HEC MONTRÉAL

Benchmarking Econometric and Machine Learning Models to Predict Crowdfunding Campaigns Outcomes on Kickstarter

par

Victor Chabot

Sciences de la gestion
(Option Économie Appliquée)

Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)

June 2020

Résumé

L'objectif de cette recherche est de mieux comprendre le *Crowdfunding* et prédire le succès des campagnes de sociofinancement. Nous utilisons une base de données publique issue du site internet Kickstarter contenant presque 300 000 projets. Avec des outils économétriques traditionnels, tels que la méthode des doubles moindres carrés et la régression logistique, nous expliquons les facteurs qui influencent le succès des projets. De plus, nous créons des variables avec du traitement automatique du langage naturel pour les intégrer à une régression logistique. L'objectif est de comprendre comment la description des projets influence l'issue des campagnes de financement. Nous utilisons aussi des modèles d'apprentissage automatique tels que les forêts aléatoires et des gradient boosting models pour prédire le succès des projets sur le site. Nous comparons ensuite la performance de ces modèles avec les modèles économétriques traditionnels. Les programmes ainsi que les bases de données utilisées dans ce projet sont disponibles sur GitHub.

Mots-clés

Socio-financement, Kickstarter, Traitement automatique du langage, succès sociofinancement, campagne de financement, changement de politique, apprentissage automatique, comparaison de modèles, économétrie

Abstract

The objective of this research is to describe, understand and predict a recent economic phenomenon: Crowdfunding. We use a publicly available dataset extracted from the website Kickstarter with almost 300,000 projects. We use traditional econometric tools such as the two-stage least square regression and the logistic regression to understand the determinants of success of crowdfunding campaigns. We also extract features from the blurbs written by each entrepreneur for their campaign. We use those features in a logistic regression to find out how the text influences the outcome of crowdfunding ventures. We also use recent machine learning models, such as extremely randomized trees and gradient boosting models, to predict the outcome of projects. We extensively compare the performance of the econometric and machine learning models. The codes and datasets of this project are available on GitHub.

Keywords

Crowdfunding, crowdsourcing, Kickstarter, NLP, determinant of success, success, campaigns, policy change, machine learning, gradient boosting model, benchmarking, econometrics

Contents

List of Tables

List of Figures

List of acronyms

AUC Area under the curve

CF Crowdfunding

BOW Bag of words

HHI Herfindahl–Hirschman Index

NLP Natural langage processing

LASSO Least Absolute Shrinkage and Selection Operator

LDA Latent Dirichlet allocation

LGBM Light gradient boosting model

OLS Ordinary least square

RMSE Root-mean-square error

ROC Receiver operating characteristic

TF-IDF Term frequency–inverse document frequency,

2SLS Two-stage least square

Acknowledgements

First and foremost, I wish to express my deepest gratitude to my thesis director, Mario Samano, for his guidance during my most meaningful academic enterprise. His experience and persistence shaped the outcome of this project. I also wish to show my gratitude to my family, who supported me emotionally and financially through my schooling. Finally, I want to thank my friends, who backed me morally during this challenge.

Introduction

Data related innovations are drastically changing the way certain markets are behaving. Tech giants are using web platforms to disturb sectors from retail to the passenger transport industry. This connectivity opens new ways to conduct business by reducing communication costs and cutting the middleman between the consumer and the producer. The increase in available data and the emergence of artificial intelligence allow firms to optimize processes and target consumers in new ways. If these innovations can profit major corporations, they also strike down barriers of entry and permit small players to disrupt markets that were not accessible before. Crowdsourcing enables newcomers to enter a worldwide market of products and services by connecting consumers directly to the entrepreneur and bypass the need for a loan or other typical financing channels. These innovations offer new opportunities in economic research. First of all, some connected markets are new economic phenomena like crowdsourcing. Second, a lot of new data are free and accessible and permit us to study those markets with a new degree of detail. Third, machine learning models might help us to predict more precisely economic phenomena.

This thesis takes advantage of those three opportunities. First, we analyze the projects on Kickstarter, one of the main crowdsourcing platforms. We find that a few thousand serial creators made over five projects each. They potentially used this platform as an alternative way to sell their products instead of an initial financing opportunity. We use a two-stage least square regression to estimate if a project's duration has a causal impact on its probability of success. We find that duration has a significant and causal effect on

the success of the campaigns. An additional day reduces by two percent the likelihood of success of campaigns. We also use a logistic regression, with variables extracted from natural language processing, and evaluate the relative importance of these variables. We then use a series of machine learning models and benchmark their prediction power with different features extracted from natural language processing. The best models are the gradient boosting model and the extremely-randomized tree, and both of those models increase performances of approximately 16 % in accuracy compared to the econometric models. Moreover, the usage of features extracted from the project descriptions increased the performance of machine learning models of approximately 0.7 % of accuracy. This small increase is small buy consistent with other findings in the current literature.

The thesis is divided as follows. The next section contains the literature review and an explanation of the different models used. The third section contains the methodology used in this project. The fourth section is an extensive analysis of the database with information on the newfound serial creators. The fifth section shows the empirical results and analysis of the two-stage OLS, logistic regression, and the benchmark comparison of the machine learning model. The conclusion then closes the thesