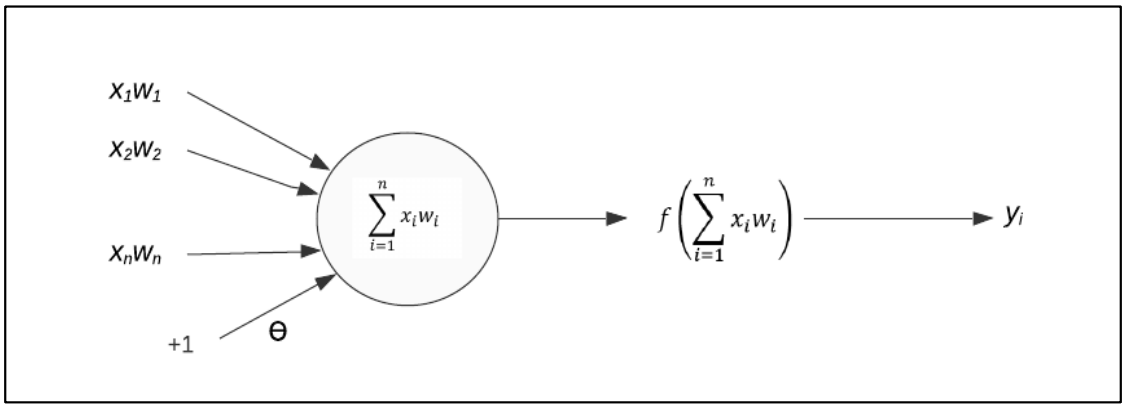
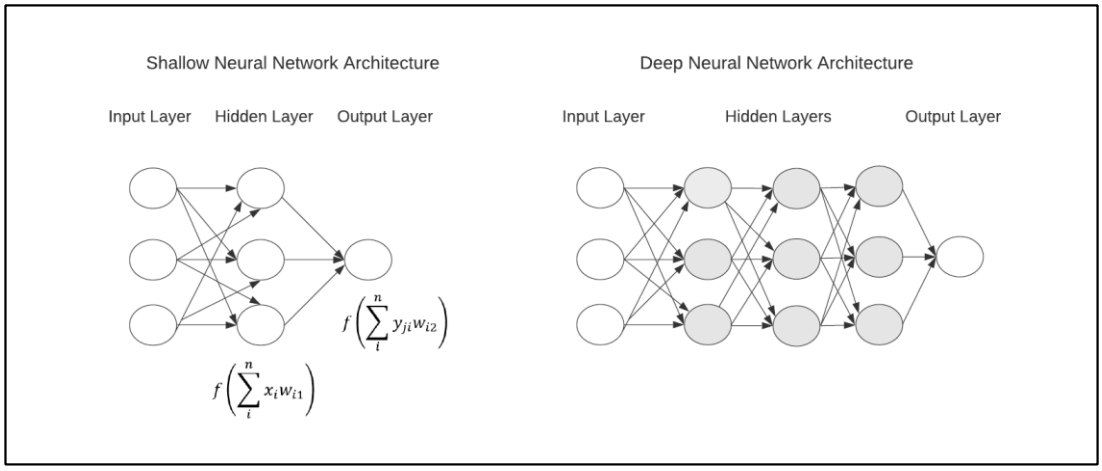


Figure 3, contrasts a shallow ANN architecture that contains an input, a single hidden layer and an output layer to a "deep" neural network architecture that contains more than one hidden layer. The size of the input layer is the number of inputs plus 1, the size of the hidden layer(s) is determined by a hyperparameter and the output layer's size is equivalent to the number of outputs. It can take the form of a scalar or a vector. Each arrow denotes a connection, and each node is the weighted sum of its inputs succeeded by a non-linear activation function as illustrated by the shallow ANN in Figure 3.

Both neural networks shown in Figure 3 are called feedforward neural networks since information flows one way, left to right, with no feedback connections that pass the outputs of the model back into the model itself (Goodfellow et. al, 2016). In training feedforward neural networks, gradient descent optimization stands out as a popular and relatively simple approach in comparison to other methods (Goodfellow et. al, 2016). The gradient descent learning algorithm relies on back-propagation, which refers to the process of updating the weights of a neural network using a loss function, $loss = (actual\ output - predicted\ output)^2$ to reduce the error in predictions and this information is fed back to the network to derive the gradient with respect to parameters *w* (Másson and Wang, 1990; Goodfellow et. al, 2016).

Figure 3: Feedforward Neural Networks

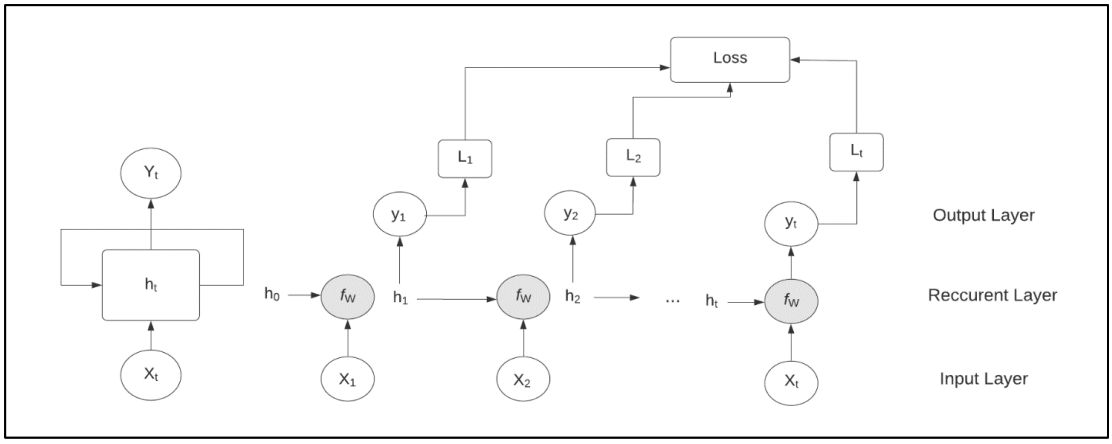


Over the years, neural networks have been extensively applied to retail sales forecasting, commonly using the multi-layered perceptron type of neural network due to its ability in mapping arbitrary inputs and outputs (Zhang et. al, 1998; Alon et. al, 2001; Kourentzes, 2013). Chang, Wang and Liu (2007) evaluate the performance of various neural networks to forecast the sales of printed circuit boards (PCB) at the manufacturer level. The authors report superior performance of the weighted fuzzy neural network, measured in terms of mean absolute percentage error (MAPE), mean absolute deviation (MAD) and root-mean-squared error (RMSE). Fuzzy systems are used to model real-world problems based on human knowledge using linguistic expression (Czabanski, Jezewski and Leski, 2017). A

fuzzy neural network learns the parameters of a fuzzy system using approximation techniques employed in neural networks (Kruse, 2008). Au et al. (2008) explore forecasting apparel sales using evolutionary neural networks on two years of historical data and found superior performance of neural network models over the traditional SARIMA method for products with low demand uncertainty and week seasonality. Evolutionary neural networks are used to overcome the drawbacks of gradient descent based training algorithms such as backpropagation (Yao, 1993). The intuition behind evolutionary neural networks is to model the training process as the evolution of the connection of weights approaching a near optimal set defined by a fitness function (Yao, 1993). Since the selection of the structure, parameters and number of neuros are various aspects that influence the performance of ANNs, evolution algorithms have been successfully used in optimizing the design and the parameters of ANNs (Ding et al., 2013). Sahin, Kizilaslan, and Demirel (2013) used neural networks to forecast demand for spare aviation parts in order to lower inventory costs and stockouts. The findings suggest that neural networks perform best as measured by mean absolute deviation for parts with intermittent demand. Although ANNs are generally known for their ability to identify non-linear patterns, neural network techniques tend to be resource intensive and it is difficult to understand how the model arrived to the predictions (Craven and Shavlik 1997). In addition, when training neural networks overfitting can be avoided by defining a limited number of the epochs hyperparameter (Das and Chaudhury, 2007).

However, ANNs do not factor in the temporal order and sequences of the input data which is an essential aspect of time-series forecasting. To that end, recurrent neural networks (RNNs) which are designed to recognize sequential patterns in data have been observed as suitable for prediction tasks with varying input and output lengths, enabling the network to learn from cross-series information (Hewamalage et al. 2020). Unlike ANNs, RNN architectures address the temporal order and dependencies of sequences in the feedback loop of the recurrent cell. Figure 4, depicts a basic RNN architecture.

Figure 4: Recurrent Neural Network



In Figure 4, input vector $X_t \in \mathbb{R}^n$ is fed into the network and a recurrence relation, $h_t = f_W(h_{t-1}, x_t)$ is applied at every time step *t* to process a sequence and generate the output vector $Y_t \in \mathbb{R}$. At each time step the internal state h_t of the RNN is updated by applying a function parametrized by a set of weights *W* based on the previous internal state h_{t-1} and the input x_t at step *t.* Generally, the same function *f* and set of parameters are used at every time step. Further, an individual loss is calculated at every time step and the model loss is the sum of individual losses. Similar to ANN, the RNN can be trained using back-propagation by computing the gradient of the loss with respect *W*. Hence, the neural network is referred to as recurrent since information is being processed internally from one time step to another prior to generating an output. Popular RNN units that capture sequence in modelling are the Elman cell, LSTM cell and gated recurrent unit (GRU) cell (Elman, 1990; Hochreiter and Schmidhuber, 1997; Cho et. al 2014). Although the sequential data processing mechanism utilized in RNNs is considered a strength over ANNs in time-series forecasting, RNNs contrast with decision tree-based models, such as XGBoost, that perform computations in parallel. The methodology section of the thesis elaborates on the structure of a RNN that contains a LSTM cell and is typically utilized in time-series forecasting.

In retail, methods emerging from RNN driven time-series forecasting have shown promising results. Das and Chaudhury (2007) use a RNN model to forecast weekly sales at a footwear firm by exploring different sizes of lags used to predict the current week's

sales. Results suggest that the RNN's performance helped in reducing costs due to improvements in inventory management. Yu et. al (2018) implement a LSTM network that uses four consecutive weeks to forecast the sales on the fifth week for 66 grocery products. The authors report that only a fourth of the products had low forecasting errors and highlight the absence of promotional data as a key challenge in identifying sales fluctuations. Additionally, the length of the entire dataset used by Yu et. al (2018) is 45 weeks and as a result there is less information on long-term seasonality patterns. Bandara et al. (2019) empirically evaluate a proposed LSTM network for Walmart's E-commerce business in order to model the non-linear demand relationship among product assortment hierarchies. The authors report favorable sales forecast performance on two datasets whereby sales data are aggregated at a product category vs. super-departmental store level. Salinas et al. (2020) develop DeepAR, a forecasting methodology that uses RNNs to learn a global model that uses cross-information from multiple series to generate forecasts. The intuition is to use data on the past behavior of similar, related time-series to make predictions on individual time-series. The approach reduces the time spent on identifying and preparing covariates that are used in traditional single item forecasting techniques and is able to produce forecasts for items that have limited or no historical data available. Salinas et al. (2020) report that the DeepAR method works well on several datasets without the need for extensive hyperparameter tuning.

2.1.4 Hybrid Models

Hybrid models work together to predict an outcome (Tsai and Chen, 2010) whereas ensemble learning methods (ex: XGBoost) work independently of each other and a voting system is used to determine a final prediction. Hybrid models allow the practitioner to utilize the strength of different techniques in forecasting and the performance of hybrid models is therefore believed to achieve desirable results over a standalone technique (Na et al. 2013). To exemplify, a hybrid model could use an unsupervised learning algorithm to cluster data points and pre-process training data. Thereafter, a supervised learning algorithm like a simple classifier can be utilized to learn how to classify new observations. Thomassey, Happiette and Castelain (2005) explore an automatic neural-fuzzy inference system that weighs and quantifies the influence of external variables on weekly sales

apparel sales at a retail distributor. The output of the short-term forecast is then used for adjusted mid-term forecasts. The results suggest that the novel hybrid model outperforms traditional and multiplicative seasonal models. Yesil, Kaya, and Siradag (2012) develop a hybrid algorithm that uses fuzzy logic to combine the forecasts obtained from the moving average, exponential smoothening, and product-lifecycle methods to predict the sales of a Turkish apparel company. Forecasts generated by each forecasting method are combined using a rule based 'fuzzy' system. The 'fuzzy' system is defined with if-then statements. The authors report that the fuzzy logic combiner is adaptive over time as weights are adjusted for better-performing methods and the results suggest that the fuzzy combiner performs better than any of the statistical methods alone. Shouwen et al. (2019) develop a new XGBoost model named as C-A-XGBoost that incorporates features and tendency of the time-series data to predict commodity sales. The model works in two steps by first clustering the data based on selected features and consequently using the features that are most influential for generating forecasts. Thereafter, a XGBoost model is defined for each cluster and hence the term C-XGBoost. Moreover, the A-XGBoost model uses ARIMA for obtaining insights on trends and seasonality of a given series and overcome the shortcomings of ARIMA by leveraging XGBoost for modeling non-linear relationships. The resulting C-A-XGBoost model's predictions are calculated by assigning weights to the forecasts made by the C-XGBoost and A-XGBoost models. Results showcase the dominance of C-A-XGBoost's performance measured in terms of mean error, MSE, RMSE and MAE over ARIMA, XGBoost, C-XGBoost and A-XGBoost (Shouwen et al. 2019).

Hybrid models have been a popular choice utilized in forecasting, as demonstrated with the winning of the M4 forecasting competition (Makridakis et al. 2020). The winning submission of the M4 forecasting competition utilized a hybrid approach that blends the exponential smoothing model with an LSTM network (Smyl, 2020). The proposed method allows to exploit the advantages of statistical and machine learning approaches by first fitting individual series with an exponential smoothing equation and then the model is fit together with global neural network weights using gradient descent. The method is described as hierarchical because the global parameters that apply to all series and local parameters, specific to each series, are utilized throughout the learning process

making use of cross-learning. The following section reviews the common time-series and machine learning approaches that incorporate Google Trends data for forecasting purposes.

2.1.5 Related Work Utilizing Google Trends

The potential value in using Google Trends for forecasting, planning, and marketing activities has been conceptualized by Google with the term *ZMOT - Zero Moment of Truth* (Lecinski, 2014). The term describes a revolution in the way consumers search for information online and make purchase decisions. Essentially, the moment that captures a decision may well be the moment consumers obtain answers to their questions on Google, reflecting their need or intent to buy. The idea of integrating Google Trends data into forecasting and planning activities could be traced back to the work of Choi and Varian (2012) who demonstrated how to use Google search data to forecast near-term values of economic indicators such as, automobile sales, consumer confidence, unemployment claims and others (Varian and Choi, 2009). The authors report correlation between real-time daily and weekly index of the volume of queries that users enter into Google with economic indicators. Additionally, the study underlines the usability of search queries as a leading indicator for complex purchase scenarios where planning activities occur much in advance of the actual transaction. The findings suggest that Google Trends data improves accuracy of predicting the present and near-term future and as a result cuts the lag of reporting that occurs when agencies release results of economic indicator forecasts ex poste. The work of Choi and Varian (2012) uses a SARIMA model and has significantly contributed to the growing interest for research on 'nowcasting'. Nowcasting can defined as the prediction of the present, near future and near past (Bańbura et al., 2010).

Huang and Penna (2010) construct a US sentiment index using popularity of Google Trends searches and generate more accurate forecasts on consumer spending relative to the indexes used by the Index of Consumer Sentiment from University of Michigan and the Consumer Confidence Index from the Conference Board. Askitas and Zimmermann (2009) investigate the relationship between Germany's unemployment rate and time-series data on select keywords using Google Trends. The results suggest a strong

correlation between the dependant and independent variables despite the bias attributed to the turbulent economic environment post recession characterized with changes in economic policy. Carrière-Swallow and Labbé (2013) use Google Trends automotive index data that captures the interest for car purchases in Chile to build an autoregressive regression model that could outperform benchmarks of in-sample and out-of-sample nowcasts towards automobile sales. Robin (2018) explores the usefulness of Google Trends to improve monthly e-commerce retail forecasts in France over traditional indices like monthly retail trend surveys. The study by Robin (2018) uses a SARIMA model that contains monthly retail trend surveys, a SARIMA model containing Google Trends series and a combined model that contains the weighted average forecast from the individual models. The lasso regression approach is used to determine the most relevant Google Trends series for forecasting. Robin (2018) reports that Google Trends do improve the predictive accuracy of the final model, obtained from combining the single models.

Boone et al. (2018) explore whether Google Trends, could be used to improve the accuracy of sales forecasts for a speciality food online retailer. The premise of the study is based on exploring the relationship between a search for a certain keyword and the purchase decision associated with a given product in an online retail setting. More specifically, the study investigates whether including the search volumes on select key words often used to describe a given product in time-series sales forecasting models would lead to improvements in the accuracy of the forecast at a stock keeping unit (SKU) level. Results suggest that including Google Trends data in the forecasting models yields favorable performance as measured by out-of-sample MAPE. The findings of this study are significant because the results may be the first to demonstrate improvements on out-of-sample errors as opposed to preliminary findings on how Google Trends improves in-sample forecast accuracy as reported by Boone et. al (2015). Nonetheless, Boone et. al (2018) allude to the challenges associated with identifying and selecting keywords that would be relevant in forecasting the sales of an item. In addition, polysemous words are likely present a challenge when using Google Trends. By the same token, terms that are semantically perceived to be unrelated could also play a major role in explaining the behaviour of the dependant variable and hence the process of selecting the keywords to

be used relies on a trial-and-error methodology along with the practitioner's experience. Silva et al. (2019) utilize Google Trends in fashion retail forecasting by comparing parametric and non-parametric forecasting models to evaluate the best model to predict the sales of Burberry using Google Trends. The work of Silva et. al (2019) reports forecast performance improvements when Google Trends data is included in a denoised neural network autoregression model. This may not come as a surprise since Google's annual report on fashion trends tends to be accurate in examining what people are wearing and what's trending (Boone, 2016).

Summarizing the related work that use Google Trends in forecasting, results from numerous studies suggest that the use of internet search information improves forecast accuracy and helps better decision making across a variety of industries including retail and e-commerce. The models used in previous works commonly employ time-series techniques like SARIMA and more recently machine learning approaches like neural networks for regression. The choice and predictive power of Google Trends search terms may vary significantly when trying to predict a US sentiment index as done by Huang and Penna (2010) vs. a micro level prediction task like in the work of Boone et al. (2018) who forecast SKU level sales of an online specialty food retailer. Additional details on the Google Trends data is provided in Section 3.1.2 Google Trends Search Index Data. Additionally, in chapter four, we provide the framework used to collect and integrate Google Trends search data in retail sales forecasting with XGBoost and LSTM-RNN machine learning models.

2.1.6 Forecast Error and Performance Metrics

The impact of forecast errors has on supply chains may vary across organizations and managerial priorities in terms of breadth and depth, affecting planning, capacity allocation and inventory management decisions (Lee and Adam, 1986; Kahn, 2003; Kerkkänen et al. 2009). It is realistically not feasible to aim for the perfect forecast in practice. According to Chopra, Meindl and Kalra (2013) forecast errors may arise due to the lack of experience of the forecaster, planned price changes, promotions, competitor behaviour, political and economic conditions among other external factors that may represent

uncertainty. Accordingly, it is important to be able to report, analyse and diagnose forecasts error (Karchere, 1976). When analysing forecast error, special attention is paid to the presence of systematic error. For example, a model that is consistently under predicting values from the actual value suggests that the model is systematically underestimating and therefore should be corrected (Giacomini and Rossi 2009). By the same token, contingency plans can be planned for by considering the forecasting error (Chopra, Meindl and Kalra, 2013). Lastly, among the various metrics used to measure forecast error, MSE, RMSE, MAD and MAPE tend to stand out as being commonly employed (Chopra, Meindl and Kalra, 2013).

When considering the supply chain from an end-to-end perspective, the variability of orders tends to increase the further a supply chain party is distanced from the end customer and this phenomenon is known as the bullwhip effect (Lee et al. 1997). To exemplify, suppose a supply chain network that produces and sells canned drinks consists of a manufacturer, distributor, and a retail store whereby the end product is made available to end customers. The retail store generates demand forecasts based on market demand i.e., point of sale (POS) data. The distributor then uses the demand signal from the retailer in producing her forecasts and similarly the manufacturer generates forecasts based on the demand signal from the distributor. Therefore, demand information tends be distorted further downstream the supply chain for multiple reasons including order batching due to long lead times for less frequently ordered goods and price fluctuations (Lee et al. 1997). Accordingly, the forecast error is intertwined with inventory management decisions across the supply chain and in the following section, an overview of inventory management policies commonly used in retail is provided.

2.2 Inventory Control and Supply Chain Performance

Inventory management plays a key role in supply chain and logistics activities. On one hand, carrying inventory smoothens operations by satisfying demand using available stock, eliminating the need to create or procure goods from scratch (Viale,1996). On the other hand, excess inventory is associated with multiple costs including storage, damage, and obsolescence (Viale, 1996). Correspondingly, inventory control policies are used in making decisions on how much inventory is needed to buffer against the uncertainty in demand and supply throughout operational cycles. Furthermore, inventory replenishment strategies take into account the nature of demand which may be independent or dependent (Toomey, 2000).

Independent demand occurs where consumption is determined by the market, to exemplify a store where the customer makes a purchase. Often, independent demand inventory is called distribution inventory that consists of a finished goods at a manufacturer warehouse or packaged items at a regional distribution center (Toomey, 2000). Whereas, dependent demand inventory is calculated based on customer orders, translating into the raw materials and components required to make the final goods available (Toomey, 2000). Typically, independent demand is forecasted, providing the input for an inventory management system that determines how much, when and where an item is required based on predictions of future operations. Accordingly, inventory management systems seek an order replenishment policy that minimizes the total cost associated with acquiring, holding, ordering and shortage of inventory (Chopra, Meindl and Kalra, 2013). The methods used in an inventory control system depend on the behaviour of demand. To exemplify, the basic economic order quantity (EOQ) model for inventory management assumes that demand is static and deterministic and therefore for a given planning horizon, an inventory level that satisfies all demand without shortage can be computed (Wee, 2011). In contrast, when demand is assumed to be stochastic, the inventory management method used includes a buffer (safety stock) to compensate for shortage risks during the replenishment period referred to as period of risk (Toomey, 2000; Wee, 2011). According to Coelho, Cordeau and Laporte (2014), demand is defined as stochastic when it is not assumed to be stationary over time. This implies that if a given

time-series were to be divided into N sections, the mean and variance measured for each section would not be equivalent. Furthermore, common inventory management methods such as *(r,Q)* and *(R,S)* that model stochastic demand, assume that demand follows a normal distribution, a property that typically applies for regular and fast-moving retail items (Kapalka, Katircioglu and Puterman, 2009).

Under a continuous review system such as *(r,Q)*, inventory control is performed in real-time using the parameter *r,* for the reorder point and parameter *Q,* for the order size of replenishment (Toomey, 2000). The premise of continuous review is that inventory levels for a given item is being monitored and once the inventory level reaches the reorder point *r,* a fixed quantity of, *Q* is placed (Toomey, 2000). The time between when an order is placed and its arrival is referred to as the lead time, *L.* Hence, the reorder point in a continuous review system is calculated by adding the anticipated demand during the lead time and the safety stock. The safety stock is calculated using statistics on the standard deviation related to the deviation in lead time and service level desired by the firm (Toomey, 2000). The (*r,Q)* inventory control method and reorder point provides an intuitive way of managing and calculating inventory when independent demand has a consistent uniform rate (Roll and Kerbs, 1982; Toomey, 2000). In retail however, counting the inventory level of each SKU may be costly due to the large number of items a supermarket store may contain. Additionally, for many retail items demand may not be continuous and is often lumpy and therefore alternative replenishment strategies have been developed (Baumol and Ide, 1956; Roll and Kerbs, 1982; Ge at al. 2019; Snyder and Shen, 2019).

When independent demand is not flowing at a uniform rate, a time-phased order replenishment policy can be implemented (Toomey, 2000; Chopra, Meindl and Kalra, 2013). Under a periodic review system like *(R,S)* an order is placed every order interval *R*. The order interval *R* is usually pre-determined, and the order quantity corresponds to the difference between the target inventory level *S* and the current inventory level when the order is placed. The target inventory level is usually determined based on anticipated demand, lead time, review period *R* and the safety stock. Although the safety stock tends to be higher when using a periodic review system since the period of risk is extended to

include the review cycle and lead time, among the advantages of the *(R,S)* system is the ability to create a consolidated purchase order for a supplier across multiple items while having the flexibility of ordering small quantities for those slow-moving items (Toomey, 2000). Another inventory control system for stochastic demand is the min-max system (Agin, 1966). The min-max system can be thought of a combination of the *(s,Q)* and *(R,S)* methods since at each time period the inventory level is checked. If the inventory level is found to be below the reorder point, then an order is placed to bring back the inventory level to the maximum target level. If the inventory level is above the reorder point, no order is made until the next inventory review period. Thus, at an organization, the selected inventory control method associated with a continuous or periodic review policy in the case of stochastic demand will vary based on multiple factors including the behaviour of demand for an item and the balancing of the trade-off between service levels and operational cost. Furthermore, comprehensive models on inventory management that take into account the dependencies among real-world multi-echelon supply chains across the stages of procurement, manufacturing and distribution have been proposed by Clark and Scarf (2004) and Dai et. al (2017).

In a retail setting, inventory control policies are often supported by the predictions made from a forecasting model and the inventory control method utilized will vary according to the trade off between supply chain costs and service levels. Based on the anticipated demand, production scheduling, inventory management and transportation activities are planned for. Therefore, forecasting models that are able to generate predictions with less error help in advising more grounded inventory control policy. Over the years, common time-series and novel machine learning driven techniques have emerged to make use of data to generate predictions on the future. The choice of the method used in modelling the data varies on the availability of data as well as the presence of level, trend, or seasonality in data in addition to the experience of the practitioner and other factors. This thesis will compare the forecasting performance of common time-series and emerging machine learning driven models. Thereafter, a simulation is run to interpret the implications that the forecasting accuracy has on inventory management and supply chain performance associated with a periodic inventory management system *(R,S)*.

Chapter 3

3.1 Data Collection and Integration

In this chapter, we describe the real-world and publicly available Brazilian e-commerce and Breakfast at the Frat datasets in addition to the Google Trends data collected for the thesis experiment. Using Python, historical sales patterns are visualized, and the scope of data utilized in the experiment is determined. Google Trends data on search terms assumed to carry predictive importance in forecasting sales for each dataset is collected from Google's website. The considerations and assumptions made in integrating Google Trends search index data in time-series forecasting is discussed. Furthermore, data pre-processing and feature engineering performed on the data is elaborated on.

These real-world datasets differ in terms of products sold and the sales channel. From a sales channel perspective, the Brazilian e-commerce dataset contains only digital sales whereas the Breakfast at the Frat data contains only in-store sales. In what follows, the business background of the real-world data is introduced.

3.1.1 Brazilian E-commerce Public Dataset by Olist

In 2020, Brazil accounted for more than a third of the Latin American e-commerce market share (Statista, 2020). In March 2019, the top two major online retailers in Brazil were B2W Companhia Digital that also owns Americana.com, Shoptime and Submarino followed by Mercado Libre which is the largest online marketplace in Latin America (eMarketer, 2019). As the Brazilian e-commerce market continues to grow, new business models, and platforms in online retailing that help local business sell online have emerged.

Olist was found in 2015 as a marketplace integrator enabling brick and mortar retailers to sell on larger virtual stores and e-commerce platforms (Dalmazo, 2018; Mandl, 2019). Besides managing listings and advertisements on behalf of the independent retailer on larger e-commerce platforms, Olist is also responsible for managing the logistics from sellers to the end-consumer. Once a purchase is made by an end-customer on the Olist store, the seller is notified to fulfill the order or depending on the contractual agreement

Olist manages the fulfillment, working with logistics partners. The customer receives a satisfaction survey upon order delivery or when the estimated due date of the order is reached. Hence, Olist's services are not only front end but also extend to operations through freight price optimization, route management and other logistics performed on behalf of the seller. By 2017, Olist managed the listings of 130,000 products for more than 2,300 independent retailers on large-scale e-commerce marketplaces like B2W Companhia Digital, Walmart, Mercado Libre, Via Varejo, Amazon and others (Sant'Ana, 2017). In 2019, Japan's SoftBank group announced an investment of 46.65 million USD in Olist (Mandl, 2019). Correspondingly, according to the founder and chief executive officer of Olist, Taigo Daliv, the company's plan is to reach 100,000 sellers by 2021 from the 7,000 sellers that were registered in 2019 as apart of an ambitious objective to make it one of the largest virtual stores in Brazil (Sant'Ana, 2017; Mandl, 2019). Figure A1 (see Appendix) provides a snapshot of the services provided by Olist for sellers.

In 2018, Olist released a public dataset on Kaggle, a data science collaboration and competition platform (Olist and Sionek, 2018). The dataset contains 100,000 orders processed by Olist between 2016 and 2018. The dataset is divided into relational tables in the form of CSV files, depicted by the schema in Figure A2 (see Appendix). Each order may have more than one item and each item in an order may be fulfilled by a unique seller. The Olist dataset does not contain any information that reveals the identity of stores, sellers, customers, and products sold as all the text has been anonymized by replacing the names with Game of Thrones characters. Altogether, the Brazilian e-commerce dataset has been made available to inspire explorations in natural language processing, demand forecasting, clustering, and other supply chain optimization problems like delivery performance.

3.1.2 Breakfast at the Frat Public Dataset by dunnhumby

The Breakfast at the Frat dataset contains 156 weeks of grocery store sales and transactional information beginning January 2009 until December 2011 across a sample of stores spread over multiple locations in the U.S. The provider of the dataset dunnhunby, a British research firm specializing in retail analytics has been recognized for its engagement with the American retail chain, Kroger (Rohwedder, 2006). Furthermore, the

Breakfast at the Frat dataset user guide describes the stores using attributes on the appeal and size of a "Kroger" store. Accordingly, the stores are believed to belong to the Kroger retail company and its subsidiaries. Kroger is an American retail chain with 435,000 employees and 2,760 locations generating approximately $122 billion in revenue reported on February, 2020 (Statista, 2019; Bureau van Dijk, 2020; Kroger, 2020). Kroger is the fourth most shopped at grocery store in the U.S (Statista, 2019). In particular, Kroger is known for its manufacturing and processing of food products sold in stores in addition to offerings in jewelry and pharmaceutical items (Bureau van Dijk, 2020). Kroger's customer base is split across income geographies that are more likely to live in urban communities relative to the average U.S consumer (Statista, 2019). Prominently, according to Statista's Global Consumer Survey, 48% of Kroger customers ordered groceries online (Statista, 2019). Naturally, Kroger's direct competitors include Walmart, Target, Costco, Sam's Club and Whole Foods Market, among others.

The dataset contains transactional data on the products sold from various brands grouped by four product categories: mouthwash, pretzels, pizza and cold cereal. Unlike the Brazilian e-commerce dataset that provides the data distributed in multiple files, the Breakfast at the Frat dataset is provided in a single source file, simplifying its usability in experimentation. Figure A3 (see Appendix) depicts the data model in the source file along with a description of the attributes available.

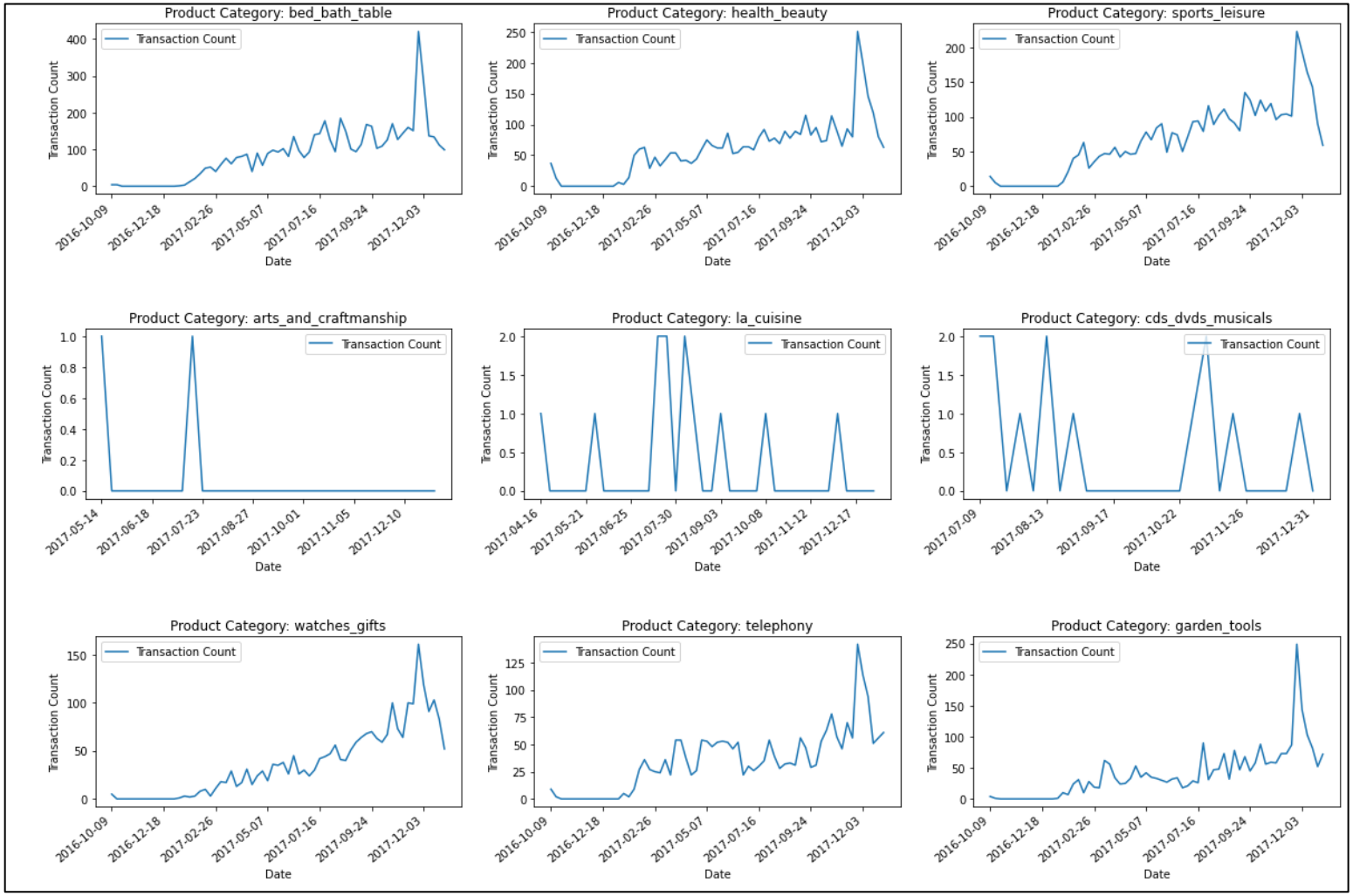
3.1.3 Exploratory Data Analysis

The purpose of the exploratory data analysis is to gain a deeper understanding on the trends and patterns present in historical sales in addition to identifying data quality issues such as missing values and outliers. Additionally, the exploratory data analysis also provides information on data pre-processing requirements that may differ across forecasting models.

Brazilian E-commerce Dataset

The raw transactional dataset includes 99,441 customer orders made on Olist. In 2016, 328 orders were recorded as opposed to 45,101 orders in 2017 and 54,011 orders in 2018. Over 90,000 orders were actually delivered to end customers while few had the status invoiced, shipped, processing, unavailable or cancelled. Approximately 60% of orders contain only one item in it as shown by the diversity of unique items sold by product category in Figure A4 (see Appendix). Moreover, 93,099 customers made only one purchase from Olist out of the 96,096 unique customers that were counted in the dataset. This suggests that historical sales data is diverse across the vast catalog offerings and there is no significant data on repeat customer purchase on specific items within a product category. Figure A5 (see Appendix) shows the concentration of customers across Brazilian states, with Sao Paulo, Rio de Janeiro and Minas Gerais receiving the highest number of unique customer orders. Since the majority of sales occur in Sao Paolo the scope of quantitative experiment is limited to this state. Figure 5 plots, the sales history that is available up to 2018, for 9 product categories. The year of 2018 is excluded from the plot to better understand the training data used in making predictions for the year of 2018. There are a total of 71 product categories, and some do not contain any data or few, intermittent data points. In figure 5, the "arts_and_craftmanship," "la_cuisine" and "cds_dvds_musicals" represent sales of product categories that is sporadic and not practical for use in time-series forecasting

Figure 5: Brazilian E-commerce – Sales History



Over the years, most product categories tend to have a positive trend of increasing sales volume, peaking towards the end of 2017, which corresponds to the holidays shopping season. The positive trend in increasing sales volume for the "bed_bath_table," "health_beauty," and "sports_leisure," may be due to growth in the number of products and sellers' listings under Olist stores in addition to the increased awareness of customers on Olist, since the company was founded in 2015. On the other hand, sales history for the majority of product categories contains random spikes and fluctuations making it more difficult to identify seasonal patterns for forecasting purposes. There are multiple missing values identified in the Brazilian e-commerce data across multiple tables as demonstrated with Figure A6 (see Appendix). The target variable is the payment_value field in the Olist dataset which corresponds to the weekly transaction count and there are multiple payment methods supported by Olist, including credit and gift cards among other local mediums. Accordingly, certain payment methods like credit cards may contain multiple installments. For simplicity, only transactions by credit card with no installments are included in the data used by the forecasting models. Predictions are made for the top selling 7 product categories that are; bed, bath & table, health & beauty, sports & leisure, furniture décor, watches & gifts, telephony, and housewares.

Breakfast at the Frat

The Breakfast at the Frat dataset contains weekly transactional data on grocery items sold across 77 stores in the US. The stores are located in Kentucky, Indiana, Ohio, and Texas. The products sold belong to one of the following categories: bag snacks, oral hygiene, cold cereal and frozen pizza. The earliest record of a transaction is on January 14, 2009 and the last transaction available is on the week of January 4, 2012. Figure 6, shows the number of unique products sold, by category for each manufacturer highlighting the areas of competition among brands.

Figure 6: Breakfast at the Frat –Unique Products By Category and Manufacturer

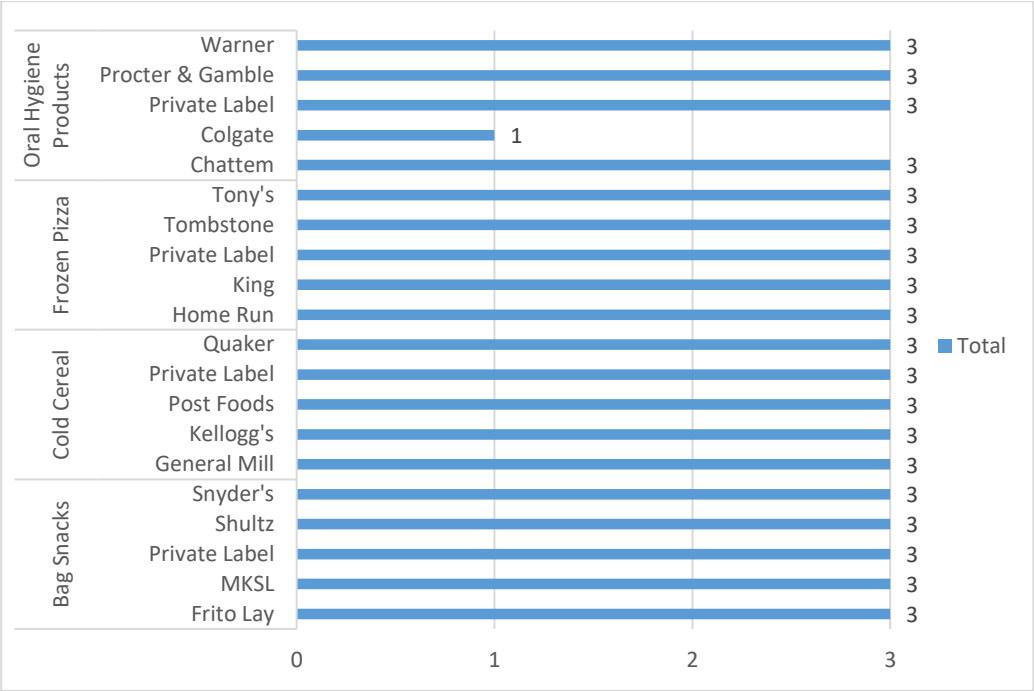


Figure 7, shows the two-year sales history, of a product sold from the bag snacks, cold cereal and oral hygiene categories. Each row represents the sales of a specific item within the category over three distinct stores. Each column is a store. Over the years, the units sold of "frosted flakes" in the cold cereal category, tend to follow a cycle of fluctuations across the stores whereas "spearmint wisp" sales demonstrate intermittent demand behaviour across the stores with low volumes of units sold. The presence of various degrees of trend and seasonality is observed. This may be due to multiple factors including store specific promotions that occur at different points of time and the demographics of the customer base shopping at the store.

Figure 7: Breakfast at the Frat – Sales History by Store

Probing into the attributes of transactional data, Table A2 (see Appendix) presents the merged information available for each weekly transaction of a product sold. Attributes of a store such as "seg_value_name" imply that visitors of a "value" store may be more reactive to promotions as opposed to visitors of an "upscale" store. Further, for a given week, information on the number of unique households that purchase from the store and number of unique purchase baskets that included the product are provided with the "HHS" and "Visits" features, respectively. This information may be helpful to model macro trends at a store level and be used for forecasting purposes. Additionally, for a given week, attributes that capture whether or not a product is on a promotional display and attributes

on price changes are provided. For the purpose of the thesis experiment, the top selling stores measured by the all-time number of items sold from the states of Ohio, Kentucky and Texas are selected. The sales of 4 products representing 3 distinct product categories are forecasted for each of the selected stores. This setup allows to explore the sales of product with different demand behavior across stores, sales of competing brands within a category and influence of promotional changes. Table 1 presents the selected products, totalling 12 possible product-store combinations for which forecasts are made for.

Table 1: Breakfast at the Frat – Selected Products

Product Category	Product Description	Brand
Cold cereal	GM HONEY NUT CHEERIOS	General Mills
	KELL FROSTED FLAKES	Kellogg's
Bag snacks	PL MINI TWIST PRETZELS	Private Label
Frozen Pizza	DIGRN PEPP PIZZA	TombStone (Owned by Nestle)

The following section provides details on the Google Trends data collected and used in predicting sales on the Brazilian e-commerce and Breakfast at the Frat datasets.

3.1.4 Google Trends Search Index Data

Google Trends provides a normalized index between zero and 100 on the volume of queries conducted on a search term by location and category of search (Choi and Varian, 2012). The index is calculated by taking the total query volume for a search term in a given a geographic area divided by the total number of queries for that location at a given period of time, which can be, daily, weekly, or monthly. Accordingly, it is not possible to compare a value of "100" for the search term "Pizza" on Google Trends data in Canada vs. U.S because the absolute search volume in each region varies. Personal information of individuals who have searched on Google is not shared. Google trends has been publicly available since January 1, 2004 and therefore the index for search terms starts with a normalized value of zero as per the data release date. As time progresses, the value of the normalized index for a given search term reflects the deviation in popularity from January 1, 2004.

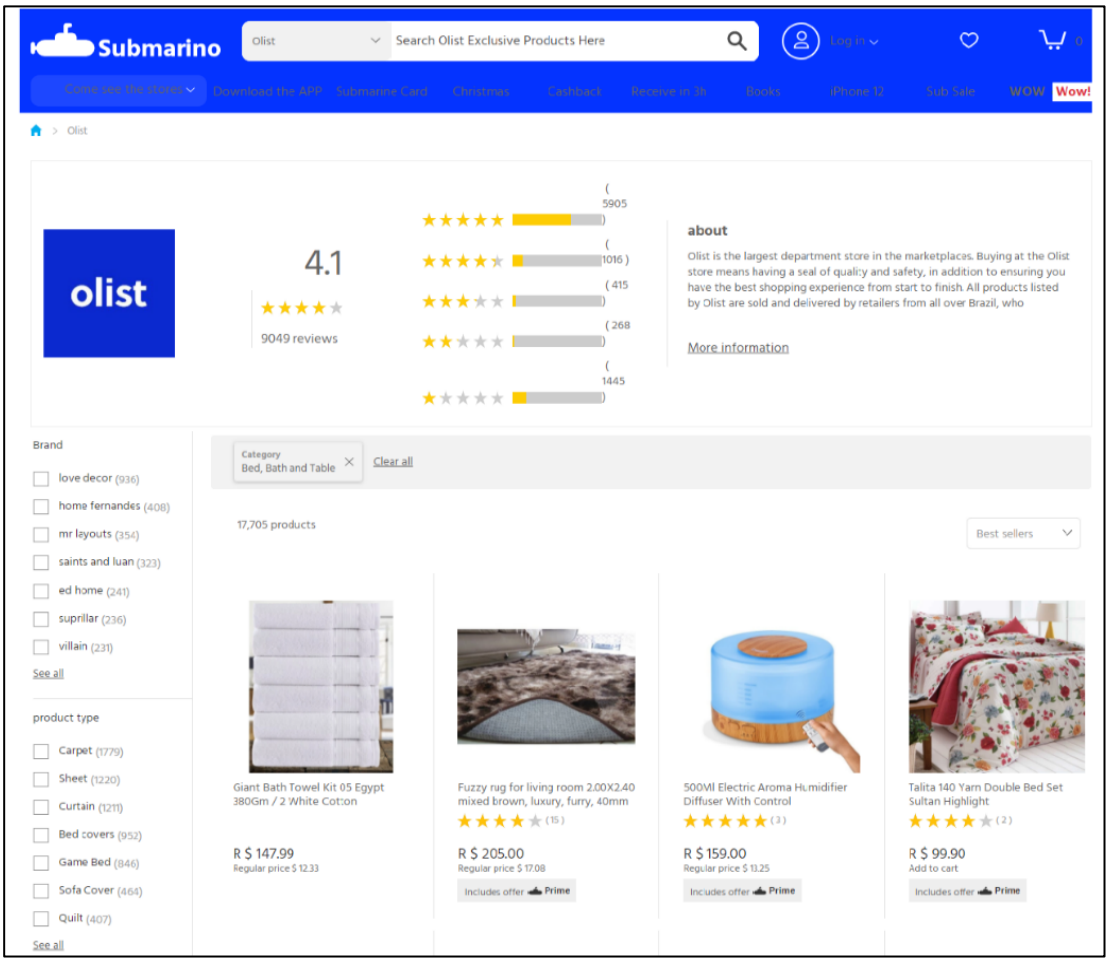
The Google Trends search index is based on an unbiased sample of real-time data that is defined as a random sample of searches from the last seven days and non-real-time data which is a random sample of search data that could go back to the first release date, January, 1st, 2004 or 36 hours prior to the time the search was conducted. Google Trends data does not include statistically insignificant searches made by very few individuals in a given geographic location. Additionality, duplicate searches are excluded from the Google Trends dataset. A duplicate search is when an individual repeats the same search over a short period of time (Google, 2020).

Figure A7 (see Appendix) demonstrates how Google Trends data can be accessed online via a user-friendly interface that includes drill down features, different visualizations, and functionality to export data into a CSV file. Nonetheless, when using the Google Trends interface, daily index series are not returned when a long period of time is selected, typically ranging more than 8 months. In return, a way considered to overcome this challenge is to extract the data in smaller date ranges to be finally merged together after the download. However, it is important that the extracted data is not simply concatenated but rescaled appropriately. To illustrate, in forecasting daily e-commerce products prices using an ARIMA model and Google Trends, Salvatore el al. (2018) used an approach proposed by Velicer and Colby (2005) to process the missing values for each day by taking the average of consecutive days. Besides the user interface that Google Trends provides to download the data, other means of accessing Google Trends is via the PyTrends pseudo-API (General Mills Inc. and Sonnek, 2016). The pseudo-API contains methods to query interest over time for a list of maximum five keywords that can be specified by category and geographical location in addition to calls that return a list of related topics and trending searches related to a search term. Despite the potential advantages that the pseudo-API might yield in terms of bringing in scale to processing Google Trends data, officially, it still remains an unsupported API and therefore for the purpose of this thesis, Google Trends series are collected from Google's official user interface. Other limitations of the pseudo-API is that additional effort is required to configure settings that would avoid running into the timeout error limit.

The intuition behind extracting Google Trends series that carry predictive power in forecasting sales of retail items is based on the brand name, product description and seller information related to the items sold in each dataset. In addition, Google Trends series related to competitor activity and search terms believed to capture consumer preference on trending topics that may influence sales of items across multiple categories are considered. The only example provided of an actual product sold in the Brazilian e-commerce dataset is shown in Figure A8 (see Appendix). In short, all product and seller information is anonymized in the Brazilian e-commerce dataset whereas the Breakfast at the Frat dataset contains the descriptions of the products sold and the respective brands. Due to the absence of information on product specifics and seller information in the Brazilian e-commerce dataset, search terms assumed to influence the sales of items related to a product category are selected by browsing the listings of Olist on partner retail sites such as Americana, Mercado Livre and Submarino. Figure 8, shows Olist's listings for the bed, bath and table product category offered on Submarino. There are thousands of sellers and items sold under Olist's listings for multiple product categories and across online retail platforms. Therefore, Google Trends is collected for sellers and items with high ratings and popularity for each product category, and it is assumed that the trends collected correspond to the anonymized products contained in the Brazilian e-commerce dataset.

Figure 8: Olist Listing – Submarino

(retrieved from[4])



Depending on the availability of data from Google Trends, search terms are collected at a country and country-state level. Assuming that series containing more granular location data are relevant in forecasting sales in a given geography, some search terms included contain series for both country and state level. For instance, when generating predictions for cold cereal items sold at a given store using the Breakfast at the Frat dataset, the "breakfast cereal" search term is collected at a country level and state level since sales

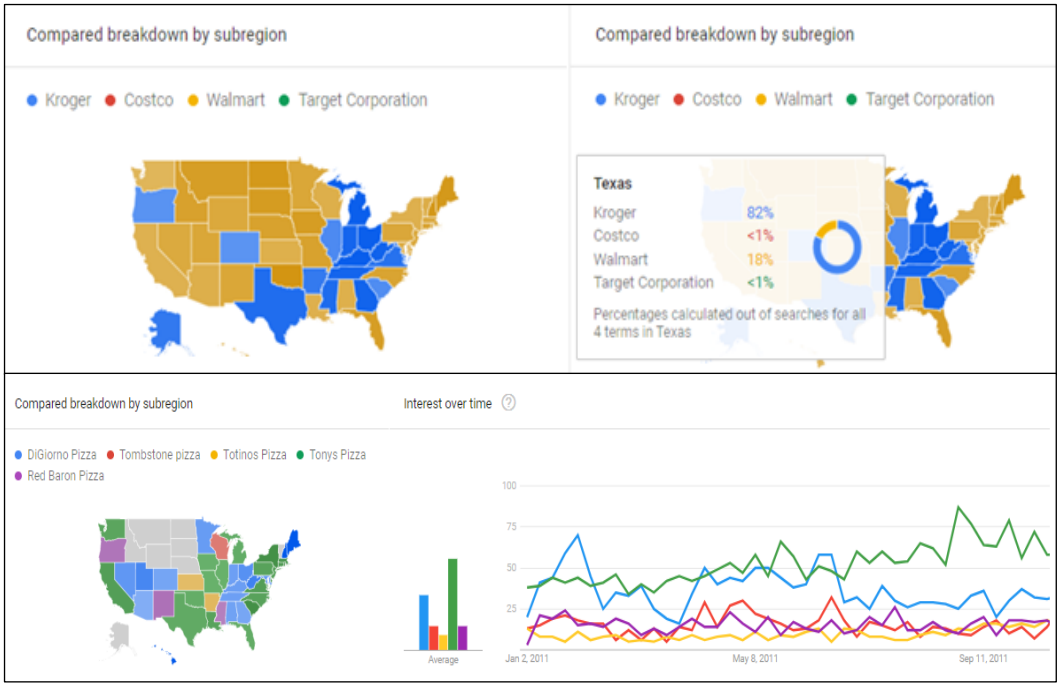
4 https://www.submarino.com.br/lojista/olist?context=lojista&filtro=%5B%7B%22id%22%3A%22categoria%22%2C%22value%22%3A%22Cama%2C%20Mesa%20e%20Banho%22%2C%22fixed%22%3Afalse%7D%2C%7B%22id%22%3A%22variation.sellerID%22%2C%22value%22%3A%2218552346000168%22%2C%22fixed%22%3Atrue%7D%5D&ordenacao=topSelling&origem=nanook&suggestion=true

behaviour of cereal may vary by store location. Table A3 (see Appendix) and Table A4 (see Appendix) present the Google Search Terms used in generating sales predictions for the Brazilian e-commerce and Breakfast at the Frat datasets, respectively.

Special attention is paid to the semantic meanings of search terms when downloading Google Trends series. When using the user-interface, Google Trends provides automatic suggestions that label the semantic meaning behind the search term, as demonstrated in Figure A9 (see Appendix), for the search term Adidas that could be auto-labelled as a raw search term or Adidas as a design company.

The Google Trends user-interface is also used to obtain qualitative information to identify patterns by eye, that may be emerging in retail. Figure 9, shows the compared breakdown of searches made on Google for the major retailers across US states for the year of 2019 in addition to Google Trends on popular frozen pizza brands searched for in the year of 2011. For each state, the percentage of searches for the selected search term is calculated out of all search terms selected in that region. This exercise and the likes help to quickly visualize and gage the interest by region on retailers, brands and product sold and is used in identifying the Google Trends data collected.

Figure 9: Visualizing Retail Patterns on Google Trends, retrieved from[56]



3.1.5 Data Preprocessing

In the aftermath of the data exploration, a set of Python scripts are developed that contain functions to clean, pre-process and address data input formatting requirements that vary across forecasting models used. Data cleaning tasks such as removing duplicates, processing special characters in Google Trends data, and replacing missing values with 'zero' for days without sales are included in the functions of the scripts. In order to ensure consistency of the processing of datasets used by each forecasting model, YAML files and Python scripts are called by each Python notebook to access data and run forecasting models. In particular, the catalog YAML file is the registry of all data sources available for use by the experiment and the parameters YAML file is where all experiment parameters are declared.

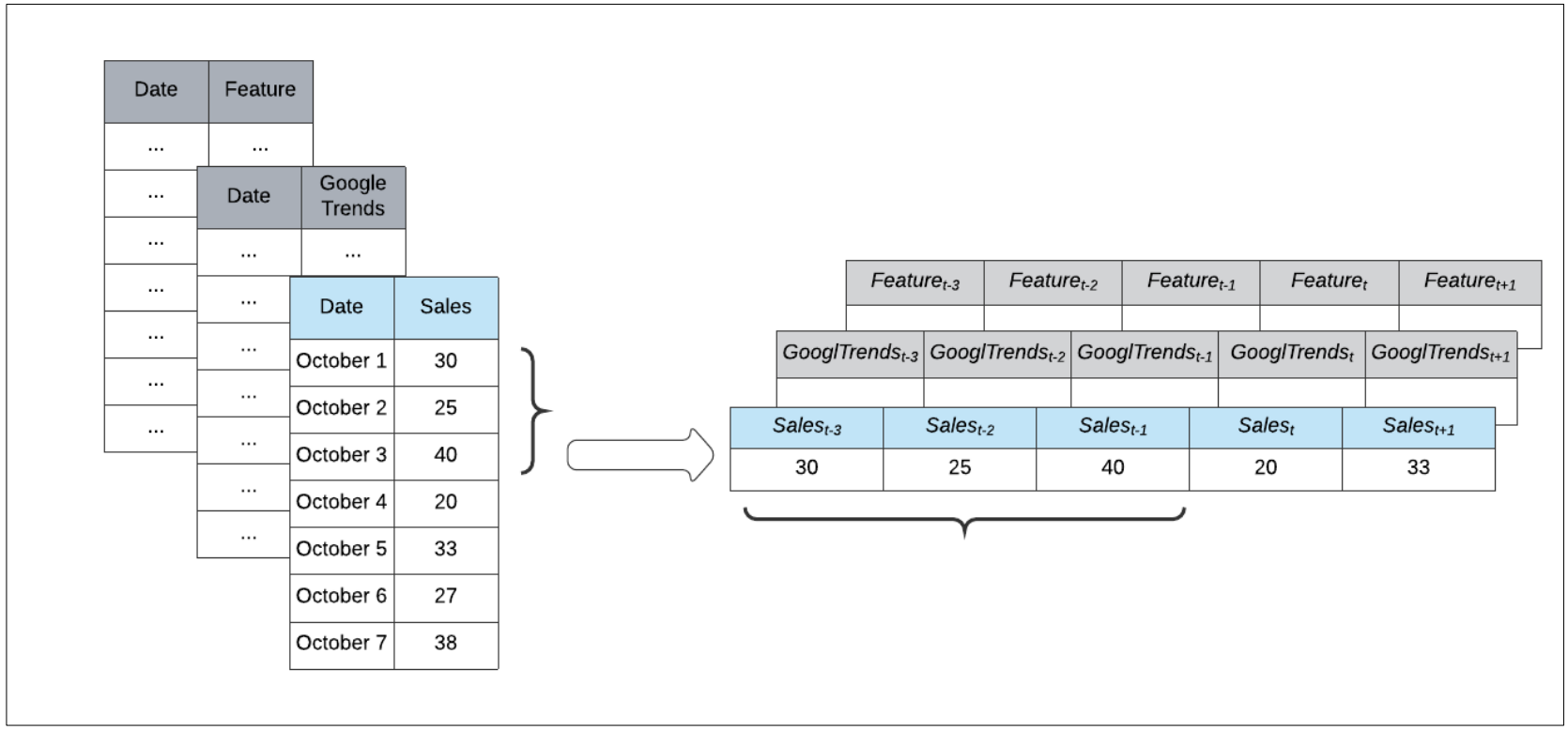
[5] https://trends.google.com/trends/explore?cat=121&date=2019-01-01%202019-12-31&geo=US&q=%2Fm%2F0204w9,%2Fm%2F029f32,%2Fm%2F0841v,%2Fm%2F01b39j

[6] https://trends.google.com/trends/explore?date=2011-01-01%202011-12-31&geo=US&q=DiGiorno%20Pizza,Tombstone%20pizza,Totinos%20Pizza,Tonys%20Pizza,Red%20Baron%20Pizza

3.1.6 Feature Engineering

The process of feature engineering could be divided into tasks related to feature extraction, scaling and selection (Sarkar and Sharma, 2018). When reframing a time-series forecasting problem into a supervised learning task, lag variables are computed to facilitate the machine learning algorithm in learning from its inputs. Whereas, the input for time-series methods used in the experiment like SARIMA and FBProphet use only univariate historical demand data. Figure 10, depicts the lag variable extraction process. Suppose that the past three days are utilized to predict sales on the next day as shown in Figure 10. Lag features, represented with columns, for the target variable, like sales, and for the independent variables such as Google Trends, that correspond in size to the look back period utilized in forecasting the value of the target variable in period $t+1$ are computed. Furthermore, features derived on geometric rolling means of historical sales and Google Trends (two-week, four-week, etc.) are computed to be used as input in the XGBoost model. In addition, for the XGBoost model, features that help the model understand information about time are created using derived date features on time such as year, month, day of the year and day of the week. We use one-hot encoding to represent categorial data, to exemplify, a flag for the weekend day. For the LSTM model, data is scaled and normalized. Specifically, we experiment with two scaling strategies: (1) min-max scaling and (2) the normalizer scaler. The min-max approach, scales data from 0 to 1 while the normalize scaler normalizes the data to have a mean of 0 and standard deviation of 1. The min-max does not change the distribution of the data whereas the normalize scalar centers the data with a mean 0, modifying the data to resemble a Gaussian distribution. The normalized scalar approach is used over min-max to avoid potential challenges in calculating forecasting error where it is not possible to divide by 'zero' since the training data contains 'zero' sales days. Lastly, scaling is not performed column-wise and rather for each set of features. To exemplify, historical sales units for every lag would be scaled at the same time since they are related, while the lags for Google Trends series are scaled separately. This means that for every additional dimension added to the data, a dedicated normalized scalar is used. In the next chapter, the methodology and the experiment setup is discussed.

Figure 10: Lag Variables



Chapter 4

4.1 Methodology

In this chapter we present the methodology of the comparative experiment investigating the predictive power of Google Trends in retail sales forecasting and applied on two real-world datasets. Using the Brazilian e-commerce dataset, the prediction task is to forecast the weekly number of transactions by product category. The scope of sales transactions from the Brazilian e-commerce dataset are limited to the Sao Paolo region and for the top 7 selling product categories. Thus, the Brazilian e-commerce dataset is split into 7 separate datasets and each forecasting model (SARIMA, FBProphet, XGBoost, LSTM) is trained and tested 7 times, once for each product category. In contrast, the prediction task for the Breakfast at the Frat dataset is to forecast the weekly number of units sold of 4 items across 3 stores. Hence, the Breakfast at the Frat dataset is split into 12 separate datasets and each forecasting model (SARIMA, FBProphet, XGBoost, LSTM) is trained and tested 12 times, once for each product and store combination. The data used from the Breakfast at the Frat dataset includes sales history, promotional, product, manufacturer, and store activity information.

The predictive power of Google Trends in forecasting retail sales is examined by comparing the performance of the XGBoost and LSTM models before and after the inclusion of Google Trends as input data to make predictions. Based on the empirical results for the Breakfast at the Frat dataset, the impact of forecasting error on supply chain performance is interpreted by simulating a *(R,S),* inventory control policy with varying lead times. Table 2, summarises the models and the data input used to generate predictions.

Table 2: Models Considered

		Model Family					
		Time-Series		Machine Learning			
Experiment ID	Data Input	SARIMA	FBProphet	XGBoost	LSTM	Performance Comparison	Real-world Dataset(s)
1	Sales History	✓	✓	✓	✓	Compare performance of Time-series models with Machine learning models.	Brazilian e-commerce & Breakfast at the Frat
2	Sales History and Google Trends	—	—	✓	✓	Compare performance of machine learning models that use only sales history (experiment 1) with models that also contain Google Trends (experiment 2).	Brazilian e-commerce & Breakfast at the Frat
3	Sales History and Additional Transactional Data (ex: Store Visits, Household Spend, Promotions etc.)	—	—	✓	✓	Compare performance of machine learning models that contain sales history and additional transactional data (experiment 3) with the models that also include Google Trends (experiment 4).	Breakfast at the Frat
4	Sales History, Additional Transactional Data (ex: Store Visits, Household Spend, Promotions etc.) and Google Trends	—	—	✓	✓		

In experiment 1, we first run time-series methods, SARIMA and FBProphet that use only univariate historic sales data as input to make sales predictions. Similarly, XGBoost and LSTM machine learning models that also use historic sales data as input to generate predictions are run. This setup facilitates the comparison of the performance between machine learning approaches and time-series methods since the data input is the same across model families.

In experiment 2, XGBoost and LSTM are extended to include Google Trends series as features used in making sales predictions. The extent to which Google Trends data improves retail sales forecasts is analysed by comparing the performance of the XGBoost and LSTM models that use only historic sales data as input (experiment 1) with the XGBoost and LSTM models that also include Google Trends as input (experiment 2). Experiments 1 and 2 are applied on the Brazilian e-commerce and Breakfast at the Frat datasets. Using the Breakfast at the Frat dataset we extend the investigation with experiments 3 and 4.

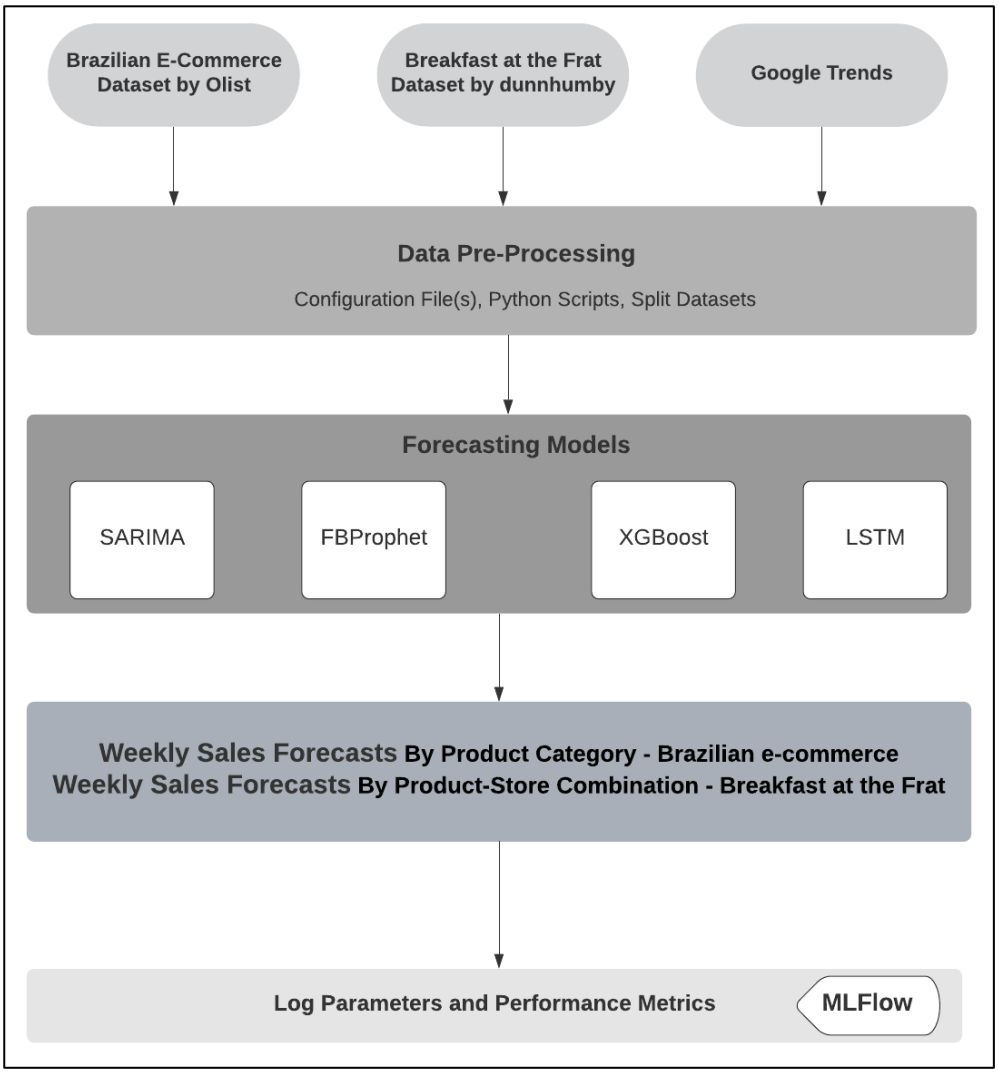
In experiment 3, using XGBoost and LSTM, predictions are made using sales history and additional transactional data capturing information on store visits and promotions among others as data input.

Furthermore, in experiment 4, predictions are made using sales history, additional transactional data, and Google Trends as data input for XGBoost and LSTM. Hence, the comparison of experiments 3 and 4 provide an additional opportunity to observe the changes in forecasting accuracy when real-world data is combined with Google Trends.

Figure 11, depicts a conceptual diagram of the experiment. The MLflow[7] open-source platform, renown for the functionalities it provides to manage machine learning projects in the areas of experimentation, reproducibility among others is used. Specifically, the MLlflow tracking API and UI for logging experiment parameters, model hyperparameters and performance metrics is used to facilitate the traceability and reproducibility of results.

[7] https://mlflow.org/

Figure 11: Experiment Conceptual Diagram



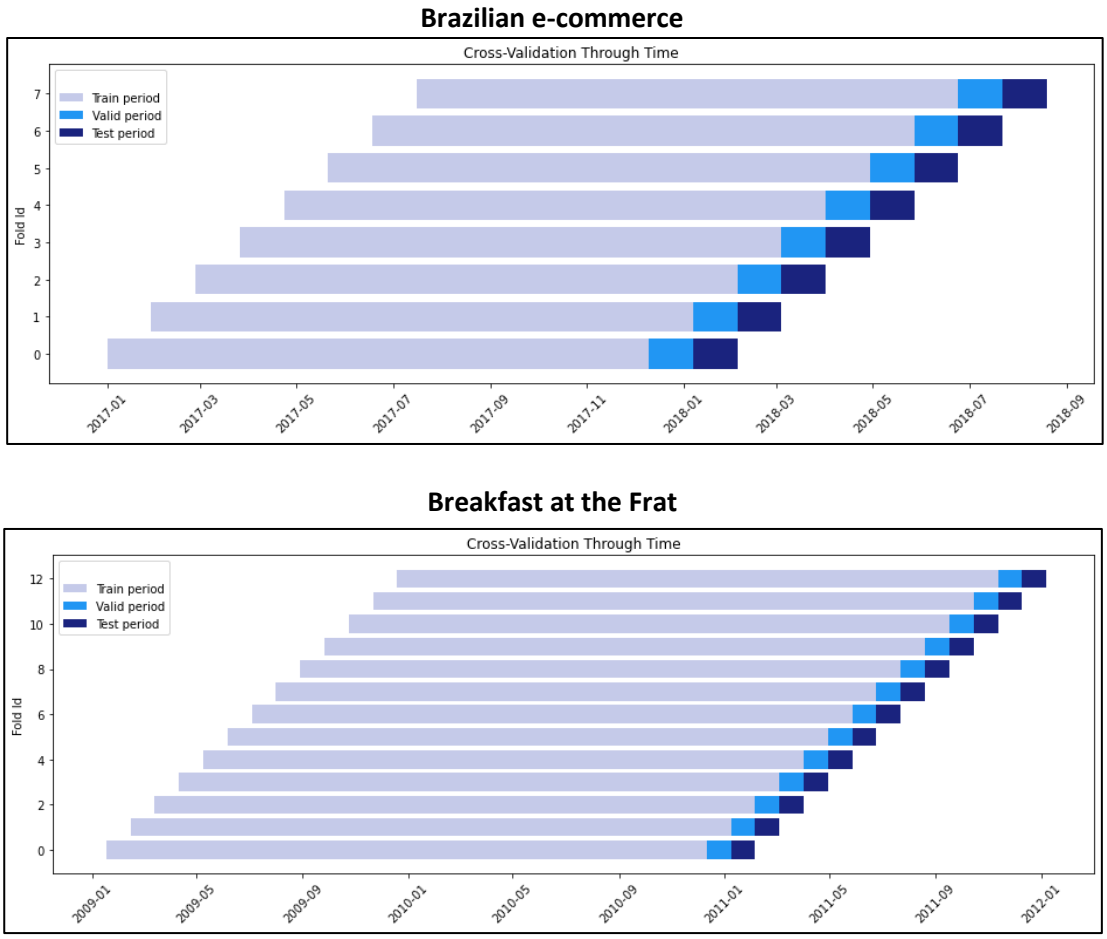
The out of sample (OSS) procedure is a typical approach used to validate the performance of time-series methods, where a section of the time-series that is sequentially at the end of the series, is not used in the training data and only for evaluation purposes (Hyndman and Athanasopoulos, 2014; Bergmeir, Hyndman and Koo, 2018). In contrast, multiple, K, number of evaluations are performed when using K-fold time-series cross-validation procedure and this is often employed for machine learning methods (Hastie, Tibshirani and Friedman, 2009; Bergmeir, Hyndman and Koo, 2018). The K-fold cross validation technique allows to assess the overall generalization of the model. In a retail context, it is important to verify the robustness of forecasts when the model is able to generate

favorable predictions on more than one testing period using the latest information available. Accordingly, to assess model performance, this experiment uses the K-fold cross validation approach adapted for time-series forecasting as it is important to not mix the sequence of data when splitting training dataset.

The start date of the experiment for the Brazilian e-commerce dataset is 2017-01-01 and the end date is 2018-08-12. We omit the data from the year of 2016 when using the Brazilian e-commerce dataset as only few data points exist. The last 32 weeks are used as test data covering inclusively the dates form 2018-01-07 to 2018-08-12. Four-week sequences are used to create the cross-validation folds over the test period. The validation period length is also set to four weeks. When the target variable, the payment_value attribute in the Brazilian e-commerce dataset is aggregated by week, data is obtained for the end of the calendar week. To exemplify, the sales for the week 2018-07-01 include transactions from 2018-06-25 to 2018-07-01 inclusively. When extracting weekly Google Trends series, data is retuned for the beginning of week, that cover the hits from Sunday to Saturday. Accordingly, when generating a prediction for time step $t+1$, Google Trends data only up until the date that corresponds to $t-1$ timestep is used. For example, to forecast the sales using the Brazilian e-commerce dataset for the week of 2018-07-1, the Google Trends series gathered from 2018-07-05, 2018-07-06,… are not used because the time we make the forecast is at 2018-07-01.

Using the Breakfast at the Frat dataset, the start date of the experiment is 2009-01-17 and the end date of the experiment is 2011-12-31. We use the last 52 weeks as our test data, covering inclusively the dates from 2011-01-08 to 2011-12-31. Four-week sequences are also used to create the cross-validation folds over the test period. Figure 12, illustrates the folds for both datasets. For each fold, the training set contains only observations that occur prior to the test set of the fold and excludes future observations.

Figure 12: Time-series Cross-Validation



The performance metrics used to measure the forecasting accuracy of models are; RMSE, R^2, MAPE, WAPE and the paired t-test. The average over all folds for each metric is used to present results. Each metric helps to assess the performance from a different angle. Table 3, defines the mathematical formulation for RMSE, MAPE, WAPE, R^2 and the t-score, where, y_t represents actual sales on time t, $\hat{y}_t$ is the forecast on time t, $\bar{y}$ is the mean of the observed data and n is the number of observations. In addition, $\bar{d}$ represents the mean difference between the paired samples while s represents the standard deviation.

Table 3: Performance Metrics

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}$$
$$R^2 = 1 - \frac{SSresidual}{SStotal}$$ $$SSresidual = \sum_t (y_t - \hat{y}_t)^2$$ $$SStotal = \sum_t (y_t - \bar{y})^2$$
$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left\lvert\frac{y_t - \hat{y}_t}{y_t + 1e-6}\right\rvert$$
$$WAPE = \frac{\sum\lvert y_t - \hat{y}_t\rvert}{\sum y_t}$$
$$t = \frac{\lvert\bar{d}\rvert}{\frac{s}{\sqrt{n}}}$$

RMSE represents the aggregated magnitudes in forecast errors into a single measure for the accuracy of the model and compare the residuals with different models on a given dataset. The RMSE is however sensitive to outliers since the impact of each forecast error in the final calculation of RMSE is proportional to the size of the error. The R^2 score, takes a value between -1 and 1 and is known as the coefficient of determination representing the proportion of variation in the dependant variable that is explained by the independent variable. A negative R^2 implies that the model performed worse than if the mean value of the dataset was used for each prediction made. Hence, a R^2 score that is positive and closer to 1 is favorable.

The MAPE metric provides an easy interpretation of relative error. A challenge associated with MAPE metric is if the observed dataset contains 'zero sales' days since this would imply division by zero in the metric calculation, which is the case in the Brazilian e-commerce dataset. Therefore, to accommodate this limitation, +1e-6 is added to the denominator of the equation. Another drawback of the MAPE metric is that the metric

tends to penalize underestimates by the model more than overestimates (Makridakis, 1993). The WAPE metric measures the average size of error produced by the model, relative to the actual values and hence is more robust to outliers.

The paired t-test is used to test the null hypothesis that there is no statistically significant difference between the means of two populations (Ali and Bhaskar, 2016, Rayat 2018). The null hypothesis and the confidence interval used to test the null hypothesis is provided in Chapter 5.

4.1.1 SARIMA

The seasonal autoregressive integrated moving average (SARIMA) model is a modification of the autoregressive integrated moving average (ARIMA) model (Box, Jenkins and Reinsel 2008). ARIMA is a linear nonstationary model that integrates past values of the target variable using autoregression with previous errors in forecasts made using the moving average method to make predictions on future values. When time-series is nonstationary, the presence of seasonality and trend impacts the value of the time-series at various time points (Coelho, Cordeau and Laporte 2014; Hyndman and Athanasopoulos, 2014). Whereas when a time-series is stationary the properties of the series, namely mean and variance, are independent of the time when the series is observed (Hyndman and Athanasopoulos, 2014). Hence, to apply the ARIMA model on nonstationary timeseries the timeseries is first converted to stationary using a degree of differencing, *d.* Differencing refers to the process of computing the differences between consecutive observations (Hyndman and Athanasopoulos, 2014).

A common notation used to express the model is ARIMA (*p,d,q)*, where *p* is the autoregressive order and *q* is the moving average order. To apply the SARIMA model on a time-series, in addition to converting nonstationary time-series to stationary using differencing of order *d,* seasonal differencing of order *D* is conducted to account for seasonality in the resulting model. The general SARIMA model is expressed as ARIMA(*p,d,q)x*(*P,D,Q)s,* where lowercase notation is for the non-seasonal part of the model, uppercase notation for the seasonal part and *s* represents the number of periods in a year. The general SARIMA model is given by equation (3), where the nonseasonal

autoregressive component of the model is represented with the polynomial φ(B) of order p and the moving average with polynomial $\theta(B)$ of order q (Box, Jenkins and Reinsel 2008; Chang et al. 2012; Hyndman and Athanasopoulos, 2014). The seasonal autoregressive and moving average components of the model are represented with $\phi_P(B^s)$ and $\Theta_Q(B^s)$ with orders P and Q respectively. $\nabla_{y_t}^d$ is the differencing on the nonseasonal part of the series and ∇_s^D is the differencing on the seasonal component. B is the backshift operator used to represent lags in the target variable and ε_t is the error term.

$$(3)\quad \phi_P(B^s)\varphi(B)\,\nabla_s^D\nabla_{y_t}^d = \Theta_Q(B^s)\theta(B)\varepsilon_t$$

The expressions for the remaining components of the general SARIMA model are provided below.

$$(3.1)\quad \varphi(\mathrm{B}) = 1 - \varphi_1\mathrm{B} - \varphi_2\mathrm{B}^2 - \cdots - \varphi_P B^P$$

$$(3.2)\quad \phi_P(B^s) = 1 - \phi_1(B^s) - \phi_2\left(B^{2s}\right) - \cdots - \phi_P\left(B^{Ps}\right)$$

$$(3.3)\quad \theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q$$

$$(3.4)\quad \Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + \cdots + \Theta_Q B^{Qs}$$

$$(3.5)\quad B^k y_t = y_{t-k}$$

$$(3.6)\quad \nabla_{y_t}^d = (1 - B)^d$$

$$(3.7)\quad \nabla_s^D = (1 - B^s)^D$$

To exemplify the computation of y_t using a SARIMA model of the form ARIMA (1,1,1) *x* $(1,1,1)_4$, the model can be written as shown below (Hyndman and Athanasopoulos, 2014). This form of the model facilitates the expansion of the expression to solve for y_t.

$$(1 - \varphi_1\mathrm{B})(1 - \phi_1\mathrm{B}^4)(1 - B)(1 - B^4)y_t = (1 + \theta_1 B) + (1 + \Theta_1 B^4)\varepsilon_t$$

A custom scikit-learn wrapper is developed for training the SARIMA model using the sm.tsa.statespace.SARIMAX[8] package. The Python itertools[9] package is used to conduct a grid search on the parameters that yield the most favorable results. The parameter combination $(p, d, q)x(P, D, Q)s$ with the lowest value Akaike's Information Criterion (AIC) are used in the forecasting models. AIC is utilized to avoid selecting a model that overfits the data (Akaike, 1974).

4.1.2 FBProphet

FBProphet is an additive regression model that contains trend, seasonality, and holiday components. The model is defined using equation (4) shown below, where *g(t)* is the trend, *s(t)* is the seasonality and *h(t)* is the holiday component (Facebook, 2017; Taylor and Letham, 2017). ε_t is the error term representing changes that are not accommodated by the model.

$$(4) \quad y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

The trend component *g(t),* models non-periodic changes in the time-series. FBProphet provides two options for modelling trend (Taylor and Letham, 2017) that includes a non-linear logistic growth shown with equation (4.1) and piecewise linear growth with shown with equation (4.2).

$$(4.1) \quad g(t) = \frac{C}{1 + e^{-k(t-m)}}$$

$$(4.2) \quad g(t) = (k + \boldsymbol{a}(t)^T \delta)t + (m + \boldsymbol{a}(t)^T \gamma)$$

In the logistic trend equation (4.1), *C* is referred to as the carrying capacity of the trend curve representing the curve's maximum value, *k* is defined as the growth rate and *m* is an offset parameter. This type of trend is typically suitable for modelling saturating growth behaviour. One practical example is to think of new members joining Facebook. When Facebook is newly launched in a country for example, one would expect to see higher rates of members joining Facebook initially with lower new membership rates

[8] https://www.statsmodels.org/stable/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html
[9] https://docs.python.org/3/library/itertools.html

observed as time progresses. In the piecewise linear growth trend equation (4.2), growth rate is also denoted with k, δ is the adjustments made to the growth rate, γ is used to model a continuous function and m is the offsetting parameter. Accordingly, vector *a(t)* $\in \{0,1\}^s$ is used to incorporate the trend changes in the growth model by defining changepoints where growth is allowed. *S* represents changepoints at times s_j, where $j = 1, \dots, S$.

The seasonality component s*(t)* of the FBProphet model uses the Fourier series to account for periodic fluctuations as shown in equation (4.3), where *P* is the expected period of seasonality (for example, weekly, monthly, yearly). To approximate seasonality using (4.3), parameters $a_1, \dots a_N$ and $b_1, \dots b_N$ are estimated by constructing a matrix of seasonality vectors for each historic and future values of *t* in the series. The parameter selection can be automated using a similar model selection scheme discussed earlier in the SARIMA model such as AIC. Although increasing N leads to the ability to model fast moving seasonality, this increases the risk of overfitting the data.

$$(4.3)\ \ s(t) = \sum_{n=1}^{N} (a_n cos\left(\frac{2\pi nt}{P}\right) + b_n sin\left(\frac{2\pi nt}{p}\right))$$

The holiday component *h(t),* of the FBProphet model allows the practitioner to make use of pre-determined country specific holiday events such as Thanksgiving in the US or upload a custom list of country specific holidays and events. This allows to incorporate predictable shocks for the business when making future time-series predictions. This is accomplished by using an indicator function *z(t)* that assigns a holiday parameter k_i to time-step *t* if *t* corresponds to holiday *i.* Equation (4.4) represents the holiday indicator function, where *L* is the number of holidays. Assuming that holidays influence extends to the days before or after the actual observance of the holiday, prior days are also used in *h(t)*, equation (4.5).

$$(4.4)\ \ z(t) = [1\ (t \in\ D_1) \dots, 1(t \in \mathrm{D}_L)]$$

$$(4.5)\ \ h(t) = z(t)K$$

The FBProphet model is practical from an implementation perspective as only few lines of code are required to setup and run the model with default parameters assigned. For the experiment, the FBProphet model is implemented using the Python API[10]. FBProphet is built in a way that facilitates modelling the domain knowledge of the user, without requiring a strong background in statistics. To illustrate, the release of a new version of a smartphone implies that the practitioner can model explicitly the corresponding date of this event using changepoints and accordingly factor in this information when forecasting sales of existing smartphones offered by the organization.

4.1.3 XGBoost

The XGBoost model is a variant of tree-based models that uses gradient boosting and ensemble learning in making predictions (Chen and Guestrin, 2016). Ensemble learning is a technique used to combine the power of multiple learners which results in a single model aggregating the output from multiple models. Each learner within XGBoost is represented by a decision tree and the final model is the sum of all trees. CART is the algorithm used in building the decision trees. A basic representation of the general XGBoost model is provided in equation (6), where lowercase k represents a unique decision tree, f_k is the prediction made from the k^{th} tree and uppercase K corresponds to the number of trees in the model (Chen and Guestrin, 2016). Given all the trees, the prediction $\hat{y}_t$ is made as a result of summing up all the predictions made from each tree in the model. The model input is captured with the feature vector x_i for the i^{th} observation. The objective function of the model is defined by a loss function L, such as RMSE in the case of a regression task, shown in equation (7), in addition to a regularization term, Ω, that controls the complexity of the model and prevents overfitting. The regularization term Ω, helps in adjusting the final learnt weights to avoid overfitting, where w is a vector of scores on leaves. Using, γ and λ as parameters that control regularization, the equation for the regularization term is shown in equation (8). The regularization term used in XGBoost is an improvement on other tree-based models that traditionally emphasize learning impurity with less consideration made on the complexity of the model. Hence,

[10] https://facebook.github.io/prophet/docs/quick_start.html

the objective function is the sum of L and Ω. The XGBoost objective function is then optimized using the gradient descent technique.

$$(6)\ \hat{y}_t = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathcal{F}$$

$$(7)\ L = \sqrt{\frac{\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}{n}}\ ,\quad (8)\ \Omega = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

$$(9)\ \text{Objective} = L + \Omega$$

The XGBoost model is implemented using the XGBoost[11] Python library and the hyperparameters are tuned using the Hyperopt[12] library. The Hyperopt library is used in searching spaces to determine the values of parameters that optimize the objective function of a model. Further, the random search, tree of Parzen estimators (TPE) and adaptive TPE algorithms are supported by the Hyperopt library (Bergstra et al. 2013). We utilize the validation set to tune hyperparameters and we retrain the model with the tuned hyperparameters to make predictions on the test set. Hence, the validation sets in the XGBoost model are included as additional training data. The learning task objective used in the XGBoost model is reg:squarederror. XGBoost hyperparameters and the corresponding range of values that they could take is shown in Table A5 (see Appendix). Furthermore, the experiment parameter "window_size" is used to specify the number of lagged values to create for the target variable. Similarly, the experiment parameter "gtrends_window_size" specifies the number of lagged values to create for each Google Trends series used in generating a forecast. To exemplify, a value of 52 specified for the experiment parameter "window_size", will create 52 lags of the target column. Similarly, the "avg_units" experiment parameter is used to create rolling averages using the lag of a feature at time step *t-1,* to avoid leakage. A value of 2 specified for the "avg_units" parameter will create two-time step rolling-average while a value of 16 will create a rolling-average of 16-time steps. Each representing a column in the dataset.

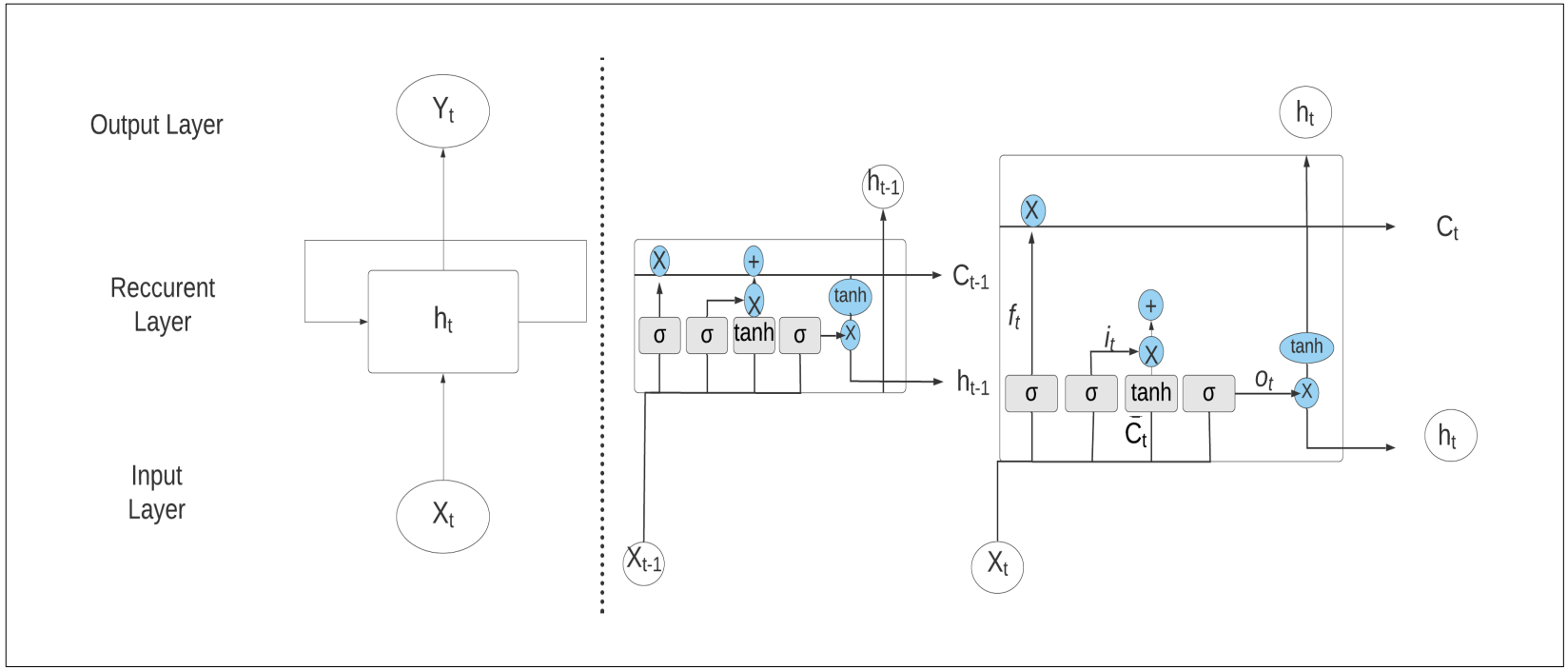
[11] https://xgboost.readthedocs.io/en/latest/python/python_intro.html
[12] http://hyperopt.github.io/hyperopt/

4.1.4 LSTM

LSTM is a type of a RNN that is commonly used for processing long sequences of information to generate output (Goodfellow et. al, 2016). Figure 13, shows the hidden layer of a RNN containing a LSTM cell, initially proposed by Hochreiter and Schmidhuber, (1997). In Figure 13, the left-hand side shows the RNN that contains an input, hidden and an output layer. The right-hand side of the diagram depicts the unfolding of the LSTM cell. Each line within the LSTM cell is a vector of output that is used as an input for another node. The blue circles depict pointwise operations, and the grey boxes are four computational blocks that control information flow. The arrows on a line represent the copy of information from one location to another.

Overall, the LSTM cell performs four operations that forget irrelevant past data points, store the relevant data points and update the hidden state prior to generating an output. The first computation in the LSTM cell is to determine the data points that will be forgotten using a sigmoid function that takes into account previous hidden state h_{t-1} and input x_t (Fischer and Kraus, 2018). If the output from the forget computation given by equation (10) is equal to 1 then the information is kept in the cell state C_{t-1}. Whereas if the forget layer computation using the sigmoid function is equal to zero then the information is omitted from the cell state C_{t-1} . The second computational block in the LSTM cell determines the information to be stored in the cell. This computation is performed in two steps. First a sigmoid function that determines which values to keep as shown in equation (11) is utilized, and second a tanh function is used that outputs a vector of candidate values, $\overline{C_t}$ to be added to the cell, shown with equation (12). Thereafter the new cell state C_t is updated using equation (13). Lastly, the RNN output is generated by using a sigmoid layer that determines which parts of the cell state to output as shown with equations (14) and (15).

Figure 13: LSTM (adapted from Christopher, 2015)



$$(10)\quad f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big)$$

$$(11)\quad i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + \mathrm{b}_i)$$

$$(12)\quad \overline{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + \mathrm{b}_C)$$

$$(13)\quad C_t = f_t \times C_{t-1} + i_t \times \overline{C_t}$$

$$(14)\quad o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + \mathrm{b}_o)$$

$$(15)\quad h_t = o_t \times \tanh(C_t)$$

where σ is a sigmoid function and *b represents an offset term*

When the sequence of data points to be processed are long, the basic RNN suffers from gradient explosion and vanishing of the gradient because of the inability of the RNN to properly adjust the weight parameters. In response, The LSTM cell proposed by Hochreiter and Schmidhuber (1997) overcome the limitations of gradients vanishing or blowing by using a set of computational blocks that enable the RNN to process memory for long term dependencies.

The Tenserflow, Keras[13] open-source library is used to implement the LSTM model. In defining the structure of the LSTM model there are trade-offs considered such as model complexity and run time. In other words, the depth of the model. When running the LSTM model, the "unit_strategy" experiment parameter is utilized to determine how to select the number of hidden units for each LSTM layer. A stable strategy will keep the number of units constant across layers. A decrease strategy will halve the number of units per layer. To exemplify, for a three-layer model with initial number of units set to 50, the stable strategy will assign 50 units for each layer while the decrease strategy will set 50 for the first layer, 25 for the second layer and 16 for the third layer. Accordingly, a function is developed to initialize the LSTM model assigning values to key hyperparameters such as number of hidden layers. The hyperparameters of the LSTM model are determined using random search with an adjustable number of search iterations. LSTM hyperparameters are presented in Table A6 (see Appendix). Similar to the approach taken with the

[13] https://keras.io/

XGBoost model, experiment parameters such as "window_size" and "gtrends_widnow_size" are used to specify the number of lagged values to create for the target column and Google Trends series. Due to the way that the LSTM model is stacked, the number of lag features computed for Google Trends series are required to correspond to the number of lag features computed for the target variable.

To avoid overfitting, we use early stopping on the validation set (Prechelt, 2012). Unlike in the XGBoost model, the validation set is not incorporated as additional training data for the LSTM model since a holdout set is required to perform early stopping. Furthermore, the LSTM model expects the input data to be formatted as a three-dimensional tensor and therefore a Python function is developed to fulfill the requirement. The three dimensions are the number of features, the length of sequence to process and the number of observations (Mussumeci and Coelho, 2020). The Adam optimizer is used for the LSTM models and MAPE is specified as the loss function. According to Kingma and Ba (2014), Adam is a computationally efficient gradient-based optimization algorithm used for stochastic objective functions.

4.1.5 Forecast Based Inventory Management

In retail, sales forecasts serve as key input in inventory management planning and control. Based on the anticipated demand a retailer plans for replenishment, procurement and or manufacturing of products required to be available to meet the demand. Using the empirical results from the Breakfast at the Frat dataset we simulate a *(R,S)* inventory control policy and demonstrate the impact of forecasting error on inventory management performance. The forecasted demand and the RMSE of forecasting models is utilized to determine the target level *S* and safety stock *SS* across *K* evaluation periods (testing sets). The calculations (Axsäter, 2015, Barrow and Kourentzes, 2016) for *SS* and *S* are given in equations (1) and (2) respectively, where; *Z* is the Z score based on a normal distribution, *R* is the pre-determined order interval, *L* is the lead time, *S* is the target level and $\sum_{i=t+1}^{t+R+L} \hat{y}_i$ is the sum of forecasted demand from $t + 1$ to $t + R + L$. The sum of the predicted demand over $R+L$ period is commonly referred to as the period of risk.

$$(1)\ SS = Z\ \times\ RMSE_{forecast} \times \sqrt{R + L}$$

$$(2)\ S = \sum_{i=t+1}^{t+R+L} \hat{y}_i + SS$$

In the simulation, various lead times are used to demonstrate inventory management performance implications across stores for the same items. The pre-determined order interval *R* and the service level that would be set by the organization are assumed. Service level is measured by the percentage of demand fulfilled from stock at hand. The service level corresponds to the probability of not running out of stock when demand arises. In addition, demand not satisfied within a time period is assumed to be lost and no back-orders are considered. Inventory performance is interpreted in terms of the resulting service level, average inventory level, number of orders placed and the inventory turnover ratio.

Chapter 5

In the final chapter of this thesis, we present experiment results investigating the use of Google Trends in forecasting retail sales. The null hypothesis is provided below:

H_0: Predictions generated by models using only real-world data as data input and predictions generated by models using real-world data and Google Trends as data input, are statistically identical and belong to the same statistical distribution.

The null hypothesis is falsified using a paired t-test. The paired t-test is used to compare two population means where the observations in one sample can be paired with observations in the other sample (Kalpić et al 2011). The underlying assumption of the paired t-test is that differences computed in the average performance score of the paired samples belong to a normal distribution with a mean of 0 and unknown standard deviation (Kalpić et al 2011). If there is no statistical difference in the average root mean scaled squared error (RMSSE) between the paired predictions, we reject the null hypothesis. The p-value is used to reject or accept the null hypotheses and indicates the probability of observing the test results under the null hypothesis. Specifically, at the $\alpha = 0.05$ level, if the computed p-value is under 0.05 we reject the null hypothesis. We use the SciPy[14] open-source library to perform the paired t-test.

The paired t-test is similar to before and after observations on a subject. In the comparative experiment, we first run models that generate predictions using only real-world data and then run models that contain Google Trends combined with real-world data as input to make predictions. As discussed in Section 3.1.4 Google Trends Search Index Data, Google Trends is collected using information believed to influence sales from the real-world datasets and assumptions made based on the business background. We use k-fold time-series cross-validation to tune model hyperparameters and test the models on multiple time periods. The models considered are SARIMA, FBProphet, XGBoost and LSTM. SARIMA and FBProphet are referred to as baseline models, using univariate

[14] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html

historical sales as data input to make predictions. Google Trends is not included in the baseline models. Google Trends is incorporated using the XGBoost and LSTM models.

The prediction task for the Brazilian e-commerce dataset is to forecast the weekly number of sales transactions aggregated by product category. The scope of sales transactions from the Brazilian e-commerce dataset are limited to the Sao Paolo region and for the top 7 selling product categories. Hence, for the Brazilian e-commerce dataset, each forecasting model (SARIMA, FBProphet, XGBoost, LSTM) is trained and tested 7 times, once for each product category. The prediction task for Breakfast at the Frat dataset is to forecast the weekly number of units sold for 4 grocery items in 3 stores. Therefore, using the Breakfast at the Frat dataset, each model (SARIMA, FBProphet, XGBoost, LSTM) is trained and tested 12 times, once for each product and store combination. The data used from the Breakfast at the Frat dataset includes sales history, promotional, product, manufacturer, and store information.

The performance metrics used to compare model accuracy are RMSE, R^2, MAPE, WAPE and the paired test. Each performance metric differs in the way error is penalized as discussed in section 4.1 Methodology. In addition, we compare the performance of the models considered in the experiment with predictions from a naïve model and use mean absolute scaled error (MASE) and root mean scaled squared errors (RMSSE) as metrics (Hyndman, 2006). MASE is defined as the mean of $(|q_t|)$ where $q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|y_t - y_{t-1}|}$ represents the scaled error based on the in sample MAE from a naïve model. Accordingly, RMSSE is a related measure to MASE (Hyndman, 2006). A naïve model assumes what happened in the past time step will occur in the next time step. MASE and RMSSE are considered as generally applicable metrics to measure forecast accuracy and contain properties that are favorable over other performance metrics. For instance, MASE is scale agnostic and can be used to compare accuracy with datasets containing different scales (Hyndmann, 2006). In addition, MASE penalizes positive and negative errors equally as well as large and small errors. Following the discussion on results, we inspect the findings and use the experiment forecasts in an inventory management simulation. Furthermore, the performance of the models considered is dependant on the time period. Therefore, the

selected graphs plotting the difference between predictions and actuals are presented for the entire test set.

5.1 Results and Findings

Results and findings are presented separately for each real-world dataset, starting with Brazilian e-commerce, and followed by Breakfast at the Frat.

Brazilian e-commerce

Table 4, presents the performance of the SARIMA, FBProphet, XGBoost and LSTM models as measured by MAPE, RMSE, WAPE and R^2. Results presented are computed by taking the average score of each metric across the 7 product categories considered in scope. Overall, results do not show large differences in performance across the models. Results in Table 4, suggest that the time-series methods (SARIMA and FBProphet) outperform the machine learning models (XGBoost and LSTM) as measured by all metrics. Specifically, FBProphet outperforms all models, as measured by MAPE, RMSE, WAPE and R^2. However, the magnitude of performance improvements on MAPE, RMSE and WAPE as a result of using FBProphet is not large relative to SARIMA. Furthermore, the R^2 of all models is poor and below 0. As discussed in Section 4.1 Methodology, each performance metric considered (MAPE, RMSE, WAPE and R^2) penalizes error differently. Therefore, in Table A7 (see Appendix), we present the performance of models considered using MASE and RMSSE as metrics which are scale independent and errors are penalized in a more symmetric fashion relative to other metrics, to exemplify MAPE (Hyndman, 2006).

Table 4: Brazilian E-commerce Experiment 1 Results

Data Input	Model	Average MAPE	Average RMSE	Average WAPE	Average R^2
Historical sales	SARIMA	0.29	44.46	3417.66	-0.56
	FBProphet	0.25	40.39	3319.33	-0.29
	XGBoost	0.34	51.99	4669.33	-1.15
	LSTM	0.33	53.75	5412.82	-1.17

Results in Table A7 (see Appendix) suggest that for almost all product categories, predictions using SARIMA, FBProphet, XGBoost and LSTM are worse than out of sample forecasts generated by a naïve model as measured by MASE and RMSSE. For the majority of product categories for which predictions are generated using SARIMA, FBProphet, XGBoost and LSTM score a value that is larger than 1 on both MASE and RMSSE. The only occasion where a model obtains a score that is less than 1 for MASE and RMSSE is found for the SARIMA model predicting the sales of the bed, bath and table product category. This implies that training data is difficult to learn. Further, the poor performance by the machine learning models may be partly due to the inability to extrapolate time-series aspects in the training data. A larger size of training data with more years of history may help in improving the machine learning models' performance.

Figure 14, plots predictions using the FBProphet model and the corresponding residuals (forecast error) for the bed, bath and table product category. FBprophet predictions tend to follow the fluctuations in actual sales and generally overpredicts as much as it underpredicts sales as shown with the residual plot. In contrast, Figure 15, shows predictions and residuals using the LSTM model for the bed, bad and table product category. The LSTM model tends to make predictions that are closer to the mean value of actuals. However, the residual plot illustrates that predictions using LSTM are generally under the actuals (frequent positive residuals).

Figure 14: Brazilian E-commerce Predictions – FBProphet

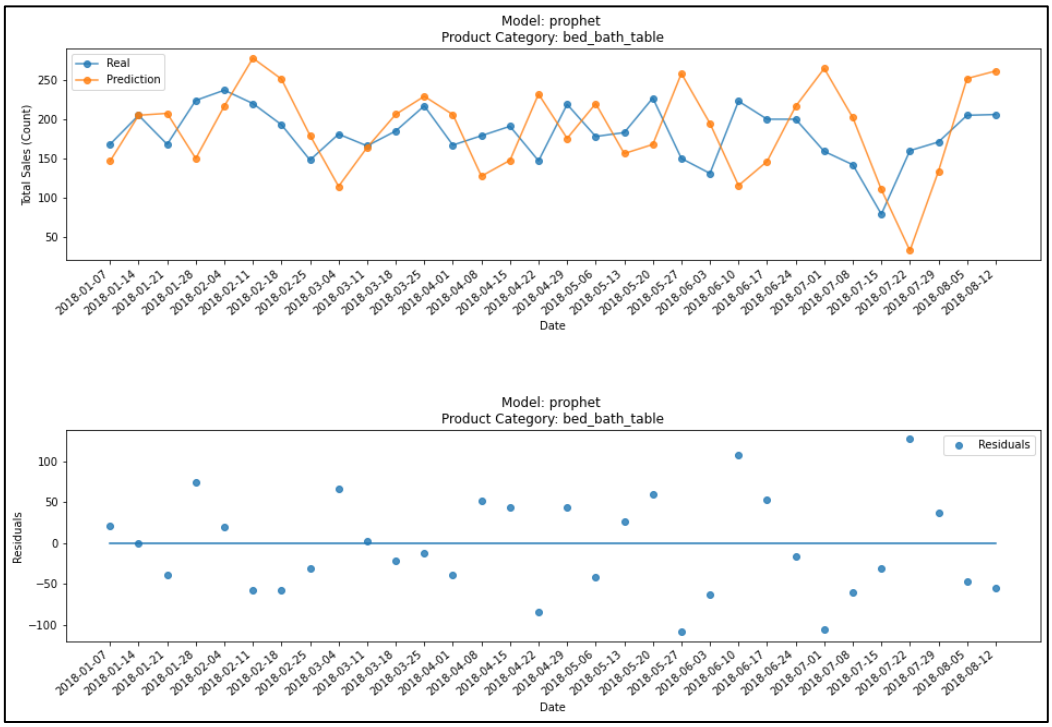


Figure 15: Brazilian E-commerce Predictions – LSTM

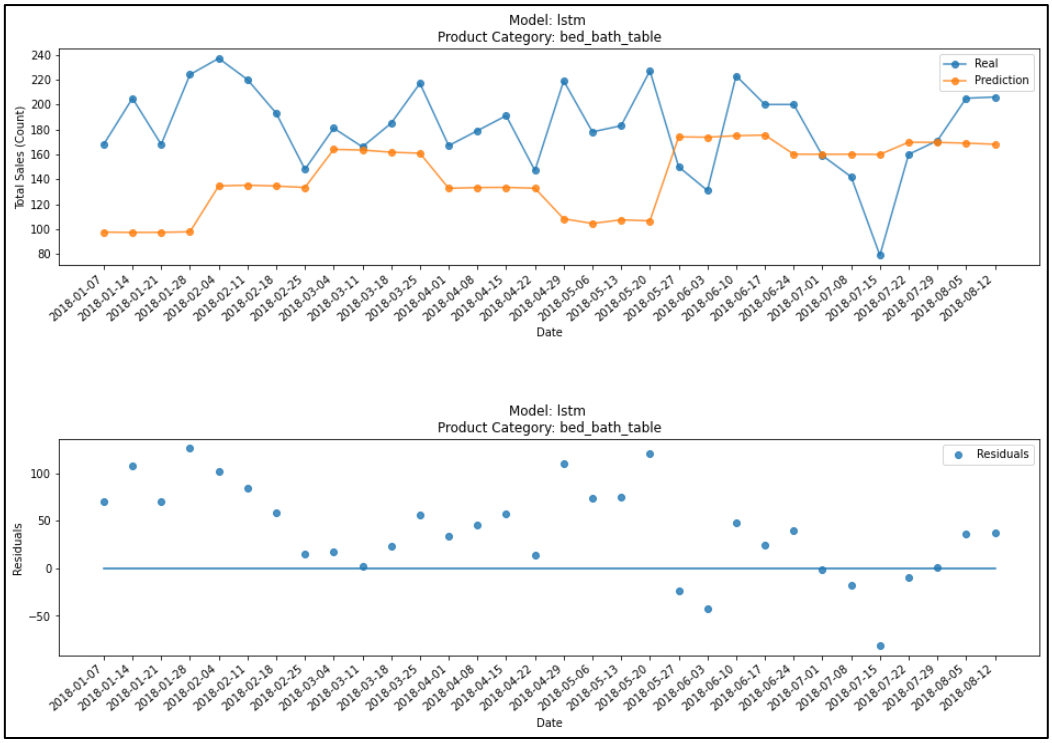


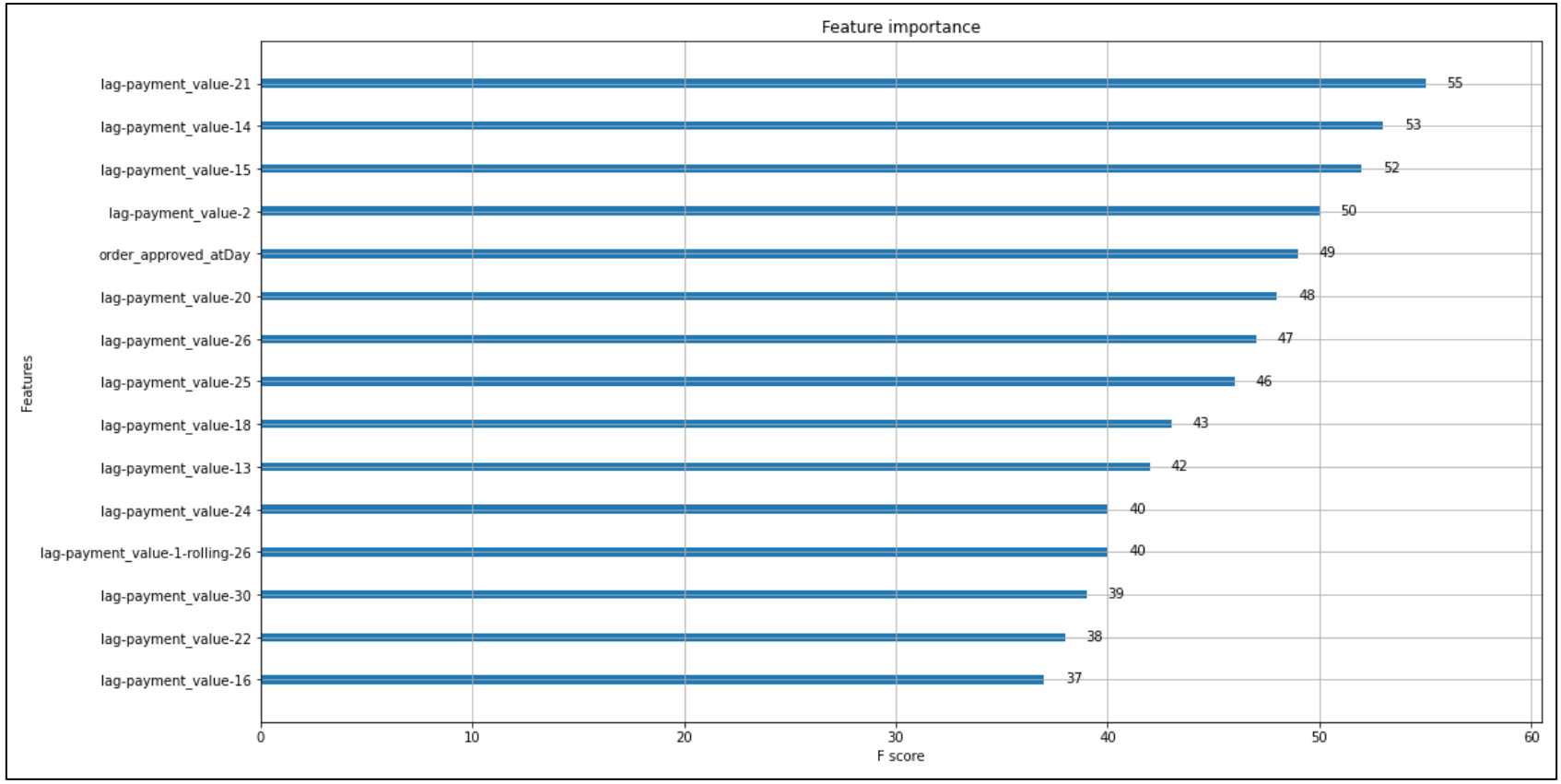
Figure 16, shows the predictions for the bed, bath and table product category using the XGBoost model. Similar to the LSTM model, there are frequent occasions where XGBoost predictions are under actual sales volumes (positive residuals). XGBoost tends to make predictions that are fluctuating from the mean value of sales.

Figure 16: Brazilian E-commerce Predictions – XGBoost

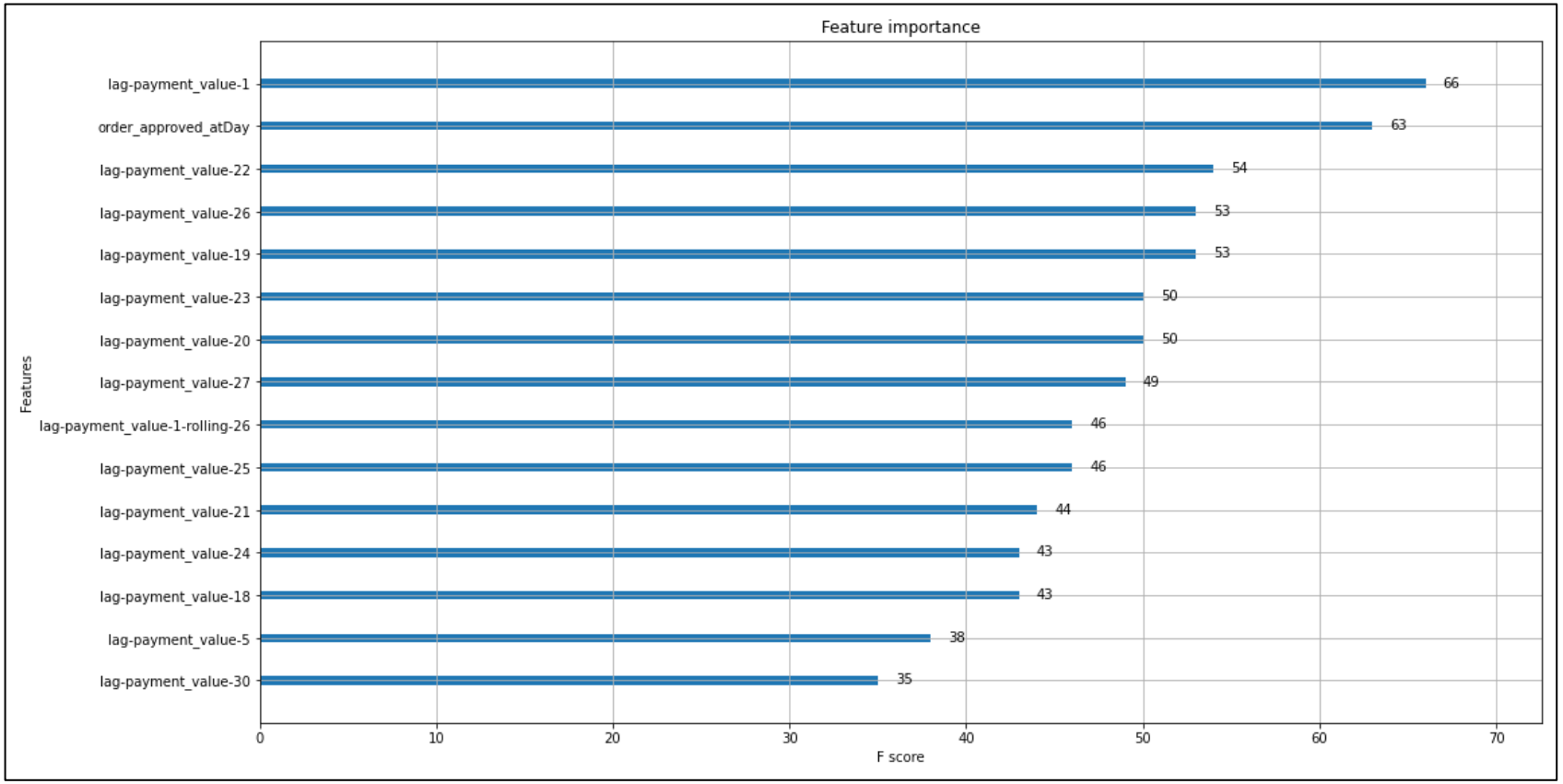
When using the XGBoost and LSTM models, the number of past time-steps for which lag features are created for the target variable are specified. During the experiment we test various sizes of target variable lags and the results discussed are based on 52 lag features created for XGBoost and LSTM. Figure 17, plots feature importance for the XGBoost model for two different test periods. The size of each test period is four weeks. Feature importance measures the number of times a variable is used in splitting a tree and weighted by the improvement to the model resulting from each split, averaged over all trees.

Figure 17: Brazilian E-commerce Predictions – XGBoost Feature Importance

Bed, Bath and Table. 4 Week Test Period Starting on: 2018-01-28.



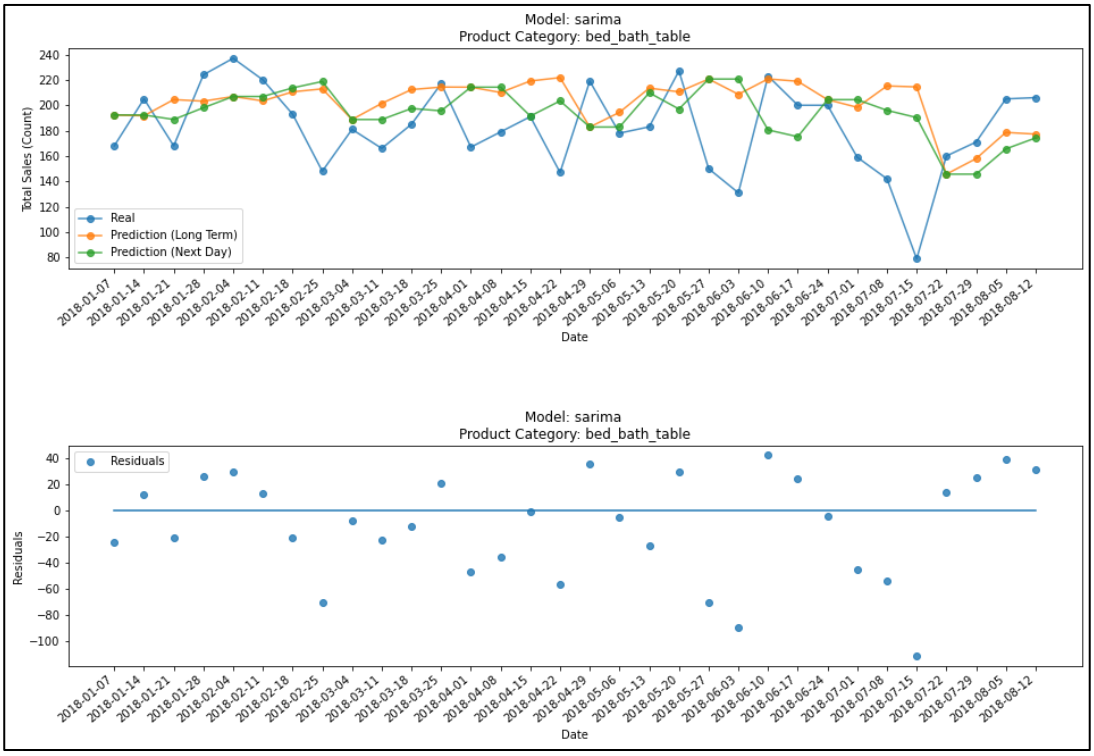
Bed, Bath and Table. 4 Week Test Period Starting on: 2018-02-25.



Viewing Figure 17, the XGBoost feature importance plot corresponding to the four-week test period starting on 2018-01-28, the lag feature representing sales from the past 21 time-steps stands out as the top feature used in splits. The derived feature representing the 26-week rolling mean of historical sales is the 12th most used feature in tree splits. However, other derived rolling mean features such as the two, four, six, eight etc. week rolling means are not considered as much in tree splits. Overall, for the test period beginning on 2018-01-28, lag features corresponding to the previous 10 to 30 past time-steps are used most heavily in tree splits. However, predictions made for the test period beginning on 2018-01-28 are far from the actuals as shown in Figure 16. Conversely, for the test period starting on 2018-02-25, the most important feature used in tree splits is the previous time step's sales followed by a derived feature representing the day of the year. Further, predictions made on the 2018-02-25 test period are generally closer to the actuals relative to the test period starting on 2018-01-28. This suggests that past weeks' sales and features that represent the time aspect of the data helps the XGBoost model's performance. Since, splits vary each time the model is run for the XGBoost model, the feature importance plots also change. This means that despite using the same hyperparameter values when the XGBoost model is run multiple times, the results and feature splits vary, leading to unstable results.

Figure 18, shows predictions using the SARIMA model that frequently overpredicts sales (negative residuals). Overall, SARIMA predictions are closer to the fluctuations in the actuals when using a one-step ahead forecast approach over long term, multi-step forecasts. All results presented and discussed for the SARIMA model are based on a one-step ahead forecasts.

Figure 18: Brazilian E-commerce Predictions – SARIMA



Despite the poor performance of the forecasting models that use only historical sales as data input relative to a naïve model, we integrate Google Trends with historical sales data using the Brazilian e-commerce dataset to explore potential performance improvements on specific product categories. Furthermore, Table 5, compares the performance of XGBoost and LSTM that use Google Trends with historical sales data as input with the XGBoost and LSTM models that do not contain Google Trends. Results suggest that the XGBoost and LSTM models that do not contain Google Trends do better than the XGBoost_GoogleTrends and LSTM_GoogleTrends models.

Table 5: Brazilian E-commerce Experiment 2 Results

Data Input	Model	Average MAPE	Average RMSE	Average WAPE	Average R^2
Historical Sales	XGBoost	0.34	51.99	4669.33	-1.15
Historical Sales & Google Trends	XGBoost_GoogleTrends	0.36	59.08	5282.20	-1.71
Historical Sales	LSTM	0.34	54.12	5450.81	-1.22
Historical Sales & Google Trends	LSTM_GoogleTrends	0.36	57.21	5888.69	-1.48

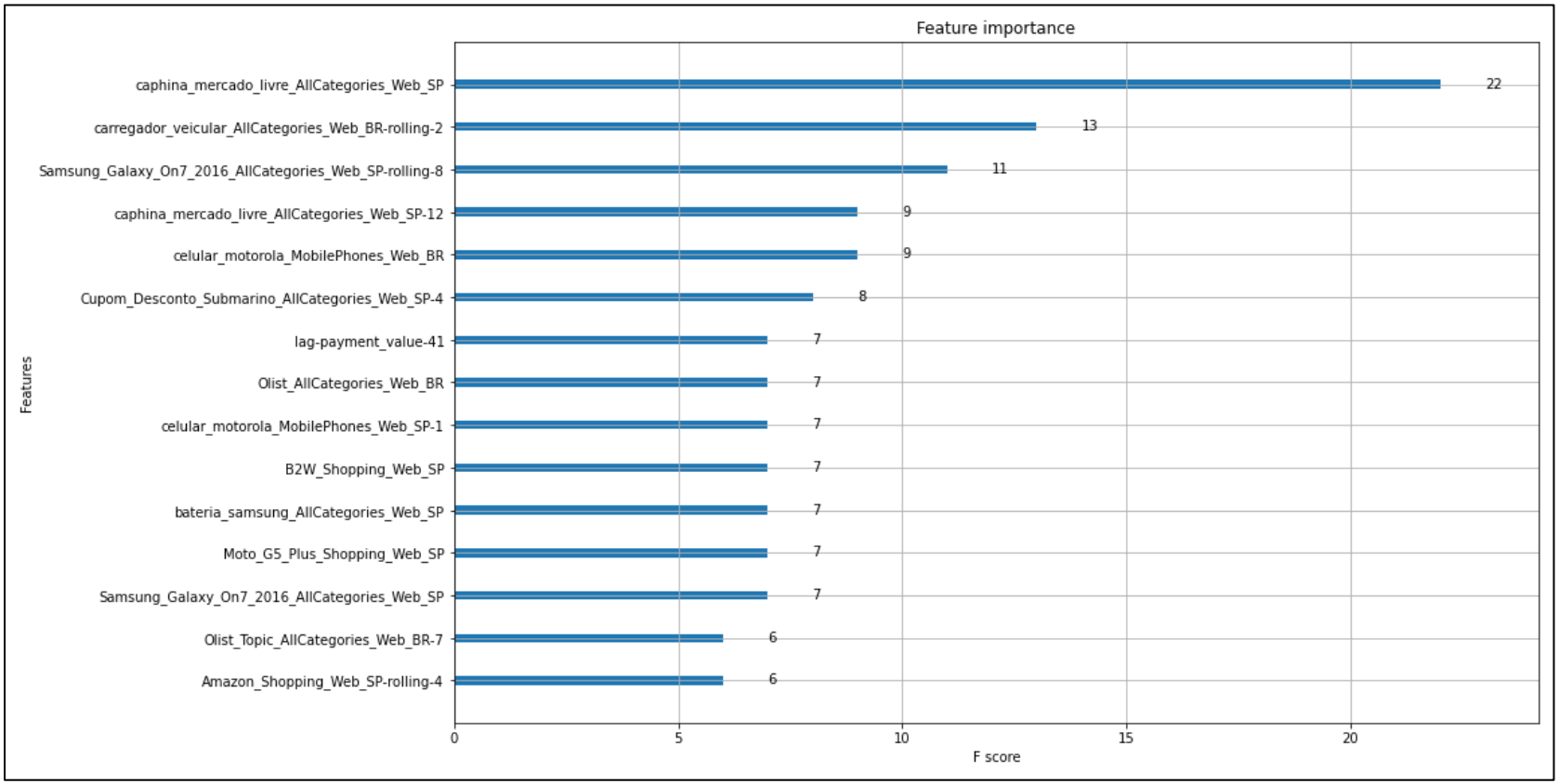
Using XGBoost, we test different sizes of lag features created for each Google Trends series used in sales forecasts. The results are based on 12 lag features generated for each Google Trends series used. Figure 19, shows XGBoost predictions that use only historical sales as data input, with XGBoost predictions that additionally use Google Trends as input, for the telephony and furniture décor product categories. For the telephony product category, including Google Trends reduces the standard deviation of predictions to 22.33 from 25.65 observed in the XGBoost model using only historical sales as input. Towards the second half of the testing period, starting on May 2018, the inclusion of Google Trends data seems to help the XGBoost model in adjusting to the decreasing trend in actual sales relative to the predictions generated using only historical sales. In contrast, for the furniture décor product category, the standard deviation in the predictions of XGBoost using Google Trends is higher than the standard deviation of XGBoost predictions that use only historical sales as data input. The inclusion of Google Trends in forecasting sales of the furniture décor product category increases fluctuations of predictions from the mean value of sales, while helping to capture changes in sales like the peak occurring on the week of 2018-01-21.

Figure 20, depicts the XGBoost model's feature importance that uses Google Trends as data input to predict telephony and furniture décor product categories' sales on a selected test period. Among the Google Trends data used in forecasting the sales of telephony, the search terms "caphina mercado livre," "carregador veicular" and "Samsung Galaxy on7 2016" are the top three features used to split trees. Further, the derived feature representing the 2-week rolling mean of the Google Trends searches for mobile chargers ("carregador veicular") and the 8-week rolling mean for "Samsung Galaxy on7 2016" are used in more splits over the lag features for the target variable. Similarly, for the furniture décor product category, lag features representing Google Trends searches for living room and home decoration are among the top features used in tree split.

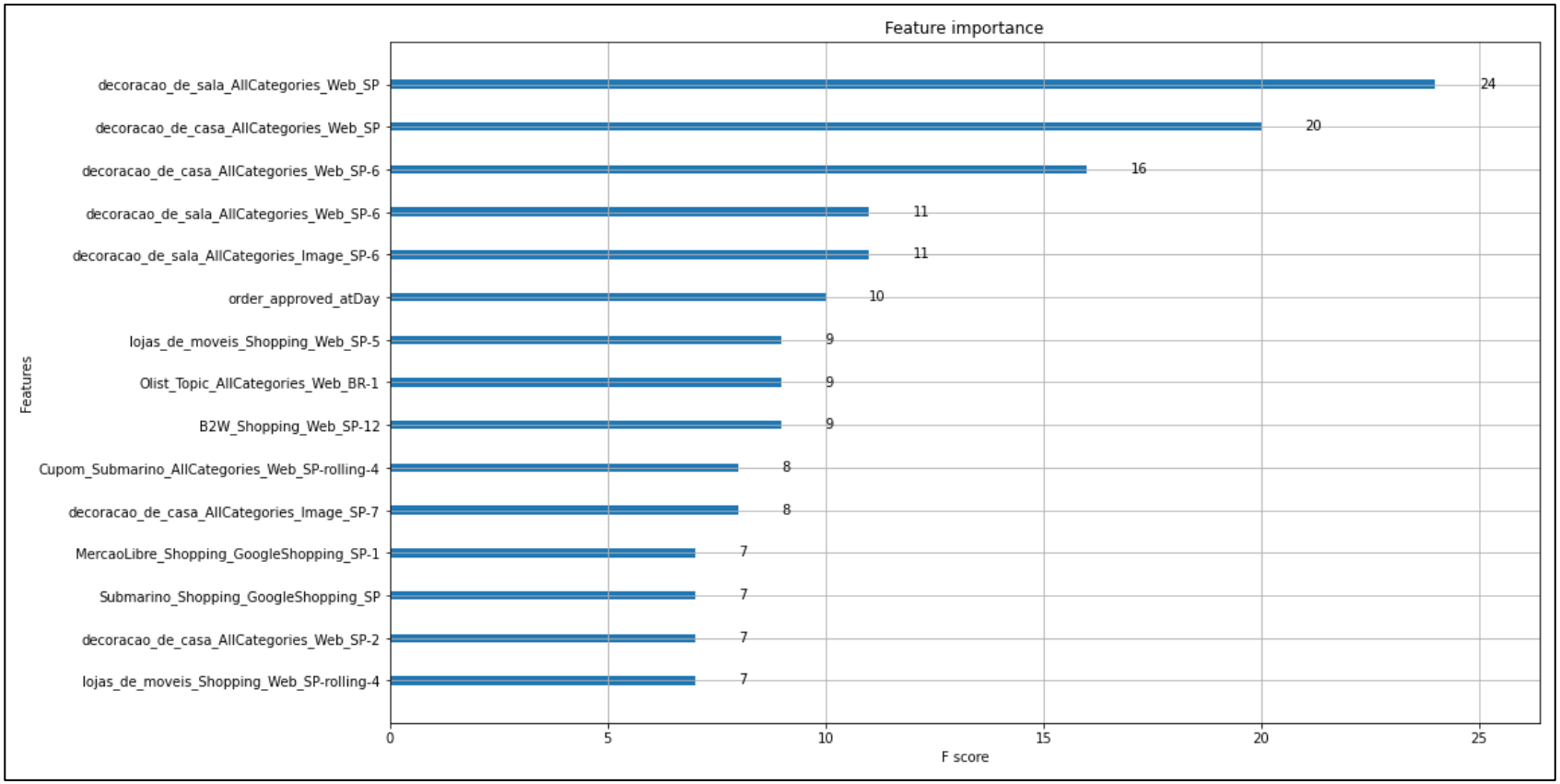
Figure 19: Brazilian E-commerce Predictions Using Google Trends – XGBoost

Figure 20: Brazilian E-commerce Predictions Using Google Trends – XGBoost Feature Importance

Telephony, Test Period 2018-06-17



Furniture Décor, Test Period 2018-01-28



Although the inclusion of Google Trends improves the performance XGBoost for the telephony product category as measured by MAPE, RMSE, WAPE and R^2, the size of improvements on the R^2 and MAPE metrics are negligible. Table 6, shows the average performance over all time-series folds for the telephony as well as the sports and leisure product categories using the XGBoost and LSTM models. Possibly, larger accuracy improvements can be achieved from including Google Trends if prior to including Google Trends in a model, a regression analysis method such as lasso, is performed to determine the Google Trends series to include in the model (Tibshirani, 1996). This may lead to improvements in forecasting accuracy with larger magnitudes and has been adopted in previous studies, to exemplify, Robin (2018).

Table 6: Brazilian E-commerce – Selected Results by Product Category

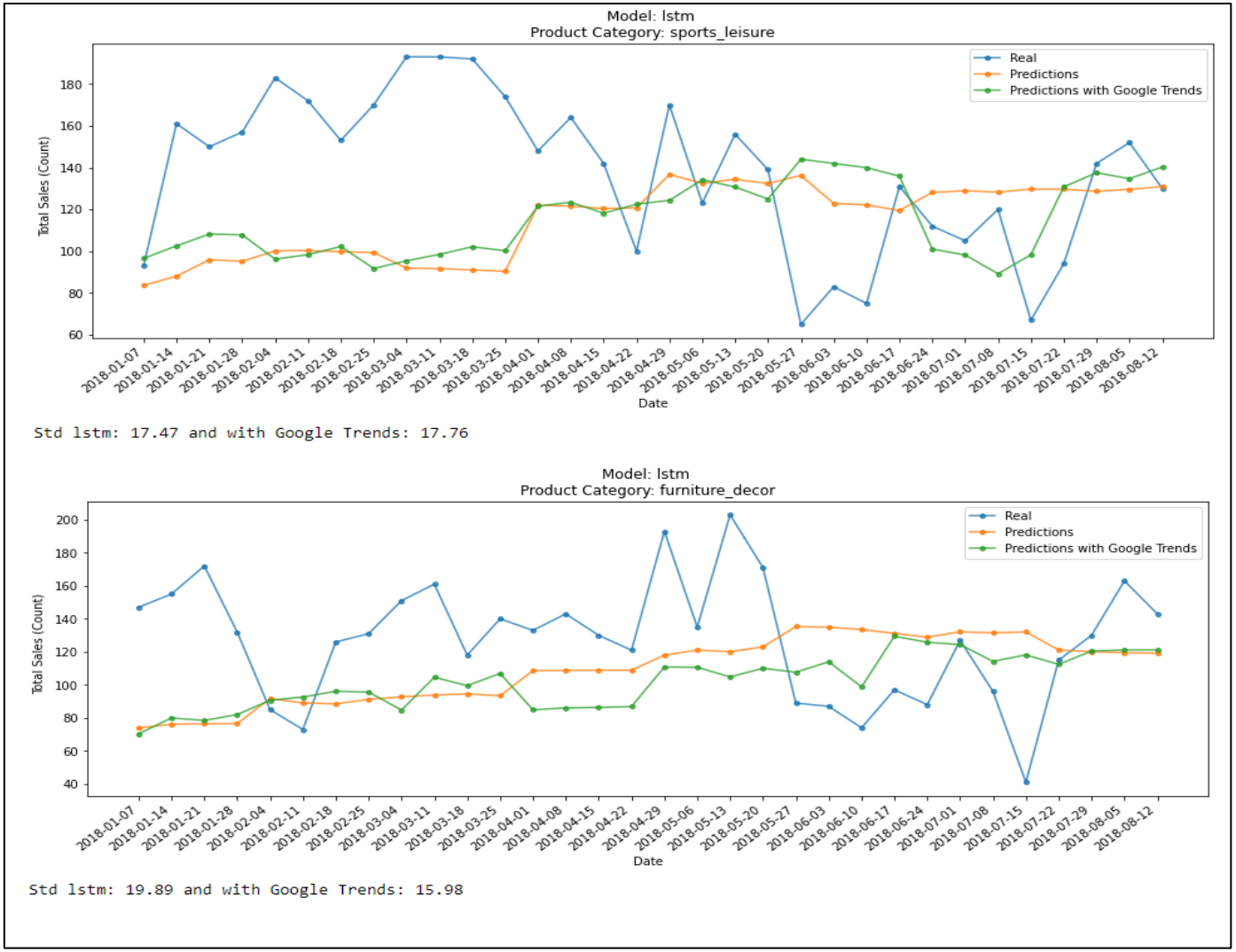
Product Category	Model	MAPE	RMSE	WAPE	R^2
Telephony	XGBoost	0.44	38.68	1892.81	-1.97
	XGBoost_GoogleTrends	0.35	33.04	1752.06	-1.17
Sports and Leisure	LSTM	0.33	53.19	5516.14	-1.09
	LSTM_GoogleTrends	0.32	51.70	5326.26	-0.98

Similar to the performance of the XGBoost model that uses historical sales and Google Trends as data input, results in Table 6, show that including Google Trends as data input in the LSTM model does not lead to more accurate forecasts relative to the LSTM model containing only historical sales for the majority of product categories considered. In particular, performance improves for 2 product categories out of the 7 in scope as measured by MAPE, RMSE, WAPE and R^2. Specifically, the forecast accuracy increases for the sports & leisure and furniture décor product categories on most metrics considered when LSTM uses Google Trends. As shown in Table 6, for the sports and leisure category, although forecasts are more accurate when using the LSTM model containing Google Trends relative to the LSTM model using only historical sales as data input, the size of improvements is not large for the metrics considered.

Figure 21, shows LSTM predictions that use Google Trends as data input relative to the LSTM model using only historical sales as data input for the sports and leisure as well as the furniture décor product categories. Including Google Trends using the LSTM model leads to small change in the standard deviation of predictions in the sports and leisure product category while reducing the standard deviation in the case furniture décor. Further, the performance on the furniture décor product category improves for MAPE, RMSE and R^2.

Despite using the same Google Trends data with XGBoost and LSTM, performance improvements are not consistent for each model across the product categories considered. On one hand this implies that the Google Trends search terms used are not strong predictors. Further, there is no strong evidence that Google Trends collected on one product category is relatively more important than the Google Trends collected for other product categories. On the other hand, including Google Trends for the furniture décor product category, overall, improves the performance for XGBoost and LSTM as measured by MAPE, RMSE, WAPE and R^2. This can be interpreted as a positive signal indicating the usefulness of Google Trends data included for the furniture décor product category. Unlike in the XGBoost model where the feature importance plot helps to gain an understanding on which variables the model mostly relies on when making predictions, due to the nature of the LSTM model it is not possible to make such granular analysis. In other words, the LSTM model results are less interpretable. Additionally, due to the way LSTM is stacked, the size of lag variables created for Google Trends is the same size as the lag variables created for the target variable, that is 52. Whereas, for XGBoost we create 12 lag variables for each Google Trends series and 52 lag variables for the target variable.

Figure 21: Brazilian E-commerce – LSTM Predictions Using Google Trends



Furthermore, to benchmark the performance of the XGBoost and LSTM models that use Google Trends with a naïve model, Table A8 (see Appendix), presents the MASE and RMSSE scores. Generally, all predictions that use Google Trends are worse off than a naïve model since the scores on MASE and RMSSE are greater than 1. However, for some product categories such as furniture décor, the MASE score for the XGBoost and LSTM models that use Google Trends is relatively lower than the MASE score of the XGBoost and LSTM models that do not use Google Trends. In summation, comparing the forecasting accuracy for the XGBoost and LSTM models that use historical sales as data input with XGBoost and LSTM that use Google Trends as data input, no consistent pattern of performance improvement is observed.

The null hypothesis is that XGBoost and LSTM predictions that use only historic sales as data input are statistically identical, belonging to the same distribution as XGBoost and LSTM predictions that additionally use Google Trends as data input, respectively. For the Brazilian e-commerce dataset, sales predictions on 7 product categories using XGBoost and LSTM models containing only sales history as input, are paired with the predictions generated using XGBoost and LSTM models that also contain Google Trends as data input.

The computed p-values for XGBoost (0.139305) and LSTM (0.176788) models respectively, are above 0.05 and therefore we do not reject the null hypothesis. The p-values computed suggest that there is no strong evidence on statistically significant difference between predictions that do not use Google Trends and predictions that use Google Trends. In what follows, we present and discuss findings for the Breakfast at the Frat dataset.

Breakfast at the Frat

Table 7, compares the performance of models that use historic sales data input to predict the weekly number of units sold for 4 products in 3 different stores. For each forecasting model (SARIMA, FBProphet, XGBoost and LSTM) the average performance as measured by MAPE, RMSE, WAPE and R^2 is based on the 12 possible product-store combinations for which sales predictions are made for. The LSTM model seems to outperform all models as measured by MAPE, RMSE, WAPE and R^2. However, this does not imply that the LSTM model simultaneously scores best on MAPE, RMSE, WAPE and R^2, for a product, in all stores. Notably, the performance of XGBoost as measured by RMSE, WAPE and R^2 is worse, relative to all other models considered.

Table 7: Breakfast at the Frat – Experiment 1 Results

Data Input	Model	Average MAPE	Average RMSE	Average WAPE	Average R^2
Historical sales	SARIMA	0.49	52.67	3758.13	-0.31
	FBProphet	0.54	52.93	3397.56	-0.35
	XGBoost	0.41	57.14	4093.34	-0.70
	LSTM	0.36	48.00	3696.69	-0.11

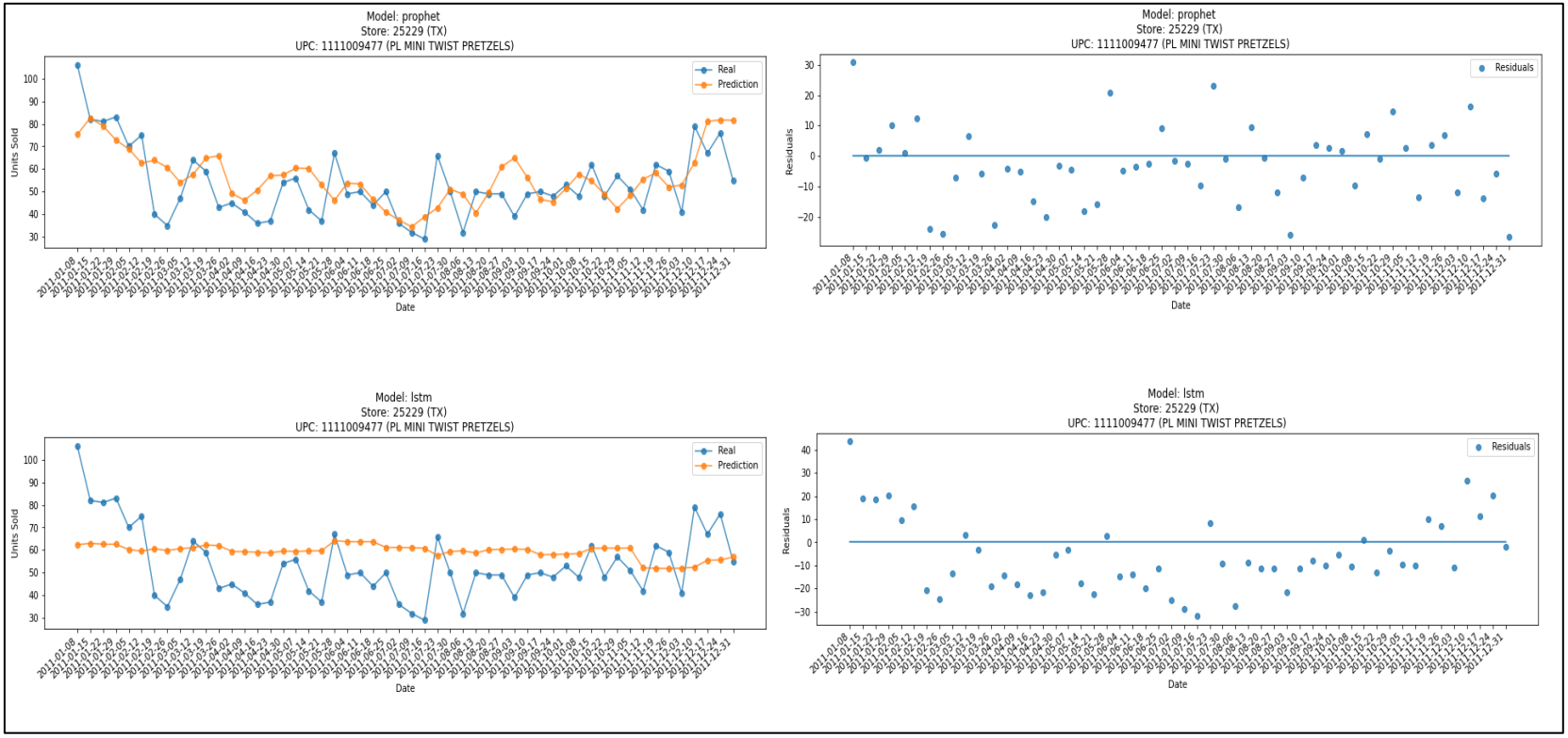
Despite the dominance of LSTM over other models as measured by average performance on MAPE, RMSE, WAPE and R^2 for all product-store combinations, overall, the forecast error of all models does not seem far from each other as summarized with Table 7. Table 8, compares the performance of models for selected product-store combinations. Furthermore, in Table 8, the performance of FBProphet and LSTM in the predicting the sales of private label mini twist pretzels in two stores, located in Texas and Kentucky is contrasted. While FBProphet outperforms LSTM in predicting the sales of private label pretzels in Texas, LSTM generates more favorable predictions over FBProphet in Kentucky. Results in Table 8, suggest that while a model may achieve the highest prediction accuracy over other models for a specific product at a specific store, another model may achieve higher accuracy for the same product that is sold at another store. Furthermore, for a product-store combination, depending on the metric used to measure performance, a model may be favorable over another.

Table 8: Breakfast at the Frat – Selected Results by Product and Store

Product	Store	Model	MAPE	RMSE	WAPE	R^2
Honey Nut Cheerios (UPC: 1600027527)	Kentucky	SARIMA	0.55	88.79	6281.30	-0.70
		FBProphet	0.88	86.77	4260.27	-0.63
		XGBoost	0.37	87.35	6339.45	-0.65
		LSTM	0.54	70.34	5510.05	-0.07
Digiorno Pepperoni Pizza (UPC: 7192100339)	Ohio	SARIMA	0.79	42.49	1719.56	-0.36
		FBProphet	0.53	33.08	1267.47	0.17
		XGBoost	0.53	45.95	2144.19	-0.59
		LSTM	0.60	38.52	1704.45	-0.12
Kellogg's Frosted Flakes (UPC: 3800031838)	Texas	SARIMA	0.36	44.33	1959.71	-0.27
		FBProphet	0.46	47.73	1917.42	-0.48
		XGBoost	0.35	54.99	2381.89	-0.96
		LSTM	0.26	41.18	1972.52	-0.10
Private Label Mini Twist Pretzels (UPC: 1111009477	Texas	FBProphet	0.21	13.00	314.85	0.30
		LSTM	0.30	16.89	451.31	-0.19
	Kentucky	FBProphet	0.23	31.13	1997.98	-1.50
		LSTM	0.16	20.71	1227.43	-0.11

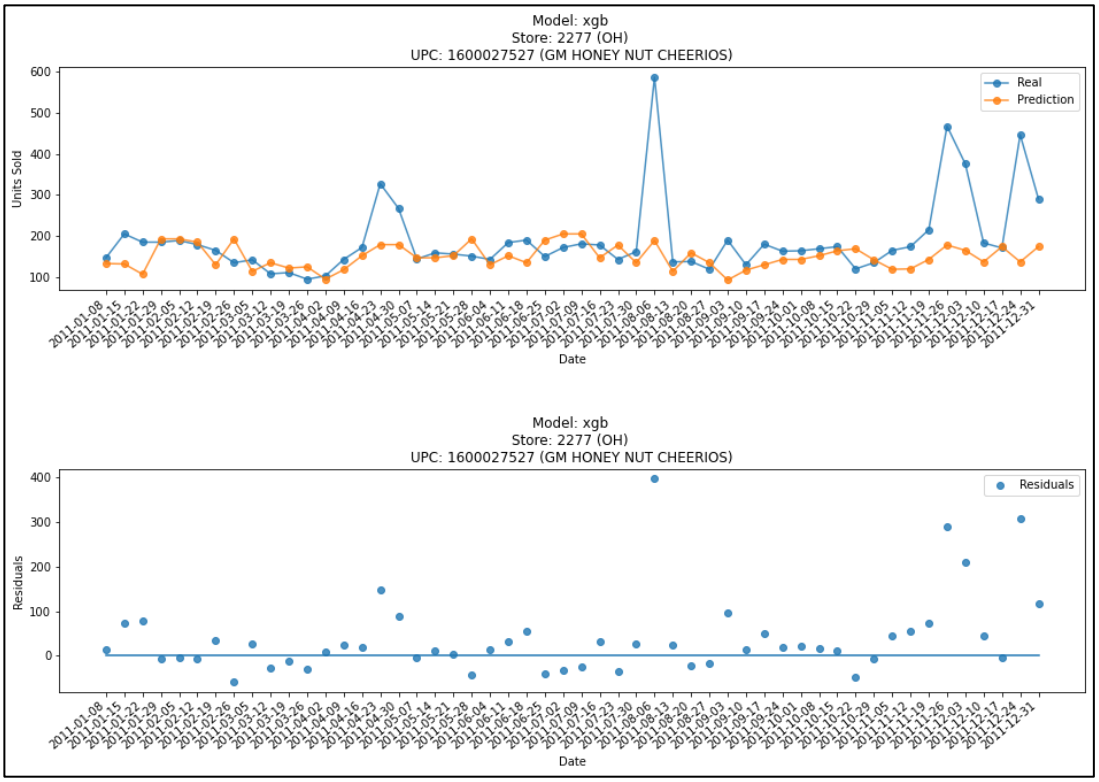
Figure 22, plots predictions and residuals of the FBProphet and LSTM models for private label mini twist pretzels sales at the Texas store. Looking at Figure 22, LSTM predictions are equivalent to more or less the mean value of sales throughout all folds we test the model on. Additionally, the residuals plot for LSTM shows that sales are overpredicted (negative residuals) frequently. Whereas, predictions using FBProphet seem to adjust to trends in sales as seen with the sharp decline that occurs during the first quarter of 2011. Visually, the line showing predictions using FBProphet in Figure 22, mimics the behavior in actuals. However, the residuals are fairly scattered in terms of overpredicting as much as underpredicting sales. In practice, differences in consumption patterns across stores may be accredited to external factors like demographics or internal factors controlled by the retailer, such as, store specific promotions. The sales history of private label mini twist pretzels at the store located in Kentucky contains less fluctuations relative to sales in Texas. Consequently, predictions around the mean generated by the LSTM model for pretzel sales in Kentucky may explain the favorable performance over FBProphet, presented in Table 8.

Figure 22: Breakfast at the Frat Predictions – FBProphet vs. LSTM



Inspecting the XGBoost model's results, we find the strongest performance to be for predicting sales of Honey Nut Cheerios in Ohio as measured by RMSE, WAPE and R^2 compared to the performance of other models used to predict Honey Nut Cheerios sales in Ohio. Figure 23, shows XGBoost's predictions and residuals for Honey Nut Cheerios sales in Ohio. Overall, residuals are close to zero and the model seems to be capturing actual sales behavior when the entire test period is considered. However, although XGBoost predictions estimate the direction of peaking sales that occur during the weeks of 2011-04-30, 2011-08-06 and 2011-12-03 the residuals are considerably large for those spikes and especially during the week of 2011-08-06. The peaks in sales may be as a result of promotions or potentially outlier data points.

Figure 23: Breakfast at the Frat Predictions – XGBoost

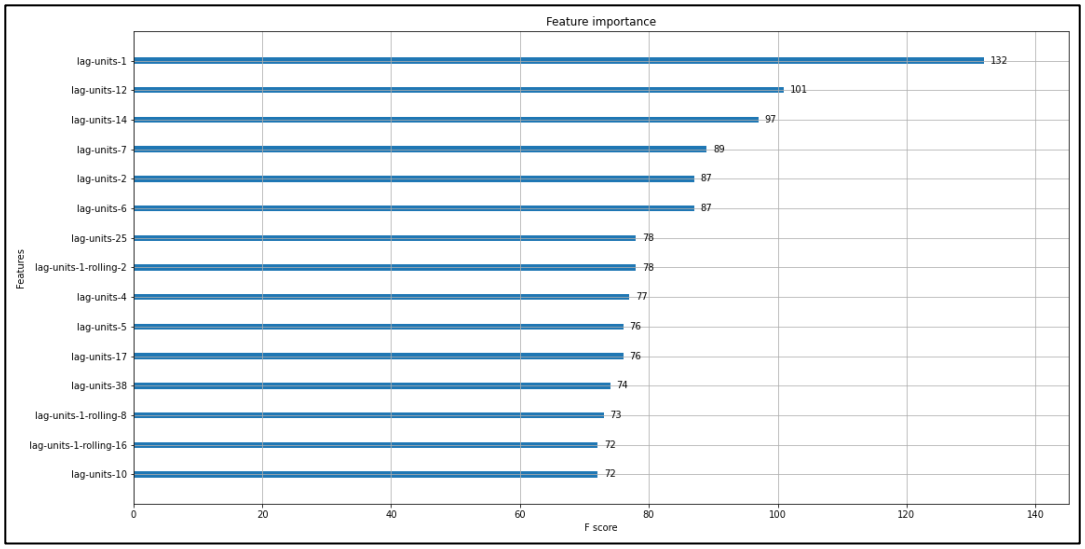


Furthermore, Figure 24, plots the XGBoost model's feature importance during the 4-week test periods ending on 2011-01-29 and 2011-08-13. During the test period for 2011-01-29 where predictions are fairly close to actuals as shown in Figure 23, the most important feature used in tree splits is the lag variable representing past week's sales as illustrated

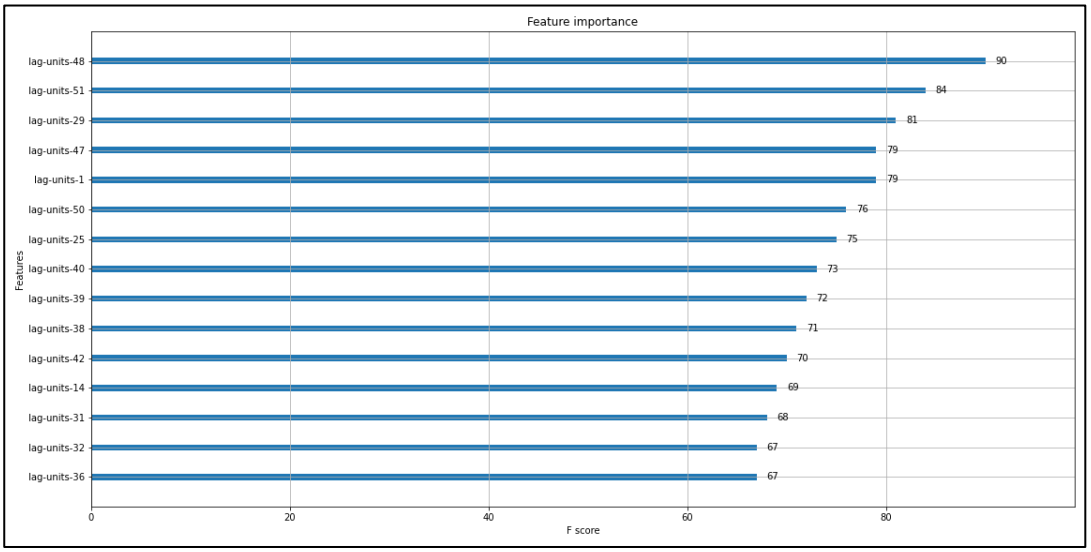
with Figure 23. Additionally, among the top features used for splits during the 2011-01-29 test period are lag variables representing the past 12 and 14 weeks' sales as well as the derived 2, 8 and 16 week rolling mean features on past sales. In contrast, during the test period 2011-08-13 the top 3 most used features in tree splits are the lag variables that represent the past 48, 51 and 29 weeks' sales. Since forecast error during the test period 2011-08-13 is larger than 2011-01-29, lag variables representing more recent sales history seem to help in generating more accurate predictions. Additionally, derived features on geometric rolling means of sales, depending on the test period, may help the XGBoost model in capturing time-series properties, leading to more accurate predictions.

Figure 24: Breakfast at the Frat Predictions – XGBoost Feature Importance

Honey Nut Cheerios, Ohio. Test Period 2011-01-29



Honey Nut Cheerios, Ohio. Test Period 2011-08-13



To sum up the results from forecasting the sales of 4 products in 3 stores using only historical sales, the LSTM model seems to be outperforming all models as measured by RMSE, MAPE and R^2. In the majority of cases, LSTM predictions revolve around the mean value of sales while FBProphet predictions tend to fluctuate more from the mean and reflect the changing patterns in actual sales. There is no strong evidence of a single model outperforming all models across multiple products, stores, and performance metrics. In some cases, we find the LSTM model scoring best on MAPE, RMSE, WAPE and R^2 for a specific product at a specific store, while FBProphet may do better on all metrics for the same product in another store. Nevertheless, while XGBoost, SARIMA and FBProphet fall behind the performance of LSTM, the differences are not large on all metrics considered.

To compliment the analysis on model accuracy, Table A9 (see Appendix), presents MASE and RMSSE scores for SARIMA, FBProphet, XGBoost and LSTM in an effort to benchmark the performance of all models considered against a naïve model. Results in Table A9 (see Appendix), suggest that for all models considered, MASE and RMSSE scores are mixed, with some being over 1 and some being below 1, depending on the product-store combination for which predictions are made for. Generally, for all models considered, the RMSSE score is below 1 implying favorable predictions over a naïve model. In what follows, we analyze performance changes in the XGBoost and LSTM models when Google Trends is combined with historical sales data to make sales forecasts.

Table 9, contrasts the performance of XGBoost and LSTM models that contain historical sales as data input with the XGBoost and LSTM models that also include Google Trends as input. Each performance metric is computed based on the average score on the 12 product-store combinations for which predictions are made for. Interestingly, the performance of the LSTM_GoogleTrends model is worse than LSTM as measured by MAPE, RMSE, WAPE and R^2. Whereas, the performance of the XGBoost_Google Trends model is favorable over the XGBoot model as measured by WAPE and R^2.

Table 9: Breakfast at the Frat Experiment 2 Results

Data Input	Model	Average MAPE	Average RMSE	Average WAPE	Average R^2
Historical Sales	XGBoost	0.41	57.14	4093.34	-0.70
Historical Sales & Google Trends	XGBoost_GoogleTrends	0.45	57.88	3951.95	-0.67
Historical Sales	LSTM	0.36	48.00	3696.69	-0.11
Historical Sales & Google Trends	LSTM_GoogleTrends	0.39	49.87	3811.68	-0.21

Table 10, compares the performance of the XGBoost model with and without Google Trends for selected product-store combinations. Notably, prediction accuracy for private label mini twist pretzels sales in Ohio, Kentucky and Texas is higher when the XGBoost_GoogleTrends model is used relative to the XGBoost model using only historical sales as data input. Although the size of forecast error improvements for mini twist pretzels in all stores is not large, MAPE, RMSE, WAPE and R^2 scores are better across all stores, implying a positive indication on the usefulness of Google Trends used to make the predictions. Similarly, prediction accuracy for the sales of Kellogg's Frosted Flakes improves for the stores located in Ohio and Texas, when the XGBoost_GoogleTrends model is used over XGBoost that only contains historical sales. Additionally, prediction accuracy for Honey Nut Cheerios sales in the Kentucky and Texas stores is generally higher when using the XGBoost_GoogleTrends model over XGBoost that only uses historical sales. This implies that the Google Trends data collected for forecasting sales of products in the cold cereal category are aiding XGBoost's forecast accuracy.

Table 10: Breakfast at the Frat Predictions Using Historical Sales & Google Trends – XGBoost

Store	Product	Data Input	MAPE	RMSE	WAPE	R^2
Ohio (2277)	Kellogg's Frosted Flakes (UPC: 3800031838)	Historical Sales	0.49	73.46	4341.41	-0.91
		Historical Sales & Google Trends	0.44	69.03	4197.63	-0.69
	Private Label Mini Twist Pretzels (UPC: 1111009477)	Historical Sales	0.21	56.65	7343.83	-0.80
		Historical Sales & Google Trends	0.22	53.93	6725.79	-0.63
Kentucky (389)	Honey Nut Cheerios (UPC: 1600027527)	Historical Sales	0.37	87.35	6339.45	-0.65
		Historical Sales & Google Trends	0.44	68.79	3714.90	-0.02
	Digiorno Pepperoni Pizza (UPC: 7192100339)	Historical Sales	0.78	40.36	1177.50	-0.87
		Historical Sales & Google Trends	0.88	38.77	1108.58	-0.73
	Private Label Mini Twist Pretzels (UPC: 1111009477)	Historical Sales	0.20	26.21	1395.11	-0.77
		Historical Sales & Google Trends	0.16	23.90	1516.42	-0.47
Texas (252299)	Kellogg's Frosted Flakes (UPC: 3800031838)	Historical Sales	0.35	54.99	2381.89	-0.96
		Historical Sales & Google Trends	0.36	48.18	2078.18	-0.50
	Honey Nut Cheerios (UPC: 1600027527)	Historical Sales	0.42	86.19	6182.93	-0.39
		Historical Sales & Google Trends	0.40	85.09	6191.02	-0.36
	Private Label Mini Twist Pretzels (UPC: 1111009477)	Historical Sales	0.33	20.41	485.15	-0.73
		Historical Sales & Google Trends	0.30	18.57	456.04	-0.44

Figure 25, shows XGBoost predictions with and without using Google Trends for the sales of mini twist pretzels in Ohio and the feature importance plot on a test period for the XGBoost model that contains Google Trends. Overall, predictions using Google Trends appear to better follow the volume of pretzel sales in Ohio while the XGBoost model that only uses historical sales as data input, frequently underpredicts sales by a larger amount than the XGBoost model that uses Google Trends.

Figure 25, suggests that XGBoost seems to rely on lag variables created for "Frito Lay" Google searches in Ohio as the top feature used in tree splits followed by hits for "Amazon Fresh" across the US. Although, for a search term, to exemplify "PepsiCo" we collect multiple Google Trends series that vary by location (ex: Kentucky, Ohio, and Texas) the feature importance plot in Figure 25 suggests that the XGBoost_GoogleTrends model determines "PepsiCo" searches in Ohio as more relevant to use in forecasting the sales of pretzels at the store located in Ohio. Interestingly, in Figure 25, lag variables representing historical sales are not among the top 5 features used in splits as one may expect. Rather, Google Trends series used across product categories such as Google searches for "printable coupons for groceries" in addition to category specific Google Trends such as brands believed to influence the sales of the private label mini twist pretzels like Synder's of Hanovar and Utz Quality Foods are among the most important features used by the XGBoost model. Unlike in the case of XGBoost where the inclusion of Google Trends data, although not large, bring some improvement to the overall forecasting accuracy as measured by multiple metrics for multiple products sold across 3 stores, Google Trends data seem to be confusing the LSTM model and not leading to generally successful improvements.

Figure 25: Breakfast at the Frat Predictions Using Google Trends – XGBoost

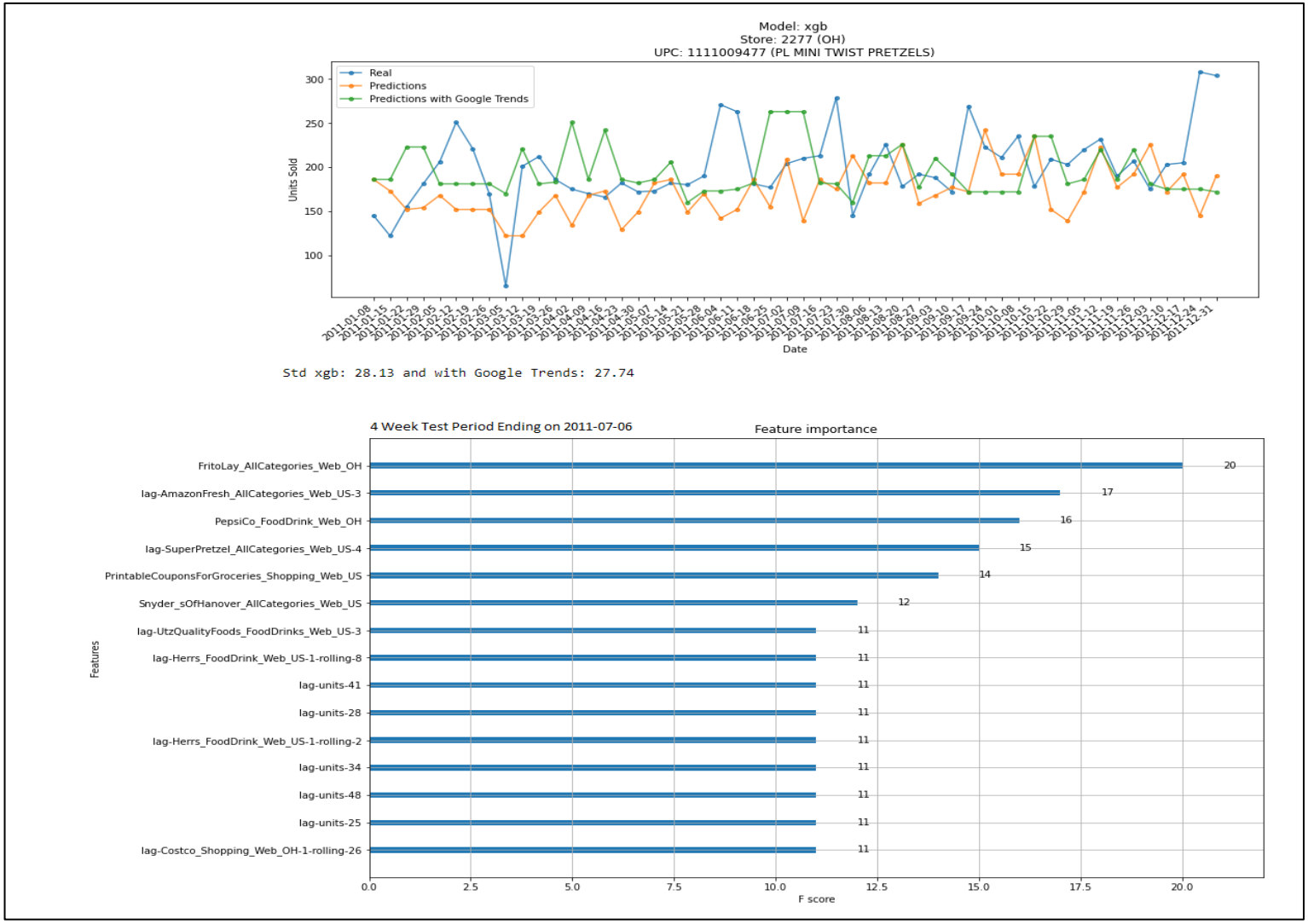
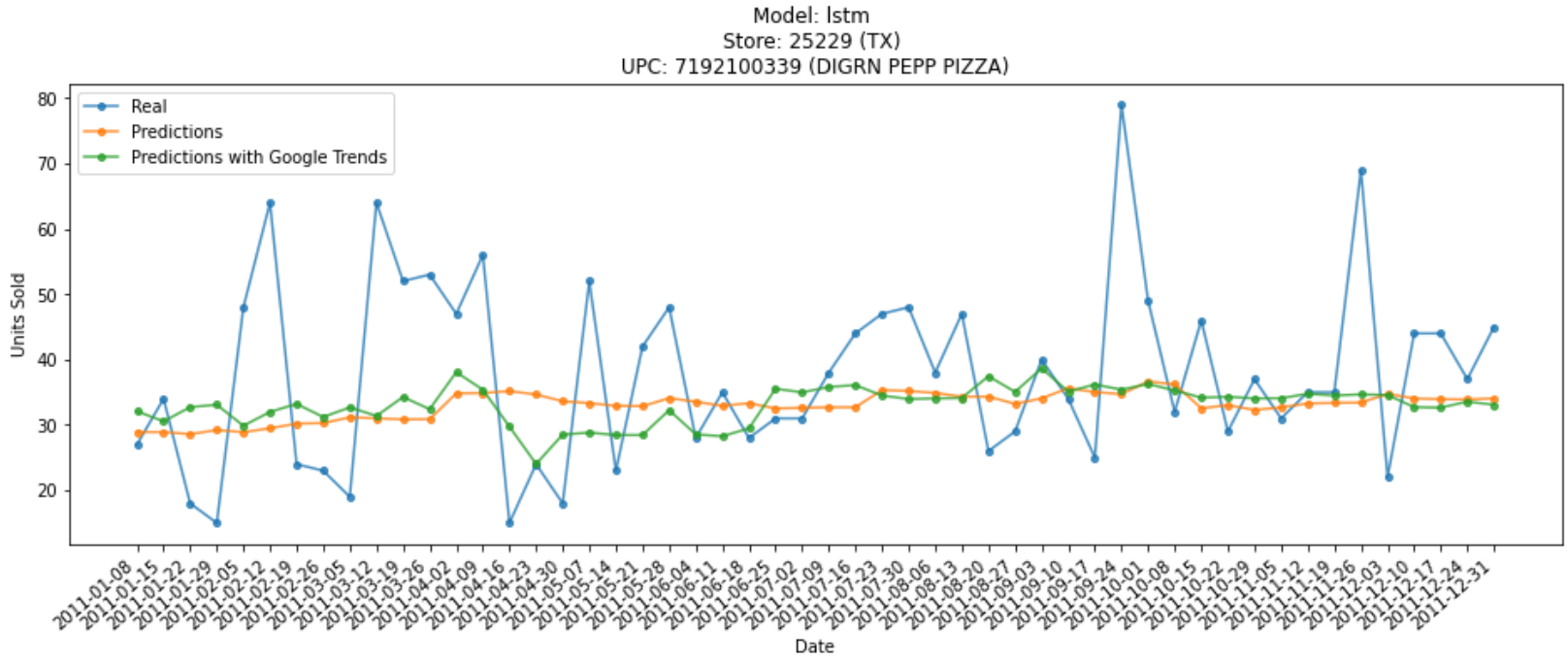


Figure 26, contrasts the predictions of the LSTM model using Google Trends with the LSTM model that only uses historical sales data to predict sales of Digiorno Pepperoni Pizza at the store located in Texas. Figure 26, shows how the performance of the LSTM model using Google Trends is stronger than the LSTM model not using Google Trends as measured by MAPE, RMSE, WAPE and R^2. However, performance differences are quite small, almost negligible, and visually it is not possible to interpret a major change in LSTM predictions that generally revolve around the mean value of sales. Including Google Trends for predicting the sales of DiGiorno Pepperoni Pizza in Texas leads to less than 1 percentage point change on the standard deviation of predictions that use Google Trends over the standard deviation of the LSTM model that only relies on historical sales as data input. Due to the nature of the LSTM model, it is not possible to analyse feature importance for Google Trends as conducted with the XGBoost model.

In summary, using historical sales and Google Trends to make predictions on 12 product-store combinations, the performance of XGBoost seems to improve relative to the XGBoost model that does not contain Google Trends as measured by, WAPE and R^2. However, the magnitude of improvements on each metric is not large and in most cases, there are fractional improvements. In contrast, in the majority of cases out of the 12 product-store combinations, the LSTM model that does not contain Google Trends performs relatively better than the LSTM model that uses Google Trends as measured by MAPE, RMSE, WAPE and R^2. Since MAPE, RMSE, WAPE and R^2 penalize error differently we benchmark the performance of the XGBoost and LSTM models against a naïve model using MASE and RMSSE.

Figure 26: Breakfast at the Frat Predictions Using Google Trends – LSTM



Std lstm: 2.03 and with Google Trends: 2.88

Store	Product	Data Input	Model	MAPE	RMSE	WAPE	R^2
Texas (252299)	Digiorno Pepperoni Pizza (UPC: 7192100339)	Historical Sales	LSTM	0.30	14.73	357.24	-0.11
		Historical Sales & Google Trends	LSTM_GoogleTrends	0.29	14.48	349.29	-0.07

Table 11 compares the performance of the XGBoost and LSTM models that use and do not use Google Trends on the MASE and RMSSE score. A score that is larger than 1 for MASE and RMSSE implies that model predictions are worst than the out of sample forecasts from a naïve model and therefore a score less than 1 is favorable. Results presented in Table 11 are diverse. The highlighted cells represent the model that scores best on MASE and RMSSE in predicting the sales of a product at a specific store.

Table 11: Breakfast at the Frat – Experiments 1 & 2 Results by MASE & RMSSE

Store	Product	Model	MASE	RMSSE
Ohio (ID: 2277)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	0.96	0.80
		XGBoost_GoogleTrends	1.41	1.10
		LSTM	0.90	0.81
		LSTM_GoogleTrends	0.86	0.79
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.22	0.98
		XGBoost_GoogleTrends	1.13	0.92
		LSTM	0.79	0.74
		LSTM_GoogleTrends	0.83	0.76
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.51	1.42
		XGBoost_GoogleTrends	1.45	1.35
		LSTM	1.24	1.22
		LSTM_GoogleTrends	1.28	1.25
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.23	1.11
		XGBoost_GoogleTrends	1.46	1.26
		LSTM	1.09	0.93
		LSTM_GoogleTrends	1.09	0.90
Kentucky (ID: 389)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	0.80	0.73
		XGBoost_GoogleTrends	0.73	0.58
		LSTM	0.90	0.59
		LSTM_GoogleTrends	0.89	0.68
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.08	1.06
		XGBoost_GoogleTrends	1.11	1.11
		LSTM	0.92	0.95
		LSTM_GoogleTrends	1.19	1.04
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.10	1.12
		XGBoost_GoogleTrends	1.01	1.02
		LSTM	0.92	0.89
		LSTM_GoogleTrends	0.97	0.96
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.42	1.29
		XGBoost_GoogleTrends	1.41	1.24
		LSTM	1.09	0.95
		LSTM_GoogleTrends	1.17	1.05

Store	Product	Model	MASE	RMSSE
Texas (ID: 252299)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	1.19	1.00
		XGBoost_GoogleTrends	1.12	0.99
		LSTM	1.00	0.85
		LSTM_GoogleTrends	1.07	0.86
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.06	0.89
		XGBoost_GoogleTrends	1.02	0.78
		LSTM	0.77	0.67
		LSTM_GoogleTrends	0.88	0.67
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.12	1.09
		XGBoost_GoogleTrends	1.04	0.99
		LSTM	1.04	0.90
		LSTM_GoogleTrends	1.11	1.02
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.27	1.08
		XGBoost_GoogleTrends	1.33	1.14
		LSTM	0.85	0.74
		LSTM_GoogleTrends	0.83	0.73

On some product-store combinations like Honey Nut Cheerios sales in Kentucky, MASE and RMSEE is below 1 for the XGBoost and LSTM models that contain Google Trends and the score is lower than the MASE and RMSSE computed for XGBoost and LSTM models that do not use Google Trends. Whereas for other product-store combinations, to exemplify, Digiorno Pepperoni Pizza sales in Texas, both XGBoost and XGBoost_GoogleTrends' MASE and RMSEE score is above 1.

Furthermore, a paired t-test is conducted, to validate weather predictions using Google Trends are statistically different and significant than predictions made using only historical sales. For each model, we pair predictions by the 12 possible product-store combinations. The p-value computed for the XGBoost and XGBoost_GoogleTrends pair is 0.869919. Based on a threshold p-value of 0.05, the paired t-test suggests that there is no statistically significant difference between the predictions of the XGBoost model that contains only historical sales as input with the XGBoost predictions that use Google Trends as data input. Therefore, the null hypothesis stating that XGBoost predictions using historical sales as data input are statistically identical to XGBoost predictions that use historical sales and Google Trends is not rejected. Similarly, the p-value computed for the LSTM and LSTM_GoogleTrends pair is 0.031197, which is below the threshold value of 0.05.

To supplement the findings, we extend the experiment on the Breakfast at the Frat dataset to compare the performance the XGBoost and LSTM models that contain historical sales as well as additional transactional data as input with XGBoost and LSTM models that use historical sales, additional transactional data, and Google Trends. Table 12, describes the additional transactional attributes used.

Table 12: Breakfast at the Frat – Additional Transactional Data

Attribute	Attribute Description
1	Count of weekly household store visits.
2	Number of households purchasing an item on a given week.
3	Total amount of dollars spent by the household during a store visit.
4	Base price for an item.
5	Actual amount for an item charged at the point of sale.
6	Flag indicating weather a product is featured in store circulation on a given week.
7	Flag indicating weather an item is part of an in-store promotional display on a given week.
8	Flag indicating temporary price reduction only on the shelf tag and not promotional advertisement on a given week.

Table 13, contrasts the performance of XGBoost and LSTM models that use historical sales and additional transactional data to generate predictions with the XGBoost and LSTM models that use historical sales, transactional and Google Trends data. Results in Table 13, suggest that generally, performance improves on RMSE, WAPE and R^2 for the XGBoost model using Google Trends over the XGBoost not using Google Trends Contrarily, the LSTM model that does not use Google Trends seems to be scoring better on MAPE, RMSE and R^2 over the LSTM model using Google Trends to make predictions.

Table 13: Breakfast at the Frat – Results for Experiments 3 and 4

Data Input	Model	Average MAPE	Average RMSE	Average WAPE	Average R^2
Historical Sales and Additional Transactional Data	XGBoost	0.41	58.13	4136.07	-0.79
Historical Sales, Additional Transactional Data and Google Trends	XGBoost_GoogleTrends	0.42	56.12	3845.18	-0.71
Historical Sales and Additional Transactional Data	LSTM	0.35	48.24	3768.98	-0.11
Historical Sales, Additional Transactional Data and Google Trends	LSTM_GoogleTrends	0.38	49.56	3752.48	-0.20

Prominently, summary results presented in Table 13 are identical to the results discussed earlier in Table 11 that compares performance changes for XGBoost and LSTM as a result of using Google Trends and historical sales to make predictions. In short, using Google Trends with real-world data seems to be improving XGBoost predictions for the majority of 12 possible product-store combinations while LSTM predictions are more accurate when Google Trends data is not used. Particularly, Table 14, shows the performance for selected product-store combinations where XGBoost that uses historical sales, additional transactional data and Google Trends is favorable over the XGBoost model that only uses historical sales and additional transactional data as input. Prediction accuracy for private label pretzel sales improves for all 3 stores. Similarly, accuracy for Kellogg's Frosted Flakes improves across 2 stores and prediction accuracy for Honey Nut Cheerios sales in Ohio also improves. Earlier as shown in Table 21, prediction accuracy for XGBoost that uses Google Trends with historical sales to make predictions, also improves for private label pretzel sales in all 3 stores. The consistency in specific product-store performance improvements when XGBoost uses historical sales and Google Trends vis-à-vis XGBoost that uses historical sales, additional transactional data, and Google Trends, implies a positive influence of Google Trends on prediction accuracy.

Table 14: Breakfast at the Frat Predictions Using Historical Sales, Transactional Data & Google Trends – XGBoost

Store	Product	Model	MAPE	RMSE	WAPE	R^2
Ohio (ID: 2277)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	0.30	102.98	12215.79	-0.23
		XGBoost_Google Trends	0.23	99.74	11948.25	-0.16
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	0.28	66.14	7665.64	-1.45
		XGBoost_Google Trends	0.24	56.22	6702.26	-0.77
Kentucky (ID: 389)	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	0.48	77.00	5002.80	-0.35
		XGBoost_Google Trends	0.35	70.92	4821.97	-0.15
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	0.23	30.17	1598.08	-1.35
		XGBoost_Google Trends	0.20	28.67	1596.30	-1.12
Texas (ID: 252299)	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	0.39	54.16	2251.84	-0.90
		XGBoost_Google Trends	0.32	52.30	2087.60	-0.77
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	0.39	22.60	593.60	-1.13
		XGBoost_Google Trends	0.32	21.15	548.33	-0.86

Lastly, using the MASE and RMSSE metrics we compare prediction performance of the XGBoost and LSTM models that use historic sales and additional transactional data with the XGBoost and LSTM models that use historic sales, additional transactional data, and Google Trends as data input. Table 15 contrasts the MASE and RMSSE scores of the models that contain and do not contain Google Trends.

Table 15: Breakfast at the Frat – Experiments 3 & 4 Results by MASE & RMSSE

Store	Product	Model	MASE	RMSSE
Ohio (ID: 2277)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	1.13	0.85
		XGBoost_GoogleTrends	0.96	0.83
		LSTM	0.86	0.81
		LSTM_GoogleTrends	0.92	0.82
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.03	0.85
		XGBoost_GoogleTrends	1.20	0.96
		LSTM	0.76	0.72
		LSTM_GoogleTrends	0.85	0.74
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.73	1.66
		XGBoost_GoogleTrends	1.48	1.41
		LSTM	1.38	1.30
		LSTM_GoogleTrends	1.26	1.18
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.40	1.24
		XGBoost_GoogleTrends	1.40	1.25
		LSTM	1.00	0.89
		LSTM_GoogleTrends	1.08	0.92
Kentucky (ID: 389)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	0.79	0.66
		XGBoost_GoogleTrends	0.83	0.64
		LSTM	0.94	0.65
		LSTM_GoogleTrends	0.89	0.63
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.08	1.07
		XGBoost_GoogleTrends	0.90	0.98
		LSTM	0.92	0.91
		LSTM_GoogleTrends	1.06	1.04
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.30	1.29
		XGBoost_GoogleTrends	1.14	1.23
		LSTM	0.93	0.89
		LSTM_GoogleTrends	0.97	1.02
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.22	1.15
		XGBoost_GoogleTrends	1.36	1.23
		LSTM	1.10	0.95
		LSTM_GoogleTrends	1.17	1.07

Store	Product	Model	MASE	RMSSE
Texas (ID: 252299)	Honey Nut Cheerios (UPC: 1600027527)	XGBoost	1.18	1.09
		XGBoost_GoogleTrends	1.21	0.97
		LSTM	0.98	0.86
		LSTM_GoogleTrends	1.19	0.88
	Kellogg's Frosted Flakes (UPC: 3800031838)	XGBoost	1.12	0.88
		XGBoost_GoogleTrends	0.95	0.85
		LSTM	0.71	0.65
		LSTM_GoogleTrends	0.82	0.64
	Private Label Mini Twist Pretzels (UPC: 1111009477)	XGBoost	1.37	1.21
		XGBoost_GoogleTrends	1.14	1.13
		LSTM	0.96	0.86
		LSTM_GoogleTrends	0.99	0.89
	Digiorno Pepperoni Pizza (UPC: 7192100339)	XGBoost	1.22	1.04
		XGBoost_GoogleTrends	1.32	1.12
		LSTM	0.85	0.73
		LSTM_GoogleTrends	0.96	0.80

Results in Table 15 are manifold with MASE and RMSSEE scores that are above 1 and below 1 for both XGBoost and LSTM models, depending on the product and store for which predictions are generated for. Accordingly, a paired t-test is performed on XGBoost and LSTM respectively, that use historical sales, additional transactional data, and Google Trends. The null hypothesis is that the predictions from the XGBoost and LSTM models that use sales history and additional transactional data to make predictions are statistically identical to the XGBoost and LSTM model predictions that use historic sales, additional transactional data, and Google Trends to generate the predictions. The computed p-value for the XGBoost and XGBoost_GoogleTrends predictions is 0.271920. Similarly, the p-value computed for the LSTM and LSTM_GoogleTrends predictions is 0.119649. Hence, using a threshold p-value of 0.05, the null hypothesis is not rejected since the p-values are larger than the threshold.

Results and Findings Summary

The experiment results are summarized with Table 16. Recapitulating the experiment results investigating the predictive power of Google Trends in retail sales forecasting and applied on the Brazilian e-commerce and Breakfast at the Frat datasets, overall, we do not find strong evidence indicating statistically significant differences between predictions generated by models that use only real-world data and predictions generated by models that use real-world data and Google Trends. In addition, results in Table 16, suggest that for experiment 1 and when using the Brazilian e-commerce dataset, FBProphet is the best performing model as measured by MAPE, RMSE, WAPE and R^2. Whereas, in experiment 1 and when utilizing the Breakfast at the Frat dataset, the LSTM model outperforms all other models as measured by MAPE, RMSE and R^2.

Table 16: Experiment Summary Results

Experiment 1					
Data Set	**Model**	**Average MAPE**	**Average RMSE**	**Average WAPE**	**Average R^2**
Brazilian e-commerce	SARIMA	0.29	44.46	3417.66	-0.56
	FBProphet	0.25	40.39	3319.33	-0.29
	XGBoost	0.34	51.99	4669.33	-1.15
	LSTM	0.33	53.75	5412.82	-1.17
Breakfast at the Frat	SARIMA	0.49	52.67	3758.13	-0.31
	FBProphet	0.54	52.93	3397.56	-0.35
	XGBoost	0.41	57.14	4093.34	-0.70
	LSTM	0.36	48.00	3696.69	-0.11

Experiment 2					
Data Set	**Model**	**Average MAPE**	**Average RMSE**	**Average WAPE**	**Average R^2**
Brazilian e-commerce	XGBoost	0.34	51.99	4669.33	-1.15
	XGBoost_GoogleTrends	0.36	59.08	5282.20	-1.71
	LSTM	0.34	54.12	5450.81	-1.22
	LSTM_GoogleTrends	0.36	57.21	5888.69	-1.48
Breakfast at the Frat	XGBoost	0.41	57.14	4093.34	-0.70
	XGBoost_GoogleTrends	0.45	57.88	3951.95	-0.67
	LSTM	0.36	48.00	3696.69	-0.11
	LSTM_GoogleTrends	0.39	49.87	3811.68	-0.21

Experiments 3 and 4					
Data Set	**Model**	**Average MAPE**	**Average RMSE**	**Average WAPE**	**Average R^2**
Breakfast at the Frat	XGBoost	0.41	58.13	4136.07	-0.79
	XGBoost_GoogleTrends	0.42	56.12	3845.18	-0.71
	LSTM	0.35	48.24	3768.98	-0.11
	LSTM_GoogleTrends	0.38	49.56	3752.48	-0.20

Furthermore, comparing the forecasting accuracy of the models considered (SARIMA, FBProphet, XGBoost and LSTM) using MAPE, RMSE, WAPE and R^2, we observe scattered performance of the models relative to each other depending on the metric used to measure accuracy, since each metric penalizes error differently. Therefore, we benchmark all models considered against a naïve model and measure the relative accuracy of models using the MASE and RMSSE metrics. MASE and RMSSE are considered as general measures of forecasting accuracy as these metrics are scale independent and penalize error in a more symmetric fashion.

Using the Brazilian e-commerce dataset, we find that only for certain product categories out of the 7 for which predictions are generated for, the forecasting models' score on MASE and RMSSE is less than 1 i.e., better than a naïve model. This implies that the training data used is difficult to learn and potentially more years of history can help improve performance. Nonetheless, using the Brazilian e-commerce dataset to predict the sales of telephony and furniture décor product categories, we find a lower MASE and RMSSE score for the XGBoost and LSTM models that use Google Trends relative to the XGBoost and LSTM models that do not use Google Trends. This hints on the potential usefulness of the Google Trends data collected for forecasting the sales of these product categories. The Google Trends data used for the Brazilian e-commerce dataset is collected based on assumptions on the products sold by Olist on digital marketplaces like Submarino and Mercado Livre among others.

Moreover, using the Breakfast at the Frat dataset, sales predictions are generated for 4 products in 3 stores. Generally, for Breakfast at the dataset, the MASE and RMSSE scores for the models considered are below 1, suggesting favorable performance over a naïve model. When measuring performance using MAPE, RMSE, WAPE and R^2, similar to the findings from the Brazilian e-commerce dataset, performance of the models considered varies depending on the metric and product-store combination for which predictions are generated for. Also, for a specific product and store, predictions made by one model may be the most favorable while another model scores relatively better for predicting the sales of the same product at another store. This may be due to the differences in consumption patterns across stores located in different geographies and other factors such as store specific promotions. The Google Trends data collected for the Breakfast at the Frat dataset is primarily based on the brand name of the products, the manufacturer of the product, competitors and searches believed to capture consumer preferences.

Furthermore, for certain product-store combinations out of the 12 that each model makes predictions for, we find lower MASE and RMSSE scores for the XGBoost and LSTM models that use Google Trends relative to the XGBoost and LSTM models that do not use Google Trends. For instance, when predicting the sales of mini twist pretzels across the 3 stores in scope, the forecasting accuracy as measured by MAPE, RMSE, WAPE, R^2,

MASE and RMSSE is higher when XGBoost uses Google Trends. In addition, when we analyse the feature importance plot for the XGBoost model that uses Google Trends, we observe that the model identifies Google Trends searches related to the product sold, manufacturer, competitor and other general terms among the top features used in splits. This suggests that Google searches on product names may be treated as viable information in forecasting the sales of grocery products, resembling the findings Boone et al. (2018). Also, results from this experiment suggest that Google searches related to coupons and competitor retailers are among the top features used by the XGBoost model in predicting grocery sales for the Breakfast at the Frat dataset. In addition, for a specific search term, depending on the store location for which predictions are generated, the XGBoost model seems to be identifying Google Trends series collected at the state level as more relevant than the Google Trends series collected at a country level. In contrast, due to the nature of the LSTM model it is not possible to analyse which Google Trends series the model deems more relevant to make predictions. Additionally, due to the way the LSTM model is stacked, the number of lag variables created for each Google Trends series is the same as the number of lags created for the target variable, that is 52. Whereas with the XGBoost model we create 52 lag variables for the target variable and 12 lag variables for each Google Trends series used. Accordingly, although the same Google Trends series is used in the XGBoost and LSTM models, the differences in the lengths of lags used for the Google Trends series in each model is an important factor to consider during the results interpretation.

Overall, FBProphet is among the most practical models to use, requiring a few code lines and achieves decent results with default settings. Although SARIMA is relatively simpler to use than XGBoost and LSTM, overall SARIMA predictions on the Brazilian e-commerce and Breakfast at the Frat datasets are less favorable relative to FBProphet. Nonetheless, the SARIMA model tends move along the fluctuations and seasonality in actual sales better when one-step ahead forecasts are made as opposed to multi-step. The LSTM model generally predicts the mean value in sales while XGBoost fluctuates more often from the mean relative to LSTM. XGBoost and LSTM contain multiple hyperparameters, required to be specified, making the use of these models more complex relative to SARIMA and FBProphet. Additionally, for LSTM, a portion of the training

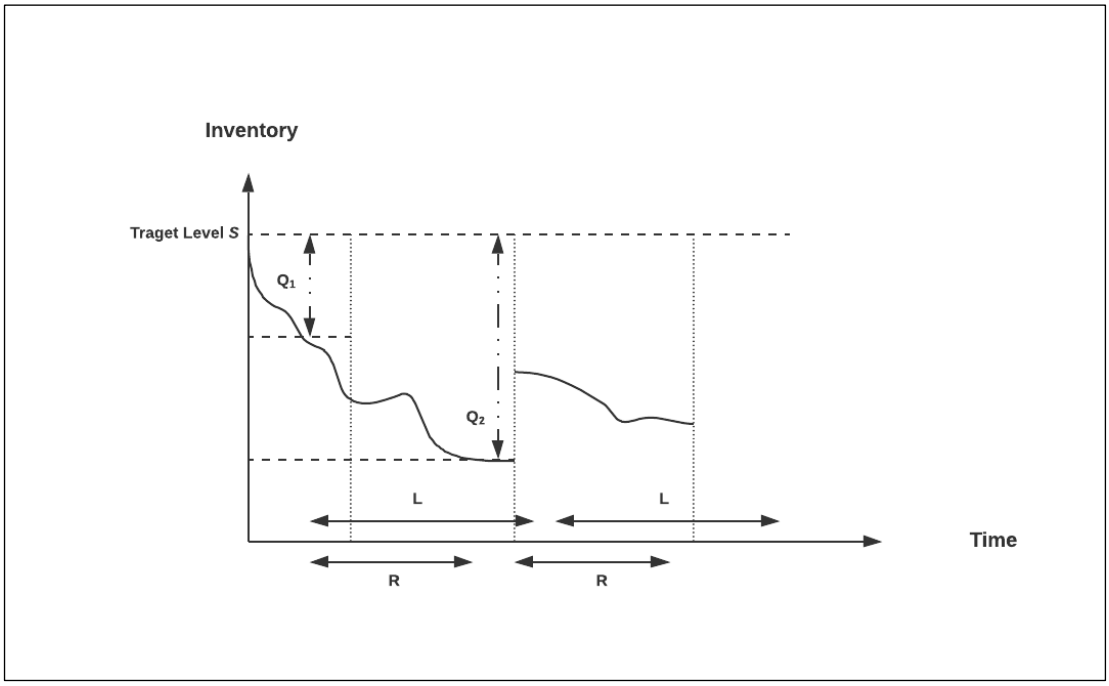
data is reserved for hyperparameter tunning as conducted for early stopping. Lastly, all models are run using a machine with an Intel core i5-7200U processor. Generally, SARIMA, FBProphet and XGBoost models run within minutes or hours for all predictions generated on the Brazilian e-commerce and Breakfast at the Frat datasets. In contrast, depending on the number of times search iterations are conducted during hyperparameter tunning for the LSTM model, the time it takes to run a LSTM model may increase from hours to days. Additionally, there is no standard or best practice on how to set the structure of the LSTM model in terms of defining the number of hidden layers and units within each layer. Given the size of the real-world datasets we first run a single layer LSTM model with 10 hidden units. However, as we experiment with more deep LSTM models, minor performance improvements are observed. Hence, we follow a decrease strategy for defining the LSTM structure. To exemplify, for a three-layer LSTM model with the initial number of units set to 50, the decrease strategy will set 50 for the first layer, 25 for the second layer and 16 for the third layer. However, running the LSTM model is computationally more intensive than running SARIMA, FBProphet and XGBoost.

The inventory performance implications of forecasting errors in inventory management process can be evaluated using a simulation. In the following section, using the sales predictions from SARIMA, FBProphet, XGBoost and LSTM on the Breakfast at the Frat dataset, we interpret prediction accuracy in terms of supply chain performance by conducting an inventory management simulation.

5.2 Inventory Management Simulation

In this section, the impact of demand uncertainty on inventory turnover performance is demonstrated through simulation. The RMSE from predictions generated by SARIMA, FBProphet, XGBoost and LSTM is used to approximate demand uncertainty. In particular, we simulate inventory replenishment cycles for Honey Nut Cheerios (cold cereal) at the store located in Kentucky and Digiorno Pepperoni Pizza (frozen pizza) at the store located in Ohio. Using a periodic review *(R,S)* policy we simulate inventory control decisions over 48 weeks. Figure 27 illustrates the *(R,S)* periodic review system.

Figure 27: Periodic Review *(R,S)*



Predictions for the first four weeks of January 2011, are excluded from the simulation as we use the RMSE of the first four weeks' forecasts to represent uncertainty during the period of risk. During each review period *R*, the target level *S* is computed dynamically as discussed in section 4.1 Methodology. A service level of 95% is used to compute the safety stock. In other words, the inventory control policy is set such that there is 5% or less chance of stockouts occurring. The calculation of safety stock is dependent on the service level, period of risk and forecast error as measured by RMSE. Therefore, despite

using the same service level to compute the safety stock using the forecast error from each model, the expectation is that the safety stock levels determined varies across models. During each review period, the order size Q is computed by deducting the inventory on hand and scheduled receipts from the target level determined on a review period. Demand not satisfied within a time period is assumed to be lost and no back-orders are considered. We simulate two scenarios for each product-store combination. The first scenario is for a review period $R=2$ *weeks* and a lead time $L=3$ *weeks* while the second scenario is for a review period $R=3$ *weeks* and a lead time $L=4$ weeks. This setup facilitates to explore the outcomes of the inventory control policy as the period of risk of is extended. The period of risk during a replenishment cycle of a periodic review system is equal to the duration of $R + L$. Typically, safety stock levels associated with periodic review policy are higher than a continuous review policy since under a continuous review policy the period of risk covers the lead time L only as opposed to $R + L$.

Table 17, presents simulation results for Honey Nut Cheerios at the Kentucky store. We assume a beginning inventory of 630 units. The selling price of Honey Nut Cheerios is assumed to be 5$. The simulation starts on 2011-01-29 and ends on 2011-12-31. Furthermore, the average inventory ratio is computed by dividing total net sales during the simulation period by the average inventory at the selling price. A low turnover rate implies overstocking, obsolesce or deficiencies in operations. Conversely a high inventory turnover rate can be interpreted as understocking, potentially leading to lower customer satisfaction. When the review period is set to 2 weeks and the lead time is 3 weeks, results in Table 17 suggest that using FBProphet predictions, 19 orders are placed throughout the simulation and the inventory turnover ratio is the lowest compared to the rest of the models. This means that average inventory levels are the highest when using FBProphet compared to the rest of the models. In contrast, the XGBoost model scores the highest on the average inventory ratio, implying lower average levels of inventory kept throughout the simulation. However, more orders are placed during the simulation when using XGBoost and LSTM predictions relative to FBProphet and SARIMA. Besides the fixed costs associated with placing an order, in a retail environment with hundreds, possibly, thousands of products that may be available at a store, very frequent review of inventory may be challenging to follow in practice.

Table 17: Simulation Results – Honey Nut Cheerios, Kentucky

Review Period (weeks)	Lead Time (weeks)	Model	Fill Rate (%)	Safety Stock (units)	Number of Orders	Average Inventory Level (Units)	Average Inventory Turnover Ratio
2	3	SARIMA	100	303	22	429	10.25
		FBProphet	100	188	19	489	8.99
		XGBoost	100	34	21	154	28.57
		LSTM	100	101	22	317	13.88
		XGBoost_GoogleTrends	100	122	20	235	18.77
		LSTM_GoogleTrends	100	56	22	220	20.02
3	4	SARIMA	100	358	22	480	9.15
		FBProphet	100	222	19	521	8.43
		XGBoost	100	40	21	159	27.59
		LSTM	100	119	22	334	13.16
		XGBoost_GoogleTrends	100	145	20	256	17.18
		LSTM_GoogleTrends	100	67	22	229	19.18

Looking at the safety stock levels in Table 17, the XGBoost and LSTM models are associated with lower safety stocks relative to SARIMA and FBProphet. This is likely due to the more favorable performance of the XGBoost and LSTM models relative to SARIMA and FBProphet in terms of RMSE used to approximate uncertainty in demand. Accordingly, based on the simulation results, lower errors in forecasting error led to more favorable inventory turnover performance. Overall, as the length of the period of risk increases, the average inventory ratio for most of the models tends to decrease. Furthermore, no stockouts are encountered and 100% fill rate is achieved for all product-store combinations considered in the simulation.

Table 18, presents simulation results for Digiorno Pepperoni Pizza at the Ohio store. We assume a beginning inventory of 500 units. The sales price of Digiorno Pepperoni Pizza is assumed to be 9$. Demand not satisfied within a time period is assumed to be lost and no back-orders are considered. The simulation starts on 2011-01-29 and ends on 2011-12-31. Similar to simulation results for Honey Nut Cheerios, XGBoost seems to be the best performer in terms of the average inventory ratio. Using XGBoost predictions, frequent, small size orders are placed keeping inventory levels fairly consistent. However, the inclusion of Google Trends in the XGBoost and LSTM models does not seem to be following a consistent pattern of impact on the average inventory turnover ratio. This supports the findings of the paired t-test conducted earlier. The results of the paired t-test suggest no strong evidence on statistically significant differences between predictions generated from models that do not use Google Trends and predictions that use Google Trends. Furthermore, no stockouts are encountered and the fill rate is 100% for Digiorno Pepperoni pizza under the scenarios considered during the simulation.

Table 18: Simulation Results – Digiorno Pepperoni Pizza, Ohio

Review Period (weeks)	Lead Time (weeks)	Model	Fill Rate (%)	Safety Stock (units)	Number of Orders	Average Inventory Level (Units)	Average Inventory Turnover Ratio
2	3	SARIMA	100	150	20	261	11.23
		FBProphet	100	141	20	246	11.92
		XGBoost	100	53	20	112	26.18
		LSTM	100	97	21	197	14.88
		XGBoost_GoogleTrends	100	79	18	182	16.11
		LSTM_GoogleTrends	100	160	21	203	14.44
3	4	SARIMA	100	178	21	286	10.25
		FBProphet	100	167	20	268	10.94
		XGBoost	100	63	20	135	21.75
		LSTM	100	115	21	214	13.70
		XGBoost_GoogleTrends	100	94	17	196	14.96
		LSTM_GoogleTrends	100	189	22	230	12.75

To summarize the simulation, we select two products from different categories that posses different demand behaviour. We simulate replenishment decisions over a period of 48 weeks with varying review periods and lead times. Generally, throughout the simulation for both products, at a service level of 95%, using the periodic review *(R,S)* policy no stockouts are encountered. However, stockouts may occur in practice and other variations of service levels, review periods and lead times can be simulated to explore possible outcomes. Furthermore, as the period of risk is extended, generally safety stocks and average inventory levels tend to be higher. For the selected products, the resulting inventory turnover ratio from the simulation is more favorable when using the XGBoost and LSTM models. This may be partly due to XGBoost and LSTM making predictions that are relatively closer to the mean value of sales in comparison to SARIMA and FBProphet. In contrast, using FBProphet and SARIMA there are less orders placed throughout the simulation though with larger order sizes.

The inventory management simulation results imply that lower forecasting error is associated with more favorable inventory turnover performance. Accordingly, this inventory management simulation can be used to evaluate inventory performance based on different forecasting models. Generally, models that make demand predictions that are closer to the mean and with lower forecast error may have a positive impact on inventory performance.

Chapter 6

6.1 Limitations

The comparative experiment is limited in scope and is based on a set of assumptions to facilitate modelling.

Furthermore, we collect Google Trends data related to product descriptions, manufacturers, suppliers, competitors, and search terms that capture consumer tastes such as searches made on “low carb diet” or “grocery coupons.” All Google Trends specific to a product category and trends that are used across product categories are fed as input in the XGBoost and LSTM models to forecast the sales of a product category or specific product. Accordingly, the experiment can be modified to explore which Google Trends data are relevant predictors of the target variable using methods like Lasso regression, prior to using Google Trends in forecasting models. As a result, expectations on performance improvements from using Google Trends are likely to be based on more concrete foundations. Lastly, it is likely that the amount of historical data used to train models is insufficient for the Brazilian e-commerce dataset since performance scores of all models are generally worst than a naïve model. This highlights the importance of using sufficient amounts of training samples and benchmarking performance to basic models such as the naïve approach.

6.2 Conclusion and Future Work

To conclude, this thesis investigates the predictive power of Google Trends in retails sales forecasting. A comparative experiment is applied on two real-world datasets, (1) Brazilian e-commerce and (2) Breakfast at the Frat. The time-series methods SARIMA and FBProphet as well as the machine learning models XGBoost and LSTM are used to make predictions. For the Brazilian e-commerce dataset, the weekly number of sales transactions for the top selling 7 product categories is predicted for 32 weeks using time-series cross validation. Results on the Brazilian e-commerce dataset suggests that all models seem to be performing poorly when benchmarked against a naïve model. This implies that the training data is difficult to learn from. Although we attempt to use the most amount of data available to train the models, a larger length of training data is likely to help in improving performance. In contrast, for Breakfast at the Frat dataset, the prediction task is to forecast the weekly number of units sold of 4 products in 3 stores. Using the Breakfast at the Frat dataset, we test predictions on 52 weeks. On some product-store combinations the performance of all models is worse than a naïve model while performance of models on the majority of product-store combinations is better than a naïve model. This suggests that sales behavior for some products sold at some stores is relatively more difficult to learn than other product-store combinations with less evident trend and seasonality patterns.

The null hypothesis is that predictions generated by models using only real-world data as data input and predictions generated by models using real-world data and Google Trends as data input, are not statistically different and belong to the same statistical distribution. Using XGBoost and LSTM we integrate real-world data with Google Trends to make predictions and compare the performance of XGBoost and LSTM that do not use Google Trends with XGBoost and LSTM that use Google Trends. The framework used to collect and integrate Google Trends for the Brazilian e-commerce and Breakfast at the Frat datasets is provided. Nonetheless, the results from the paired t-test performed to test the null hypothesis suggest that there is no statistically significant evidence to reject the null hypothesis. Based on the experiment results we find that predictions by models that use real-world data and models that use real-world data and Google Trends, are not

statistically identical. Further investigation is required to explore the predictive power of Google Trends since the results form this experiment suggest that there is no statistically significant average difference in making predictions with and without using Google Trends.

Despite the results of the paired t-test, on some occasions, forecasting the sales of a specific product at specific store, to exemplify mini twist pretzels sales in Kentucky, Ohio, and Texas, we find the models that contain Google Trends performing better than the models that do not contain Google Trends. The improvement of forecasting accuracy on mini twist pretzels across all three stores as a result of using Google Trends hints a positive a signal on the usefulness of Google Trends. This could be related to the keywords used for mini twist pretzels. M Furthermore, the experiment compliments previous studies by Boone et. al (2018) and Silva et. al (2019) that primarily use descriptions of products as the Google Trends search terms used as predictors for the target variable, by including search terms that are broader, related to consumer preferences, competitors of the product, manufacturers, and searches on promotions. Accordingly, the XGBoost feature importance plot suggests that broad search terms like “grocery coupons” are among the top features used in tree splits for multiple products. In addition, depending on the store location for which the forecast is made for, using the XGBoost feature importance plot we identify how local search terms such as “PepsiCo” searches in Ohio are used in more tree splits than global searches like “PepsiCo” searches in the U.S.

Lastly, we demonstrate the impact of forecasting error on inventory turnover performance through a simulation. In particular, using a periodic inventory control system *(R,S)*, we simulate inventory replenishment cycles over 48 weeks for Honey Nut Cheerios sold at the Kentucky store and Digiorno Pepperoni Pizza sold at the store located in Ohio. The key performance indicator used to interpret inventory performance is the inventory turnover ratio. Overall, for the selected products, the XGBoost and LSTM models tend to yield the highest inventory turnover ratio relative to SARIMA and FBProphet. This means that average inventory levels are lower when the predictions from XGBoost and LSTM are used during the simulation. The service level is set to 95% and under the two scenarios for which the simulation is conducted, no stockouts are found. The first scenario of the

simulation is set such that the review period R is equal to 2 weeks and the lead time L is 3 weeks. In the second simulation scenario the review period R is set to 3 weeks and the lead time L is set to 4 weeks. Generally, as the period of risk extends, we observe higher safety stock levels and lower inventory turnover ratios, implying higher average inventory levels. Additionally, based on simulation results, when the forecasting error (RMSE in this experiment) used as a proxy to represent uncertainty in demand is lower, the inventory turnover ratio is likely to be higher (favorable). In short, findings suggest that besides measuring accuracy of a forecasting model using various metrics such as MAPE, RMSE, WAPE among others, inventory management simulation helps to understand the associated inventory control performance associated with predictions and uncertainty in the predictions of each model. As a result, decisions on which forecasting model's predictions to use are based on more extensive what-if analysis that includes organizational objectives and key performance indicators.

On the whole, research on timeseries forecasting has been studied over the decades with emerging publications focusing on the use of machine learning methods. Furthermore, Google Trends data continues to gain popularity as a source of external information that can be used in prediction tasks. Based on the results of the comparative experiment it is evident that there is room for more work exploring the predictive power of Google Trends in retail sales forecasting. The source code of the experiment is made available to the public and can be adapted for future projects. The experiment applied on the two real-world datasets is tailored towards a prediction task related to the sales of a specific product category or specific product at a specific store. Hence, possible future work may investigate forecasting at different levels of aggregation, ex: at a store or market unit level. Additionally, the use of Google Trends can be explored in predicting the sales of upstream supply chain partners vs. downstream. To exemplify, a purchase decision of manufacturing equipment required to package frozen food products is a more complex purchasing process, with high level of research and involvement by the buyer vs. an end customer shopping for grocery at a retail store. Moreover, studying the qualitative value of Google Trends in sales forecasting can be investigated by conducting field work with practitioners and executives.

Bibliography

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews,* 29(5-6), 594–621. https://doi.org/10.1080/07474938.2010.481556

Agin, N. (1966). A min-max inventory model. *Management Science*, 12(7), 517–529.

Akaike, H. (1974). A new look at the statistical model identification. Ieee *Transactions on Automatic Control*, 19(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

Ali, Z., & Bhaskar, S. B. (2016). Basic statistical tools in research and data analysis. *Indian Journal of Anaesthesia*, 60(9), 662–669. https://doi.org/10.4103/0019-5049.190623

Alon, I., Qi, M., & Sadowski, R. J. (2001). Forecasting aggregate retail sales:. *Journal of Retailing and Consumer Services, 8*(3), 147-156. doi:10.1016/s0969-6989(00)00011-4

Aly Al-Amyn Valliani, Ranti, D., & Oermann, E. K. (2019). Deep learning and neurology: A systematic review. *Neurology and Therapy*, , 1-15. doi:http://dx.doi.org.proxy2.hec.ca/10.1007/s40120-019-00153-8

Au, K.-F., Choi, T.-M., & Yu, Y. (2008). Fashion retail forecasting by evolutionary neural networks. *International Journal of Production Economics, 114(2)*, 615–630. https://doi.org/10.1016/j.ijpe.2007.06.013

Askitas, N., & Zimmermann, K. F. (2009). Google econometrics and unemployment forecasting. *Applied Economics Quarterly, 55(2)*, 107–120. https://doi.org/10.3790/aeq.55.2.107

Axsäter S. (2015) Forecasting. In: Inventory Control. International Series in Operations Research & Management Science, vol 225. *Springer*, Cham. https://doi-org.proxy2.hec.ca/10.1007/978-3-319-15729-0_2

Bandara, K., Bergmeir, C., Hewamalage, H., Shi, P., Tran, Q., Seaman, B., & 26th International Conference on Neural Information Processing, ICONIP 2019 26th 2019 12 12 - 2019 12 15. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11955 Lncs, 462–474. https://doi.org/10.1007/978-3-030-36718-3_39

Barrow, D. K., & Kourentzes, N. (2016). Distributions of forecasting errors of forecast combinations: implications for inventory management. *International Journal of Production Economics*, 177, 24–33. https://doi.org/10.1016/j.ijpe.2016.03.017

Baraniak, K. (2018). ISMIS 2017 Data Mining Competition: Trading Based on Recommendations - XGBoost Approach with Feature Engineering. *Studies in Big Data Intelligent Methods and Big Data in Industrial Applications,* 145-154. doi:10.1007/978-3-319-77604-0_11

Bańbura, Marta; Giannone, Domenico; Reichlin, Lucrezia (2010) : Nowcasting, ECB Working Paper, No. 1275, European Central Bank (ECB), Frankfurt a. M

Bayraktar, E., Lenny, K. S. C., Gunasekaran, A., Sari, K., & Tatoglu, E. (2008). The role of forecasting on bullwhip effect for e-scm applications. *International Journal of Production Economics*, 113(1), 193–204. https://doi.org/10.1016/j.ijpe.2007.03.024

Baumol, W. J., & Ide, E. A. (1956). Variety in retailing. *Management Science*, 3(1), 93–101.

BBC News. (2020, October 16). Covid: What are the lockdown rules in place across Europe? https://www.bbc.com/news/explainers-53640249

Behera, G., & Nain, N. (2019, September). A comparative study of big mart sales prediction. In International Conference on Computer Vision and Image Processing (pp. 421-432). Springer, Singapore.

Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. To appear in Proc. of the 30th International Conference on Machine Learning (ICML 2013).

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, 120, 70–83. https://doi.org/10.1016/j.csda.2017.11.003

Boone, T. (2016, August). Fashion trends 2016: Google Data shows what shoppers want. Think with Google. https://www.thinkwithgoogle.com/advertising-channels/search/fashion-trends-2016-google-data-consumer-insights/

Boone, T., Ganeshan, R., & Hicks, R. L. (2015). Incorporating Google trends data into sales forecasting. Foresight: *The International Journal of Applied Forecasting*, (38), 9-14.

Boone, T., Ganeshan, R., Hicks, R. L., & Sanders, N. R. (2018). Can google trends improve your sales forecast? *Production and Operations Management*, 27(10), 1770–1774. https://doi.org/10.1111/poms.12839

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2008). Time series analysis : forecasting and control (Fourth, Ser. Wiley series in probability and statistics). J. Wiley & Sons.

Bureau van Dijk. (2020). [KROGER CO, key information]. https://orbis-bvdinfo-com.proxy2.hec.ca/version-2020115/orbis/1/Companies/report/Index?format=_standard&BookSection=PROFILE&seq=0

Carrière-Swallow Yan, & Labbé Felipe. (2013). Nowcasting with google trends in an emerging market. *Journal of Forecasting*, 32(4), 289–298. https://doi.org/10.1002/for.1252

Cadavid, J. P. U., Lamouri, S., & Grabot, B. (2018, July). Trends in Machine Learning Applied to Demand & Sales Forecasting: A Review. International Conference on Information Systems, Logistics and Supply Chain, Lyon, France. ffhal-01881362f

Castillo, P. A., Mora, A. M., Faris, H., Merelo, J. J., García-Sánchez, P., Fernández-Ares, A. J., … García-Arenas, M. I. (2017). Applying computational intelligence methods for predicting the sales of newly published books in a real editorial business management environment. *Knowledge-Based Systems*, 115, 133–151. https://doi.org/10.1016/j.knosys.2016.10.019

Chauhan, N. S. (2020, January). *Decision Tree Algorithm, Explained*. KDnuggets. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

Chang, P.-C., Wang, Y.-W., & Liu, C.-H. (2007). The development of a weighted evolving fuzzy neural network for pcb sales forecasting. Expert Systems with Applications, 32(1), 86–96. https://doi.org/10.1016/j.eswa.2005.11.021

Chang, X., Gao, M., Wang, Y., & Hou, X. (2012). Seasonal autoregressive integrated moving average model for precipitation time series. *Journal of Mathematics & Statistics*, 8(4).

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. doi:10.1145/2939672.2939785

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Choi, H., & Varian, H. (2012). Predicting the present with google trends. Economic Record, 88(Suppl.1), 2–9. https://doi.org/10.1111/j.1475-4932.2012.00809.x

Choi, J. Y., & Lee, B. (2018). Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting. Mathematical Problems in Engineering, 2018, 1–8. https://doi.org/10.1155/2018/2470171

Chopra, S., Meindl, P., & Kalra, D. V. (2013). Supply chain management: strategy, planning, and operation (Vol. 232). Boston, MA: Pearson.

Chu, C., & Zhang, G. P. (2003). A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, 86(3), 217-231. doi:10.1016/s0925-5273(03)00068-9

Clark, A. J., & Scarf, H. (2004). Optimal policies for a multi-echelon inventory problem. *Management Science*, 50(12), 1782.

Coelho, L. C., Cordeau, J.-F., & Laporte, 34 G. (2014). Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52, 55–67. https://doi.org/10.1016/j.cor.2014.07.001

Craven, M. W., & Shavlik, J. W. (1997). Using neural networks for data mining. *Future Generation Computer Systems,* 13(2-3), 211–229. https://doi.org/10.1016/S0167-739X(97)00022-8

Christopher, O. (2015). *Understanding LSTM Networks*. Colah's Blog. http://colah.github.io/posts/2015-08-Understanding-LSTMs/?fbclid=IwAR0Qd_4afRk6WdjiAbMPaMby9vFG94YerAGw2FUJGmU1DhyoUiJ7pmcknk

Czabanski R., Jezewski M., Leski J. (2017) Introduction to Fuzzy Systems. In: Prokopowicz P., Czerniak J., Mikołajewski D., Apiecionek Ł., Ślęzak D. (eds) Theory and Applications of Ordered Fuzzy Numbers. Studies in Fuzziness and Soft Computing, vol 356. *Springer*, Cham. https://doi.org/10.1007/978-3-319-59614-3_2

Das, P., & Chaudhury, S. (2007). Prediction of retail sales of footwear using feedforward and recurrent neural networks. *Neural Computing and Applications, 16*(4), 491-502.

Dalmazo, L. (2018, July 11). Empreendedores contam como recomeçar após fim de startups. O Estado de S. Paulo. https://link.estadao.com.br/noticias/inovacao,empreendedores-contam-como-recomecar-apos-fim-de-startups,70002397754

Dai, Z., Aqlan, F., & Gao, K. (2017). Optimizing multi-echelon inventory with three types of demand in supply chain. *Transportation Research Part E*, 107, 141–177. https://doi.org/10.1016/j.tre.2017.09.008

Della Penna, N., & Huang, H. (2010). Constructing consumer sentiment index for US using Google searches (No. 2009-26).

Denham, B. E. (2012). Thinking, fast and slow. Journal of Communication, 62(5), 11.

D. Gao, N. Wang, Z. He and T. Jia, "The Bullwhip Effect in an Online Retail Supply Chain: A Perspective of Price-Sensitive Demand Based on the Price Discount in E-commerce," in IEEE Transactions on Engineering Management, vol. 64, no. 2, pp. 134-148, May 2017, doi: 10.1109/TEM.2017.2666265.

De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. International Journal of Forecasting, 22(3), 443–473. https://doi.org/10.1016/j.ijforecast.2006.01.001

Ding, S., Li, H., Su, C. et al. Evolutionary artificial neural networks: a review. *Artif Intell Rev* 39, 251–260 (2013). https://doi-org.proxy2.hec.ca/10.1007/s10462-011-9270-6

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science, 14*(2), 179-211. doi:10.1207/s15516709cog1402_1

eMarketer. (June 27, 2019). Retail e-commerce sales worldwide from 2014 to 2023 (in billion U.S. dollars) [Graph]. In Statista. Retrieved August 25, 2020, from https://www-statista-com.proxy2.hec.ca/statistics/379046/worldwide-retail-e-commerce-sales/

eMarketer. (May 17, 2019). Most popular online retailers in Brazil in March 2019, based on number of unique visitors (in millions) [Graph]. In Statista. Retrieved August 30, 2020, from https://www-statista-com.proxy2.hec.ca/statistics/254739/most-popular-online-retailers-in-brazil/

Euromonitor International. (September 14, 2020). The Coronavirus Era: Where Consumers Shop. Retrieved from https://www-portal-euromonitor-com.proxy2.hec.ca/portal/analysis/tab#

Facebook. (2017). Seasonality, Holiday Effects, And Regressors. Prophet. https://facebook.github.io/prophet/docs/seasonality,_holiday_effects,_and_regressors.html#built-in-country-holidays

Fallah Tehrani A., Ahrens D. (2018) Enhanced Predictive Models for Purchasing in the Fashion Field by Applying Regression Trees Equipped with Ordinal Logistic Regression. In: Thomassey S., Zeng X. (eds) Artificial Intelligence for Fashion

Industry in the Big Data Era. Springer Series in Fashion Business. Springer, Singapore. https://doi-org.proxy2.hec.ca/10.1007/978-981-13-0080-6_3

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing & Service Operations Management, 18(1), 69-88.

Fisher, M. and Raman, A. (2018), Using Data and Big Data in Retailing. Prod Oper Manag, 27: 1665-1669. doi:10.1111/poms.12846

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Fisher, M., Raman, A., McClelland, A. S. (2014, August 01). Are You Ready? Retrieved August 25, 2020, from https://hbr.org/2000/07/are-you-ready

Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: research and practice. *International Journal of Forecasting, (201912).* https://doi.org/10.1016/j.ijforecast.2019.06.004

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29(5)*, 1189-1232. Retrieved August 29, 2020, from http://www.jstor.org/stable/2699986

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4), 367–378.

General Mills Inc., & Sonnek, P. (2016). GeneralMills/pytrends. GitHub. https://github.com/GeneralMills/pytrends

Ge, D., Pan, Y., Shen, Z.-J. (M., Wu, D., Yuan, R., & Zhang, C. (2019). Retail supply chain management: a review of theories and practices. *Journal of Data, Information and Management*, 1(1-2), 45–64. https://doi.org/10.1007/s42488-019-00004-z

Giacomini, R., & Rossi, B. (2009). Detecting and Predicting Forecast Breakdowns. *Review of Economic Studies, 76*(2), 669-705. doi:10.1111/j.1467-937x.2009.00545.x

Gocheva-Ilieva, S. G., Voynikova, D. S., Stoimenova, M. P., Ivanov, A. V., & Iliev, I. P. (2019). Regression trees modeling of time series for air pollution analysis and forecasting. *Neural Computing and Applications, 31(12)*, 9023–9039. https://doi.org/10.1007/s00521-019-04432-1

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Goodwin, Paul. (2018). Profit from your forecasting software: a best practice guide for sales forecasters.

Google. (2020). FAQ about Google Trends data - Trends Help. Trends Help. https://support.google.com/trends/answer/4365533?hl=en&ref_topic=6248052

Gür Ali Ö, Sayin, S., Woensel, van, T., & Fransoo, J. C. (2009). Sku demand forecasting in the presence of promotions. *Expert Systems with Applications, 36(10)*, 12340–12348.

Hand, C., & Judge, G. (2012). Searching for the picture: forecasting UK cinema admissions using Google Trends data. Applied Economics Letters, 19(11), 1051–1055. doi: 10.1080/13504851.2011.613744

Hastie T., Tibshirani R., Friedman J. (2009) Introduction. In: The Elements of Statistical Learning. Springer Series in Statistics. Springer, New York, NY.

Hastie T., Tibshirani R., Friedman J. (2009) Additive Models, Trees, and Related Methods. In: The Elements of Statistical Learning. Springer Series in Statistics. Springer, New York, NY.

Hastie T., Tibshirani R., Friedman J. (2009) Model Assessment and Selection. In: The Elements of Statistical Learning. Springer Series in Statistics. Springer, New York, NY.

Hendricks, K. B., & Singhal, V. R. (2005). Association between supply chain glitches and operating performance. Management Science, 51(5), 695-711.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2020). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*. doi:10.1016/j.ijforecast.2020.06.008

Hofmann, E. and Rutschmann, E. (2018), "Big data analytics and demand forecasting in supply chains: a conceptual analysis", The International Journal of Logistics Management, Vol. 29 No. 2, pp. 739-766. https://doi.org/10.1108/IJLM-04-2017-0088

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting, 20(1), 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

Huan, S. H., Sheoran, S. K., & Wang, G. (2004). A review and analysis of supply chain operations reference (scor) model. Supply Chain Management, 9(1), 23–29.

Hyndman, R. J. (2006). Another look at forecast-accuracy metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, 4(4), 43-46.

Hyndman, R. J., & Athanasopoulos, G. (2014). Forecasting : principles and practice. OTexts.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. International Journal of Forecasting, 18(3), 439–454. https://doi.org/10.1016/S0169-2070(01)00110-8

Hwarng, H. B. (2001). Insights into neural-network forecasting of time series corresponding to arma(p, q) structures. *Omega, 29(3)*, 273–289. https://doi.org/10.1016/S0305-0483(01)00022-6

Ji, S., Wang, X., Zhao, W., & Guo, D. (2019). An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-commerce Enterprise. *Mathematical Problems in Engineering, 2019*, 1-15. doi:10.1155/2019/8503252

Juniper research highlights AI spending by retailers. (2019). Entertainment Close - Up, Retrieved from http://proxy2.hec.ca/login?url=https://proxy2.hec.ca:2379/docview/2209408696?accountid=11357

Kahn, K. B. (2003). How to measure the impact of a forecast error on an enterprise?. The Journal of Business Forecasting, 22(1), 21.

Kalpić D., Hlupić N., Lovrić M. (2011) Student's t-Tests. In: Lovric M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg.

Karchere, A. (1976). Forecast Error and Planning. Business Economics, 11(3), 70-73. Retrieved August 29, 2020, from http://www.jstor.org/stable/23481468

Kapalka, B. A., Katircioglu, K., & Puterman, M. L. (2009). Retail inventory control with lost sales, service constraints, and fractional lead times. *Production and Operations Management*, 8(4), 393–408. https://doi.org/10.1111/j.1937-5956.1999.tb00315.x

Kerkkänen, A., Korpela, J., & Huiskonen, J. (2009). Demand forecasting errors in industrial context: Measurement and impacts. International Journal of Production Economics, 118(1), 43-48.

Kim, Z., Oh, H., Kim, H. et al. Modeling long-term human activeness using recurrent neural networks for biometric data. BMC Med Inform Decis Mak 17, 57 (2017). https://doi.org/10.1186/s12911-017-0453-1

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.

Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics, 143*(1), 198-206. doi:10.1016/j.ijpe.2013.01.009

Krishna, A., V, A., Aich, A., & Hegde, C. (2018). Sales-forecasting of Retail Stores using Machine Learning Techniques. *2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions* (CSITSS), 160–166. https://doi.org/10.1109/csitss.2018.8768765

Kroger. (April 1, 2020). Kroger's total sales in the United States from 2007 to 2019 (in billion U.S. dollars) [Graph]. *In Statista*.

Krzywinski, M., & Altman, N. (2017). Points of significance: Classification and regression trees. *Nature Methods*, 14(8), 757-758.

Lee, T. S., & Adam Jr, E. E. (1986). Forecasting error evaluation in material requirements planning (MRP) production-inventory systems. *Management Science, 32(9)*, 1186-1205.

Lee, H. L., Padmanabhan, V., & Whang, S. (1997). The bullwhip effect in supply chains. *Sloan management review, 38*, 93-102.

Lecinski, J. (2014, August). ZMOT: Why It Matters Now More Than Ever. Think with Google. https://www.thinkwithgoogle.com/marketing-resources/micro-moments/zmot-why-it-matters-now-more-than-ever

Ljung, G. M., Ledolter, J., & Abraham, B. (2014). George Box's contributions to time series analysis and forecasting. Applied Stochastic Models in Business & Industry, 30(1), 25–35.

Lockamy III, A., & McCormack, K. (2004). Linking scor planning practices to supply chain performance: an exploratory study. *International Journal of Operations and Production Management, 24(12)*, 1192–1218.

Loureiro, A. L. D., Miguéis, V. L., & da Silva, L. F. M. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. Decision Support Systems, 114, 81–93. https://doi.org/10.1016/j.dss.2018.08.010

Liu, G., Shao, Q., Lund, R., & Woody, J. (2016). Testing for seasonal means in time series data. *Environmetrics, 27(4)*, 198–211. https://doi.org/10.1002/env.2383

Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. International Journal of Forecasting, 9(4), 527–529. https://doi.org/10.1016/0169-2070(93)90079-3

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting, 36*(1), 54-74. doi:10.1016/j.ijforecast.2019.04.014

Mandl, C. (2019, October 23). Japan's SoftBank invests in Brazilian marketplace integrator Olist. U.S. https://www.reuters.com/article/softbank-latam-olist/japans-softbank-invests-in-brazilian-marketplace-integrator-olist-idUSL2N2771HD

Másson, E., & Wang, Y. (1990). Introduction to computation and learning in artificial neural networks. European Journal of Operational Research, 47(1), 1-28. doi:10.1016/0377-2217(90)90085-p

Mitchell, T. M. (1997). Does machine learning really work? AI Magazine, 18(3), 11.

Murphy, Kevin P.. Machine Learning : A Probabilistic Perspective, MIT Press, 2012. ProQuest Ebook Central.

Mussumeci, E., & Codeço Coelho, F. (2020). Large-scale multivariate forecasting models for Dengue - LSTM versus random forest regression. Spatial and Spatio-Temporal Epidemiology, 35, 100372. https://doi.org/10.1016/j.sste.2020.100372

Na, L., Shuyun, R., Tsan-Ming, C., Chi-Leung, H., & Sau-Fun, N. (2013). Sales forecasting for fashion retailing service industry: a review. Mathematical Problems in Engineering, 2013. https://doi.org/10.1155/2013/738675

Ohrimuk, E. S., Razmochaeva, N. V., Mikhailov, Y. I., & Bezrukov, A. A. (2020). Study of Supervised Algorithms for Solve the Forecasting Retail Dynamics Problem. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 441–445. https://doi.org/10.1109/eiconrus49466.2020.9039112

Olist and André Sionek, (2018) “Brazilian E-commerce Public Dataset by Olist.” Kaggle, doi: 10.34740/KAGGLE/DSV/195341.

Oujdi, S., Belbachir, H., & Boufares, F. (2019). C4.5 Decision Tree Algorithm for Spatial Data, Alternatives and Performances. *Journal of Computing & Information Technology*, 27(3), 29–43.

Önder, I. (2017). Forecasting tourism demand with Google trends: Accuracy comparison of countries versus cities. International Journal of Tourism Research, 19(6), 648–660. doi: 10.1002/jtr.2137

Pandis, N. (2015). Comparison of 2 means for matched observations (paired t test) and t test assumptions. *American Journal of Orthodontics and Dentofacial Orthopedics*, 148(3), 515–516. https://doi.org/10.1016/j.ajodo.2015.06.011

Park, J., Park, Y., & Lee, K. (1991). Composite modeling for adaptive short-term load forecasting. IEEE Transactions on Power Systems, 6(2), 450-457. doi:10.1109/59.76686

Prechelt L. (2012) Early Stopping — But When?. In: Montavon G., Orr G.B., Müller KR. (eds) *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106. https://doi.org/10.1023/A:1022643204877

Sant'Ana, J. (2017, November 14). Olist quer se tornar a maior loja virtual dentro dos marketplaces. Gazeta Do Povo. https://www.gazetadopovo.com.br/economia/nova-economia/olist-quer-ser-a-maior-loja-virtual-dentro-dos-principais-marketplaces-do-pais-3xbe9k4mn13uzs1pm60ym6znz/

Randall, W.S., Gibson, B.J., Clifford Defee, C. and Williams, B.D. (2011), "Retail supply chain management: key priorities and practices", The International Journal of Logistics Management, Vol. 22 No. 3, pp. 390-402. https://doi.org/10.1108/09574091111181381

R. C. Garcia, J. Contreras, M. van Akkeren and J. B. C. Garcia, "A GARCH forecasting model to predict day-ahead electricity prices," in IEEE Transactions on Power Systems, vol. 20, no. 2, pp. 867-874, May 2005, doi: 10.1109/TPWRS.2005.846044.

Rayat C.S. (2018) Tests of Significance. In: Statistical Methods in Medical Research. Springer, Singapore.

Robin, F. (2018). Use of google trends data in banque de france monthly retail trade surveys. Economie Et Statistique, 2018(505-506), 35–63. https://doi.org/10.24187/ecostat.2018.505d.1965

Rohwedder, C. (2006). No. 1 Retailer in Britain Uses' Clubcard'to Thwart Wal-Mart. *Wall Street Journal*, (June 6).

Roll, Y., & Kerbs, A. (1982). Retail inventory control with sales dependent on the stock level. *International Journal of Operations & Production Management*, 2(3), 13–18. https://doi.org/10.1108/eb054682

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. https://doi.org/10.1037/h0042519

Rudolf Kruse (2008) Fuzzy neural network. *Scholarpedia*, 3(11):6043.

Sahin, M., Kizilaslan, R., & Demirel, Ö. F. (2013). Forecasting Aviation Spare Parts Demand Using Croston Based Methods and Artificial Neural Networks. Journal of Economic and Social Research, 15(2), 1-21

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181-1191. doi:10.1016/j.ijforecast.2019.07.001

Salvatore, C., Andrea, M., Alessio, P., Diego, R. R., & Roberto, S. (2018). Forecasting e-commerce products prices by combining an autoregressive integrated moving average (arima) model and google trends data. Future Internet, 11(1). https://doi.org/10.3390/fi11010005

Sarkar D., Bali R., Sharma T. (2018) Feature Engineering and Selection. In: Practical Machine Learning with Python. Apress, Berkeley, CA. https://doi-org.proxy2.hec.ca/10.1007/978-1-4842-3207-1_4

Shefrin, H., & Statman, M. (2003). The contributions of daniel kahneman and amos tversky. *The Journal of Psychology & Financial Markets*, 4(2), 54–58.

Shouwen, J., Xiaojing, W., Wenpeng, Z., & Dong, G. (2019). An application of a three-stage xgboost-based model to sales forecasting of a cross-border e-commerce enterprise. Mathematical Problems in Engineering, 2019. https://doi.org/10.1155/2019/8503252

Silva, E., Hassani, H., Madsen, D., & Gee, L. (2019). Googling fashion: forecasting fashion consumer behaviour using google trends. *Social Sciences*, 8(4), 111–111. https://doi.org/10.3390/socsci8040111

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting, 36*(1), 75-85. doi:10.1016/j.ijforecast.2019.03.017

Souza, G. C. (2014). Supply chain analytics. Business Horizons -Bloomington-, 57(5), 595–605.

StatCounter. (May 10, 2020). Worldwide desktop market share of leading search engines from January 2010 to April 2020 [Graph]. In Statista.

Statista. (August 10, 2020). Distribution of the e-commerce market in Latin America in 2020, by country [Graph]. In Statista.

Statista. (November 9, 2018). Retail e-commerce sales in BRIC countries in from 2016 to 2023 (in million U.S. dollars) [Graph]. In Statista.

Statista. (2019). Grocery Stores: Kroger brand report [Dataset]. Statista Global Consumer Survey. https://www.statista.com/study/46058/kroger-company/

Snyder, L. V., & Shen, Z.-J. M. (2019). Fundamentals of supply chain theory (Second). John Wiley & Sons. http://proquest.safaribooksonline.com/?fpi=9781119024842.

Taylor, J. W. (2003). Exponential smoothing with a damped multiplicative trend. International Journal of Forecasting, 19(4), 715–725. https://doi.org/10.1016/S0169-2070(03)00003-7

Taylor, S. J., & Letham, B. (2017). Forecasting at scale. PeerJ PrePrints,

Teng, Y., Bi, D., Xie, G., Yuan, J., Huang, Y., Lin, B., . . . Tong, Y. (2017). Dynamic forecasting of zika epidemics using google trends. PLoS One, 12(1)

The Canadian Press. (2020, March 22). Quebec closes shopping malls, restaurants, extends school closure till May. National Post. https://nationalpost.com/news/quebec-closes-shopping-malls-restaurants-extends-school-closure

Thomassey, S., Happiette, M., & Castelain, J. M. (2005). A short and mean-term automatic forecasting system--application to textile logistics. European Journal of Operational Research, 161(1), 275–284.

Thomassey, S. (2013). Sales Forecasting in Apparel and Fashion Industry: A Review. Intelligent Fashion Forecasting Systems: Models and Applications, 9-27. doi:10.1007/978-3-642-39869-8_2

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal- Royal Statistical Society Series B*, 58(1), 267–288.

Toomey J.W. (2000) Replenishing Independent Demand. In: Inventory Management. Materials Management / Logistics Series, vol 12. Springer, Boston, MA.

Tsai, C.-F., & Chen, M.-L. (2010). Credit rating by hybrid machine learning techniques. *Applied Soft Computing*, 10(2), 374–380. https://doi.org/10.1016/j.asoc.2009.08.003

Varian, H., & Choi, H. (2009, April 2). Predicting the Present with Google Trends. Google AI Blog. https://ai.googleblog.com/2009/04/predicting-present-with-google-trends.html

Velicer, W. F., & Colby, S. M. (2005). A comparison of missing-data procedures for arima time-series analysis. Educational and Psychological Measurement, 65(4), 596–615. https://doi.org/10.1177/0013164404272502

Viale, J. D. (1996). *Basics of inventory management : From warehouse to distribution center*. ProQuest Ebook Central https://ebookcentral-proquest-com.proxy2.hec.ca

Volkovs, M., Yu, G. W., & Poutanen, T. (2017). Content-based neighbor models for cold start in recommender systems. In Proceedings of the Recommender Systems Challenge 2017 (pp. 1-6).

Wee, H. (2011). Inventory systems : *Modeling and research methods*. ProQuest Ebook

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6(3), 324–342. https://doi.org/10.1287/mnsc.6.3.324

Wolfgang Digital. (January 15, 2020). Distribution of global e-commerce sessions as of October 2019, by source and medium [Graph]. In Statista.

Woo, J., & Owen, A. L. (2018). Forecasting private consumption with Google Trends data. Journal of Forecasting, 38(2), 81–91. doi: 10.1002/for.2559

Wu, Z., Huang, N. E., Long, S. R., & Peng, C.-K. (2007). On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings- National Academy of Sciences Usa, 104(38)*, 14889–14894.

Wu, C.-S. M., Patil, P., Gunaseelan, S., & 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) Beijing, China 2018 Nov. 23 - 2018 Nov. 25. (2018). 2018 ieee 9th international conference on software engineering and service science (icsess). In *Comparison of different machine learning algorithms for multiple regression on black friday sales data* (pp. 16–20). essay, IEEE. https://doi.org/10.1109/ICSESS.2018.8663760

Yao, X. (1993). A review of evolutionary artificial neural networks. *International journal of intelligent systems*, 8(4), 539-567.

Yesil, E., Kaya, M., & Siradag, S. (2012, July). Fuzzy forecast combiner design for fast fashion demand forecasting. In 2012 International Symposium on Innovations in Intelligent Systems and Applications (pp. 1-5). IEEE.

Yu, L., Zhao, Y., Tang, L., & Yang, Z. (2019). Online big data-driven oil consumption forecasting with Google trends. *International Journal of Forecasting*, 35(1), 213–223. doi: 10.1016/j.ijforecast.2017.11.005

Yu Q., Wang K., Strandhagen J.O., Wang Y. (2018) Application of Long Short-Term Memory Neural Network to Sales Forecasting in Retail—A Case Study. In: Wang K., Wang Y., Strandhagen J., Yu T. (eds) Advanced Manufacturing and Automation VII. IWAMA 2017. Lecture Notes in Electrical Engineering, vol 451. Springer, Singapore.

Xia, Z., Xue, S., Wu, L., Sun, J., Chen, Y., & Zhang, R. (2020). ForeXGBoost: passenger car sales prediction based on XGBoost. DISTRIBUTED AND PARALLEL DATABASES.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:. *International Journal of Forecasting, 14*(1), 35-62. doi:10.1016/s0169-2070(97)00044-7

Appendix

Table A1: Supply Chain Decisions and the SCOR Model (adapted from Souza, 2014)

	SCOR Domain Area			
	Source	**Make**	**Deliver**	**Return**
Supply Chain Decisions	Procure products and materials	Manufacture, repair and recycle products and materials	Packaging and shipping. Inbound and outbound delivery.	Process merchandise return and determine disposal for products, materials and assets.
Strategic Decisions Timeframe: years	Strategic sourcing partners and global processes.	Where to locate manufacturing plants. Which products to produce/store in a plant.	Where to locate a warehouse or distribution center. Transportation fleet planning.	Location of returns and disposal facilities.
Tactical Decisions Timeframe: months	Supplier contracts and risk management.	Sales and operations planning.	Inventory control policies. Distribution planning.	Reverse logistics planning.
Operational Decisions Timeframe: days/weeks	Materials requirements planning. Inventory replenishment orders.	Manufacturing capacity leveling and master production schedules. Workforce scheduling.	Vehicle route planning for outbound delivery.	Vehicle route planning for collecting returns.
	Plan Demand forecasts support short-, mid- and long-term decisions.			

Table A2: Breakfast at the Frat Weekly Transactional Data

WEEK_END_DATE	2009-01-14 00:00:00
STORE_NUM	367
UPC	1111009477
UNITS	13
VISITS	13
HHS	13
SPEND	18.07
PRICE	1.39
BASE_PRICE	1.57
FEATURE	0
DISPLAY	0
TPR_ONLY	1
DESCRIPTION	PL MINI TWIST PRETZELS
MANUFACTURER	PRIVATE LABEL
CATEGORY	BAG SNACKS
SUB_CATEGORY	PRETZELS
PRODUCT_SIZE	15 OZ
STORE_ID	367
STORE_NAME	15TH & MADISON
ADDRESS_CITY_NAME	COVINGTON
ADDRESS_STATE_PROV_CODE	KY
MSA_CODE	17140
SEG_VALUE_NAME	VALUE
PARKING_SPACE_QTY	196
SALES_AREA_SIZE_NUM	24721
AVG_WEEKLY_BASKETS	12706.5

Table A3: Brazilian E-commerce – Google Trends Series

Product Category	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Bed, Bath & Table	Casas Bahia	Shopping	Retail chain company	Google Shopping	Sao Paolo
	jogo de cama	Shopping	—	Web	Sao Paolo
	lojas de colchoes	All categories	—	Web	Sao Paolo
	Enxovais	All categories	—	Web	Sao Paolo
	Pillow	Shopping	Topic	Web	Sao Paolo
	Travesseiro fibrasca	All categories	—	Web	Brazil
	Jolitex	All categories	—	Web	Sao Paolo
	Veste a Casa	All categories	—	Web	Sao Paolo
Health and Beauty	212 sexy	Beauty & Fitness	—	Web	Sao Paolo
	perfume hugo boss	Perfumes & Fragrances	—	Web	Sao Paolo
	versace perfume	Shopping	—	Web	Sao Paolo
	perfumes importados	All categories	—	Web	Sao Paolo
	Schwarzkopf	Beauty & Fitness	Topic	Web	Sao Paolo
	Schwarzkopf Professional IGORA	Beauty & Fitness	Topic	Web	Sao Paolo
	Igora royal	All categories	—	Web	Brazil
	mercado livre mascara	All categories	—	Web	Brazil
	Senscience	All categories	—	Web	Sao Paolo
	Curls	Beauty & Fitness	Topic	Web	Sao Paolo
	óleo de côco	Beauty & Fitness	—	Web	Sao Paolo
	mascara matizadora	Beauty & Fitness	—	Web	Sao Paolo
	Shampoo	All categories	Topic	Google Shopping	Sao Paolo
Sports and Leisure	barraca de camping	All categories	—	Web	Sao Paolo
	barraca de praia	All categories	—	Web	Sao Paolo
	barraca mor	All categories	—	Web	Brazil
	caixa termica	All categories	—	Web	Sao Paolo
	Albatroz Fishing	All categories	—	Web	Brazil
	Marine sports	All categories	—	Web	Sao Paolo
	Fish hook	All categories	Topic	Web	Sao Paolo

Product Category	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Furniture Décor	cadeira brasil	All categories	—	Web	Sao Paolo
	lojas de moveis	Home furnishings, Shopping	—	Web	Sao Paolo
	lojas de moveis online	All categories	—	Web	Brazil
	puff moveis	All categories	—	Web	Sao Paolo
	decoração de casa	All categories	—	Web, Image	Sao Paolo
	decoração de sala	All categories	—	Web, Image	Sao Paolo
	Do it yourself	Shopping	Topic	Web	Sao Paolo
Watches and Gifts	relógio de pulso	Shopping	—	Web	Brazil
	relogio orient	Shopping	—	Web	Sao Paolo
	mercado livre relogio	Shopping	—	Web	Sao Paolo
	relogio Armani	Shopping	—	Web	Sao Paolo
	Men's watch	All categories, Shopping	Topic	Web	Sao Paolo
	relogio feminine	Shopping	—	Web	Sao Paolo
	relogio feminino michael kors	All categories	—	Web	Sao Paolo
	relogio citizen	Shopping	—	Web	Sao Paolo
	Orient watch	Shopping	Company	Web	Sao Paolo
	Apple watch	Shopping	Watch	Web	Sao Paolo
	Watch	Shopping	Topic	Google Shopping	Sao Paolo
Telephony	Moto G5	Shopping	Mobile phone	Web	Brazil, Sao Paolo
	bateria moto g5	All categories	—	Web	Sao Paolo
	Moto G5 Plus	Shopping	—	Web	Brazil
	celular Motorola	Mobile Phones	—	Web	Brazil, Sao Paolo
	Cellular Samsung	Shopping	—	Web	Brazil, Sao Paolo
	Bateria Samsung	All categories	—	Web	Sao Paolo
	Samsung Galaxy On7 (2016)	All categories	—	Web	Sao Paolo
	J7 prime durado	All categories	—	Web	Sao Paolo
	iPhone 6	Shopping	Mobile Phone	Web	Sao Paolo
	Capa Motorola	All categories	—	Web	Brazil
	Mobile phone accessories	All categories	Topic	Web	Brazil
	carregador veicular	All categories	—	Web	Brazil
	pelicula de vidro	All categories	—	Web	Sao Paolo
	Caphina mercado livre	All categories	—	Web	Sao Paolo

Product Category	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Housewares	Hamilton Beach Brands	All categories	Home appliance company	Web	Brazil, Sao Paolo
	Kärcher	All categories, Shopping	Company	Web	Brazil, Sao Paolo
	Robert Bosch	Shopping	Engineering Company	Web	Sao Paolo
	Misturador Monocomando Cozinha	All categories	—	Web	Brazil
	Lorenzetti	Shopping	Topic	Web	Sao Paolo
	Duo Shower Quadra Lorenzetti Multitemperature - 110-130V - 5500W	All categories	—	Web	Sao Paolo
	a lavadora	All categories	—	Web	Sao Paolo
	maquina de lavar	Shopping	—	Web	Sao Paolo
	Luminaria	Shopping	—	Web	Sao Paolo
	utilidades domesticas	All categories	—	Web	Sao Paolo
	Canecas	Kitchen & Dinning	—	Web	Sao Paolo
Common Google Trends series used in forecasting the sales across categories.	Olist	All categories	Topic	Web	Brazil, Sao Paolo
	Mercado Olist	All categories	—	Web	Brazil, Sao Paolo
	Submarino	Shopping	Company	Web, Google Shopping	Sao Paolo
	Cupom Submarino	All categories	—	Web	Brazil, Sao Paolo
	Cupom Desconto Submarino	All categories	—	Web	Sao Paolo
	B2W	Shopping	Online retail company	Web	Sao Paolo
	Amazon	Shopping	E-commerce copmany	Web	Sao Paolo
	MercadoLibre	Shopping	Online marketplace company	Google Shopping	Sao Paolo
	MercadoLibre SA	Shopping	Company	Web	Sao Paolo

Table A4: Breakfast at the Frat – Google Trends Series

Product Category	Manufacturer & Product Sold Name	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Cold Cereal	Kellogg's, Frosted Flakes	Kellogg's	All Categories, Food & Drink, Shopping	Company	Web, Image	USA, Ohio, Kentucky, Texas
		Frosted Flakes	All Categories, Food & Drink	Breakfast cereal	Web, Image, YouTube	USA
	General Mills, Honey Nut Cheerios	General Mills	All Categories, Food & Drink, Shopping	Food Company	Web, Image	USA, Ohio, Kentucky, Texas
		Honey Nut Cheerios	All categories, Food & Drink, Shopping	Breakfast cereal	Web, Image	USA
		Cheerios cereal	All categories, Food & Drink	—	Web	USA
		Cheerios coupons	All categories, Shopping	—	Web	USA
		Cereal coupons	All categories, Shopping	—	Web	USA
		Special K	All categories	Breakfast cereal	Web	USA
		Breakfast cereal	All categories	—	Web	USA
		Cold cereal	All categories	—	Web	USA
		Quaker cereal	All categories	—	Web	USA
		Cinnamon Toast Crunch	All categories, Food & Drink	Breakfast Cereal	Web	USA
		Kellogg's Froot Loops	All categories	Breakfast cereal	Web	USA
		Corn flakes	All categories, Food & Drink	Breakfast cereal	Web	USA
		Life cereal	All categories, Food & Drink	—	Web	USA
		Kashi	All categories, Food & Drink	Food company	Web	USA
		Cereal nutrition	Food & Drink	—	Web	USA
		Low-carbohydrate diet	Food & Drink	Topic	Web	USA Ohio Kentucky Texas
		Healthy cereal	All categories, Food & Drink	—	Web	USA

Product Category	Manufacturer & Product Sold Name	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Bag Snacks	Private Label, Mini Twist Pretzels	Pretzels	All categories, Food & Drink, Shopping	—	Web	USA Ohio Kentucky Texas
		Pretzel	Food & Drink, Shopping	Topic	Web	USA
		Snyder's of Hanover	All categories, Food & Drink	Company	Web	USA
		Rold Gold	All categories	Topic	Web	USA
		Frito-Lay	All categories, Food & Drink, Shopping	Food Company	Web	USA
		PepsiCo	All categories, Food & Drink, Shopping	Food Company	Web	USA Ohio Kentucky Texas
		Utz Quality Foods	All categories, Food & Drinks	Company	Web	USA
		Nestle	All categories, Food & Drinks, Shopping	Food Company	Web	USA Ohio Kentucky Texas
		Herrs	Food & Drink	—	Web	USA
		Philly Pretzel Factory	All categories	Topic	Web	USA
		chocolate pretzel	All categories	—	Web	USA
		Best pretzels	Food & Drinks	—	Web	USA
		Snack	Shopping	Food	Web	USA
		SuperPretzel	All categories	Topic	Web	USA
		Pretzel Nutrition	All categories	—	Web	USA
		Pretzel coupons	All categories	—	Web	USA
Frozen Pizza	Tomb-Stone, DiGiorno Pepperoni Pizza	DiGiorno Pizza	All categories, Food & Drinks, Shopping	—	Web	USA
		DiGiorno	All categories, Food & Drinks, Shopping	Topic	Web	USA
		Digiorno pizza coupons	All categories, Shopping	—	Web	USA
		Tombstone pizza	All categories, Shopping	—	Web	USA
		Totinos pizza	All categories	—	Web	USA
		Tonys pizza	All categories, Food & Drinks	—	Web	USA
		Red Baron pizza	All categories, Food & Drinks	—	Web	USA
		Frozen pizza	All categories, Food & Drinks	—	Web	USA
		Pepperoni pizza	All categories, Food & Drinks	—	Web	USA

Product Category	Manufacturer & Product Sold Name	Search Term	Search Category	Search Semantic Tag	Search Type	Search Location
Common Google Trends series used in forecasting the sales of products sold across categories.		Coupons.com	All categories, Shopping	—	Web	USA Ohio Kentucky Texas
		printable coupons for groceries	All categories	—	Web	USA
		Nutrition	All categories	Topic	Web	USA
		Gluten-free diet	All categories	Topic	Web	USA
		Gluten	Shopping	Food	Web	USA
		Kroger	All categories, Food & Drinks, Grocery & Food Retailers, Shopping	Retail company	Web	USA Ohio Kentucky Texas
		Kroger coupons	Shopping	—	Web	USA
		Kroger weekly ad	All categories, Shopping	—	Web	USA Ohio Kentucky Texas
		Kroger ad	Shopping	—	Web	USA
		Walmart	Grocery & Food Retailers, Shopping	Retail Company	Web	USA
		Walmart Grocery Pickup	All categories	Topic	Web	USA
		Walmart coupons	All categories, Food & Drinks, Grocery & Food Retailers, Shopping	—	Web	USA Ohio Kentucky Texas
		Amazon Fresh	All categories, Shopping	—	Web	USA
		Costco	Grocery & Food Retailers, Shopping	Retail Company	Web	USA Ohio Kentucky Texas
		Target Corporation	Grocery & Food Retailers	Retail Company	Web	USA
		Online grocery	Shopping	—	Web	USA
		Holiday shopping	All categories	—	Web	USA

Table A5: XGBoost Hyperparameters

Hyperparameter	Range of Values	Function
colsample_bytree	[0,1]	Defines the subsample ratio of columns when constructing each tree.
learning_rate (eta)	[0,1]	Specific the size shrinkage of weights when weights are assigned to new features after each boosting step. This helps in preventing overfitting.
max_depth	[0,∞] where zero is only accepted when tree_method parameter is set to 'hist'	Sets the maximum depth of a tree. The larger the depth, the more complex is the model which also leads to higher memory consumption. Complex models tend to lead to overfitting.
min_child_weight	[0,∞]	For a regression task, this parameter specifies the minimum number of instances required to be in each decision node. The larger the value is the more conservative is the model.
n_estimator	[0,∞]	Specifies the number of boosted trees in the final model
random_state	[0,∞]	Sets the random number seed
subsample	[0,∞]	Specifies the subsample ratio of training observations
tree_method	approx.; hist and gpu_his	Specifies the tree construction algorithm used by the model. The default parameter is auto which makes a heuristic decision to choose the fastest and most conservative method. This parameter also impacts training time.

Table A6: LSTM Hyperparameters

Hyperparameter	Range of Values / Selected Value	Function
Learning Rate	Sampled from a uniform distribution between the values of 0.0001 and 0.01	Scales the magnitude of weight updates that minimize the loss function.
Total Layers	Sampled from a discrete uniform distribution between the values of 1 and 2	Defines the number of layers in the model.
Number of Units	Sampled from a discrete uniform distribution between the values of 5 and 100	Specifies the number of hidden units.
Dropout	The value(s) for dropout is applied after each LSTM layer is added and is always the same value. By default, we apply 0.1.	Reduces overfitting and could be included in the hyperparameter search if desired.
Optimizer	Adam (Kingma and Ba, 2014)	Specifies the optimizer to use.

Table A7: Brazilian E-commerce – Experiment 1 Results by MASE and RMSSE

Data Input	Product Category	Model	MASE	RMSSE
Historical sales	Bed, Bath and Table	SARIMA	0.98	0.80
		FBProphet	1.47	1.14
		XGBoost	1.09	0.94
		LSTM	1.49	1.20
	Furniture Décor	SARIMA	1.70	1.65
		FBProphet	1.12	0.89
		XGBoost	1.62	1.21
		LSTM	1.60	1.23
	Health and Beauty	SARIMA	1.80	1.37
		FBProphet	1.61	1.37
		XGBoost	2.33	1.79
		LSTM	2.79	2.01
	Housewares	SARIMA	1.84	1.69
		FBProphet	1.71	1.44
		XGBoost	1.76	1.65
		LSTM	2.12	1.87
	Sports and Leisure	SARIMA	1.33	1.24
		FBProphet	1.38	1.21
		XGBoost	2.70	2.23
		LSTM	2.05	1.81
	Telephony	SARIMA	1.47	1.30
		FBProphet	1.01	0.95
		XGBoost	2.39	2.08
		LSTM	2.03	1.64
	Watches and Gifts	SARIMA	1.75	1.50
		FBProphet	1.65	1.49
		XGBoost	2.24	1.70
		LSTM	2.37	1.93

Table A8: Brazilian E-commerce – Experiment 1 & 2 Results by MASE and RMSSE

Data Input	Product Category	Model	MASE	RMSSE
Historical sales & Google Trends	Bed, Bath and Table	XGBoost	1.09	0.94
		XGBoost_GoogleTrends	1.67	1.37
		LSTM	1.49	1.20
		LSTM_GoogleTrends	1.65	1.27
	Furniture Décor	XGBoost	1.62	1.21
		XGBoost_GoogleTrends	1.51	1.24
		LSTM	1.60	1.23
		LSTM_GoogleTrends	1.56	1.22
	Health and Beauty	XGBoost	2.33	1.79
		XGBoost_GoogleTrends	2.73	2.17
		LSTM	2.79	2.01
		LSTM_GoogleTrends	3.36	2.41
	Housewares	XGBoost	1.76	1.65
		XGBoost_GoogleTrends	1.90	1.87
		LSTM	2.12	1.87
		LSTM_GoogleTrends	2.24	1.97
	Sports and Leisure	XGBoost	2.70	2.23
		XGBoost_GoogleTrends	2.71	2.32
		LSTM	2.05	1.81
		LSTM_GoogleTrends	2.00	1.76
	Telephony	XGBoost	2.39	2.08
		XGBoost_GoogleTrends	2.07	1.78
		LSTM	2.03	1.64
		LSTM_GoogleTrends	2.10	1.66
	Watches and Gifts	XGBoost	2.24	1.70
		XGBoost_GoogleTrends	2.61	1.99
		LSTM	2.37	1.93
		LSTM_GoogleTrends	2.55	2.00

Table A9: Breakfast at the Frat – Experiment 1 Results by MASE & RMSSE

Data Input	Store	Product	Model	MASE	RMSSE
Historical Sales	Ohio (ID: 2277)	Honey Nut Cheerios (UPC: 1600027527)	SARIMA	1.08	0.86
			FBProphet	1.20	0.86
			XGBoost	0.96	0.80
			LSTM	0.90	0.81
		Kellogg's Frosted Flakes (UPC: 3800031838)	SARIMA	1.10	0.80
			FBProphet	1.11	0.81
			XGBoost	1.22	0.98
			LSTM	0.79	0.74
		Private Label Mini Twist Pretzels (UPC: 1111009477)	SARIMA	1.30	1.23
			FBProphet	1.28	1.17
			XGBoost	1.51	1.42
			LSTM	1.24	1.22
		Digiorno Pepperoni Pizza (UPC: 7192100339)	SARIMA	1.29	1.02
			FBProphet	0.95	0.80
			XGBoost	1.23	1.11
			LSTM	1.09	0.93
	Kentucky (ID: 389)	Honey Nut Cheerios (UPC: 1600027527)	SARIMA	1.02	0.74
			FBProphet	1.27	0.73
			XGBoost	0.80	0.73
			LSTM	0.90	0.59
		Kellogg's Frosted Flakes (UPC: 3800031838)	SARIMA	1.27	1.12
			FBProphet	1.53	1.21
			XGBoost	1.08	1.06
			LSTM	0.92	0.95
		Private Label Mini Twist Pretzels (UPC: 1111009477)	SARIMA	0.99	0.98
			FBProphet	1.39	1.33
			XGBoost	1.10	1.12
			LSTM	0.92	0.89
		Digiorno Pepperoni Pizza (UPC: 7192100339)	SARIMA	1.45	1.13
			FBProphet	1.20	0.98
			XGBoost	1.42	1.29
			LSTM	1.09	0.95

Data Input	Store	Product	Model	MASE	RMSSE
Historical Sales	Texas (ID: 252299)	Honey Nut Cheerios (UPC: 1600027527)	SARIMA	1.04	0.87
			FBProphet	1.10	0.91
			XGBoost	1.19	1.00
			LSTM	1.00	0.85
		Kellogg's Frosted Flakes (UPC: 3800031838)	SARIMA	0.96	0.72
			FBProphet	1.11	0.77
			XGBoost	1.06	0.89
			LSTM	0.77	0.67
		Private Label Mini Twist Pretzels (UPC: 1111009477)	SARIMA	0.76	0.75
			FBProphet	0.72	0.70
			XGBoost	1.12	1.09
			LSTM	1.04	0.90
		Digiorno Pepperoni Pizza (UPC: 7192100339)	SARIMA	0.94	0.83
			FBProphet	0.91	0.79
			XGBoost	1.27	1.08
			LSTM	0.85	0.74

Figure A1: Olist Solutions (retrieved from the company website: https://olist.com/#)

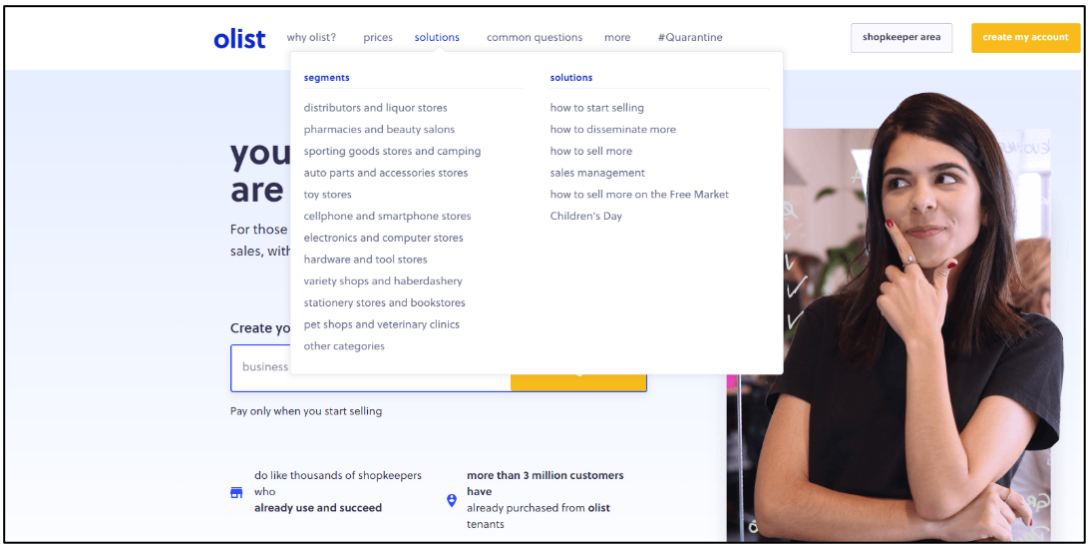


Figure A2: Olist Data Model (retrieved from Olist and Sionek)

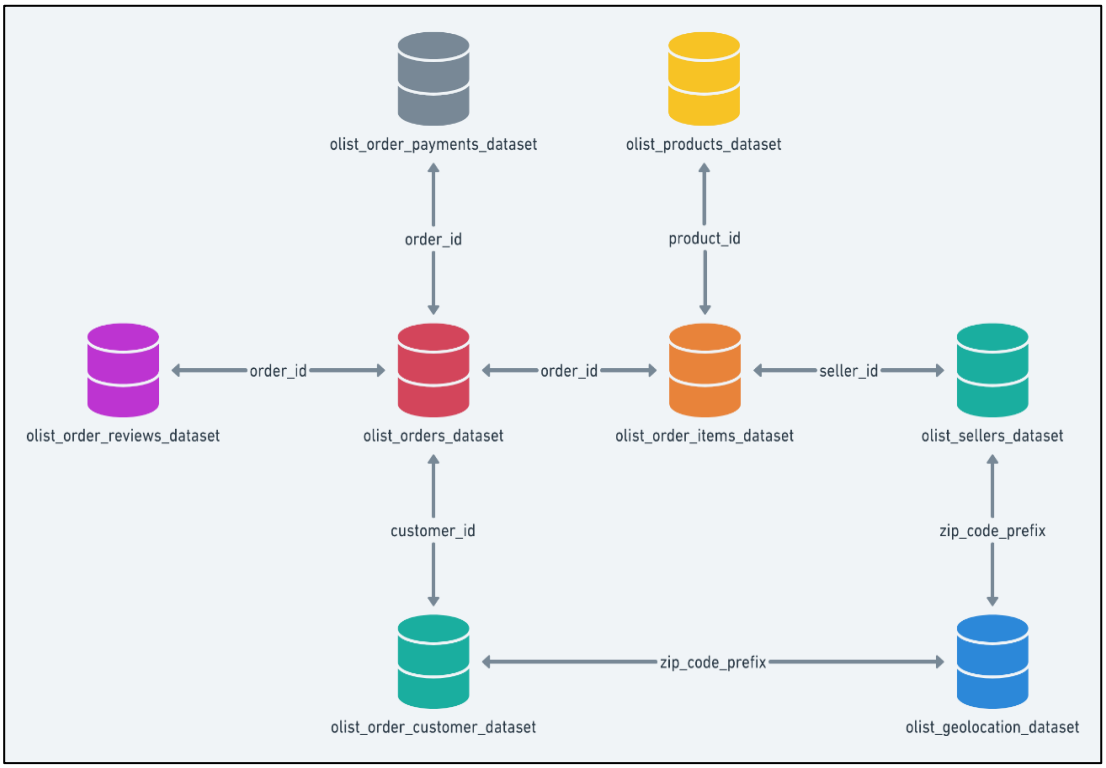


Figure A3: Breakfast at the Frat Data Model

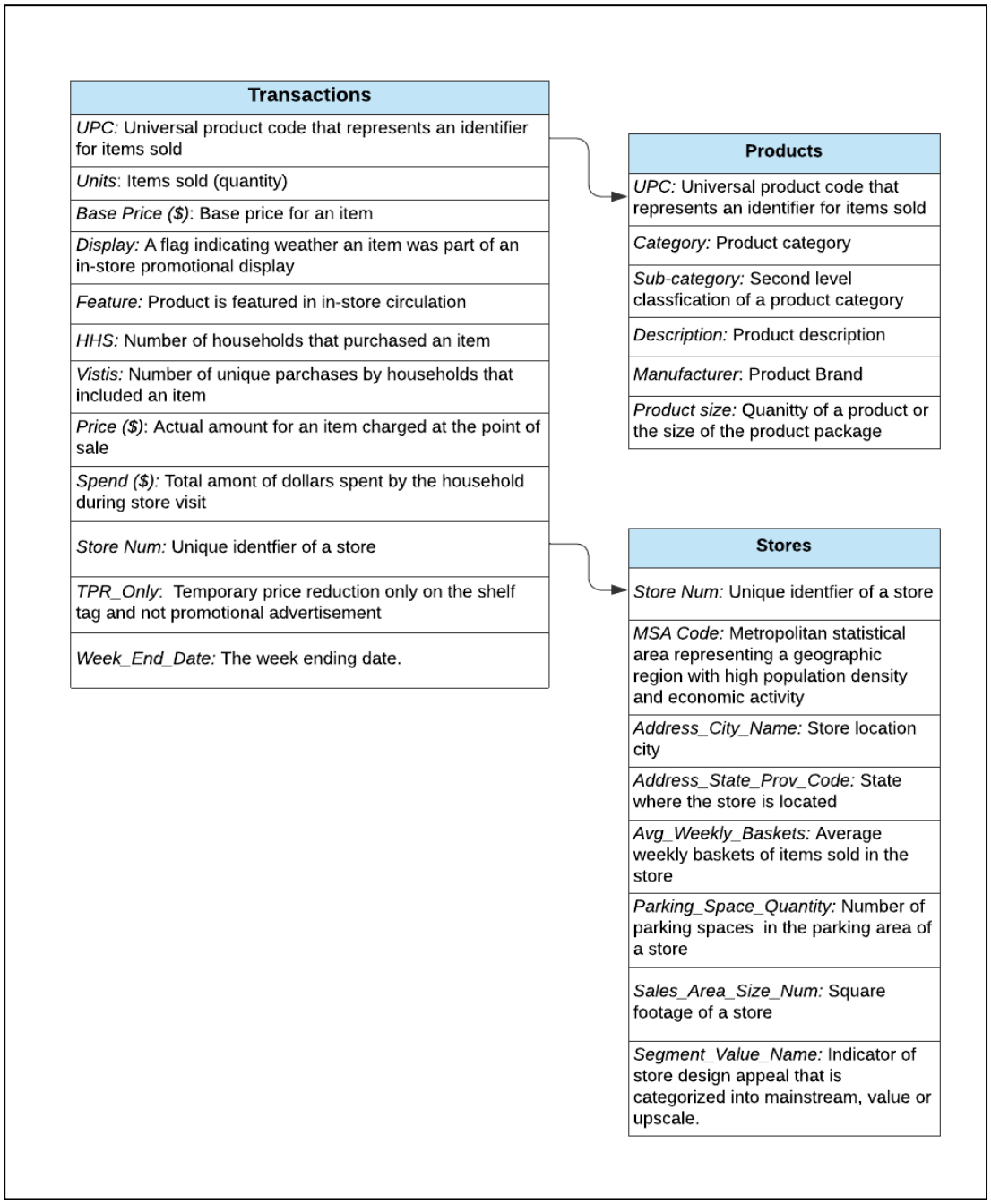


Figure A4: Brazilian E-commerce, Top 10 Unique Products per Category

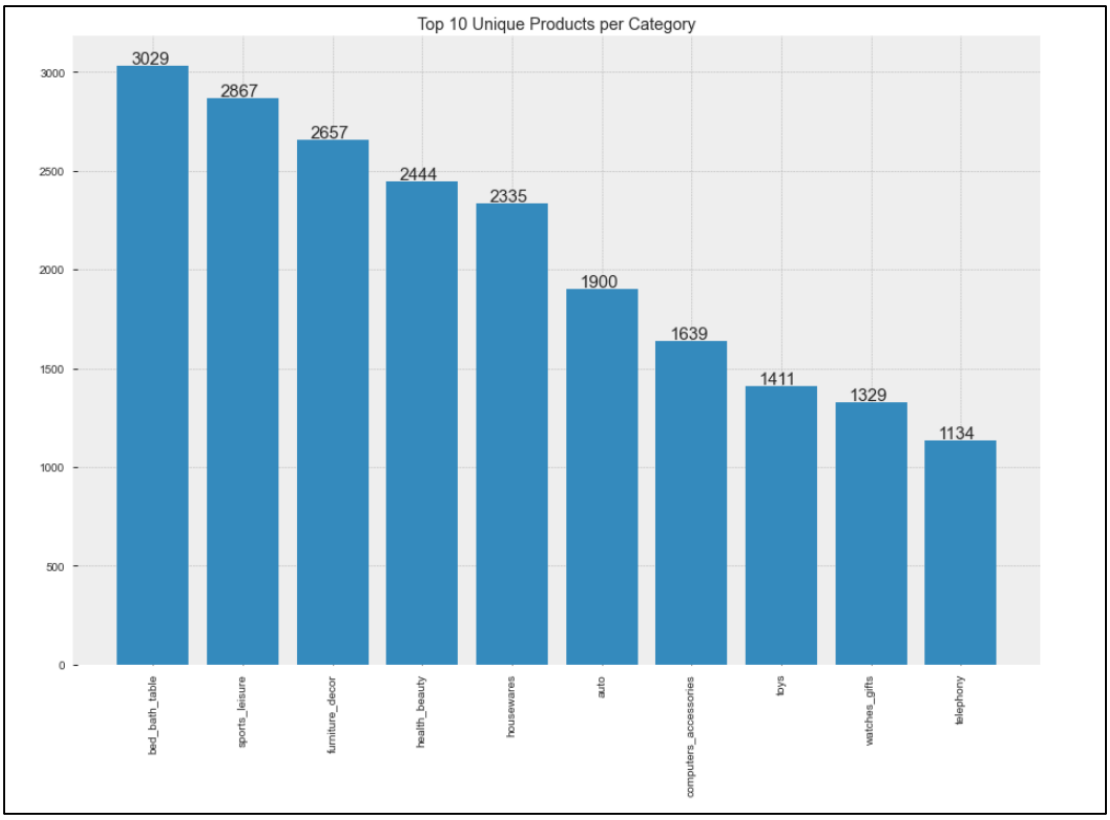


Figure A5: Brazilian E-commerce, Customer Distribution by State

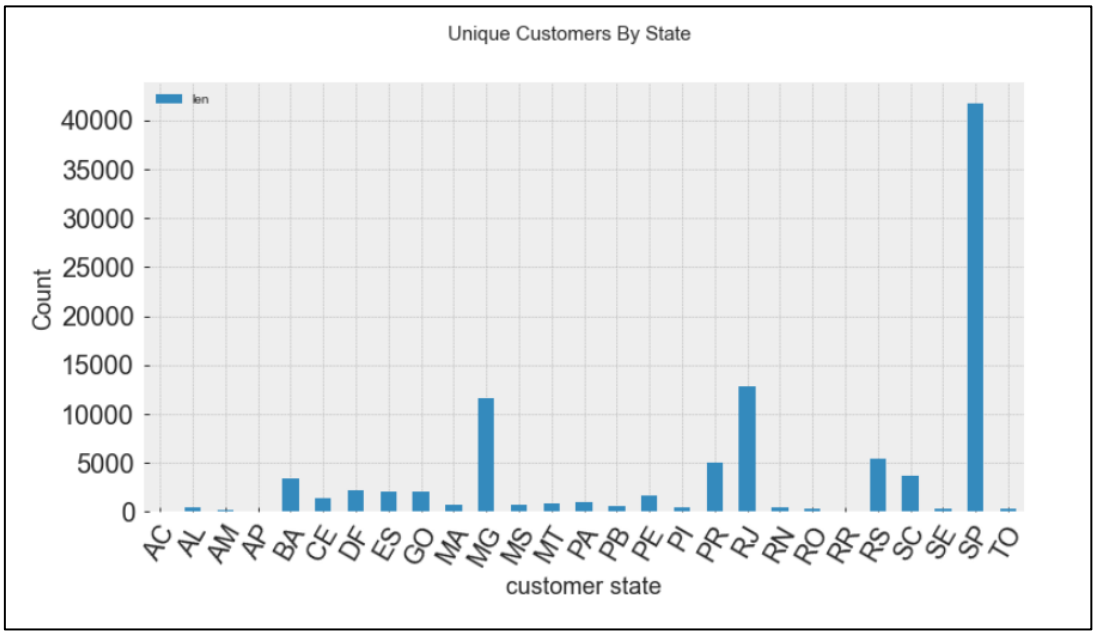


Figure A6: Brazilian E-commerce, Missing Values in the Products Table

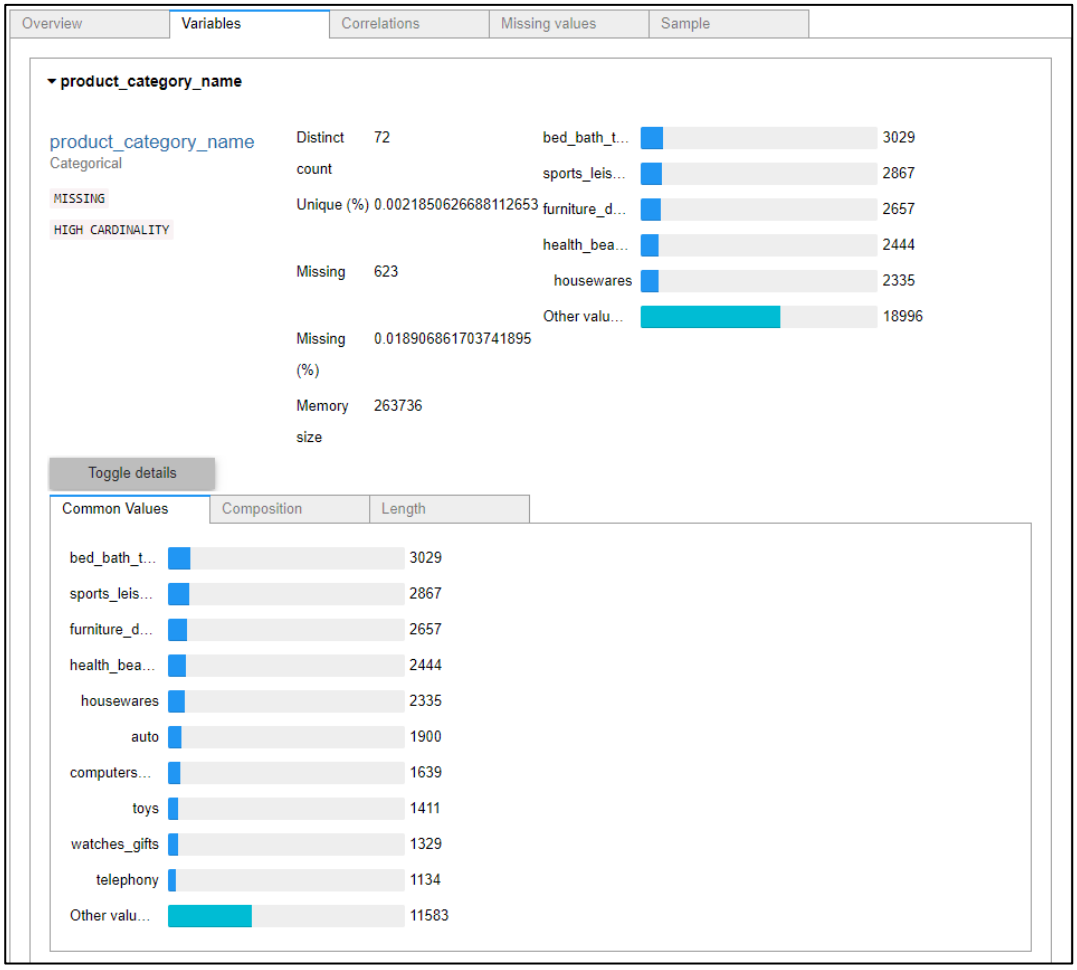


Figure A7: Google Trends User Interface, retrieved from (https://trends.google.com/trends/explore?date=today%203-m&geo=US&q=Frozen%20Pizza)

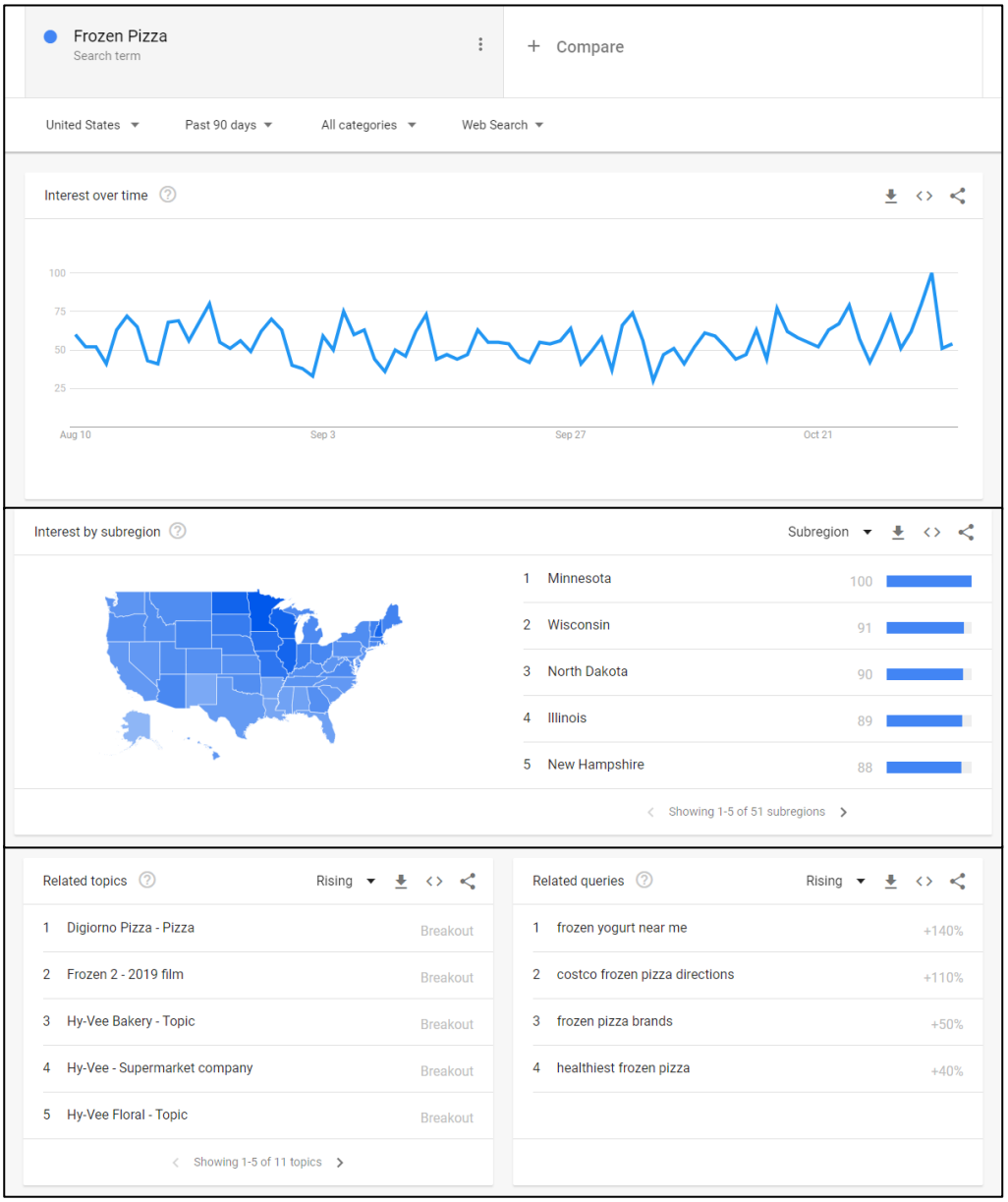


Figure A8: Olist Sample Product Listing (retrieved from Olist and Sionek)

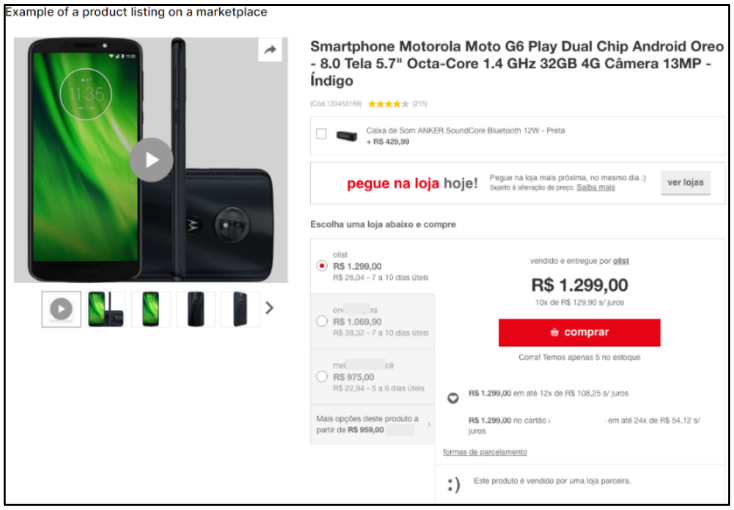


Figure A9: Semantic Labels of Google Trends Search Terms, retrieved from (https://trends.google.com/trends/explore?geo=BR&q=Adidas)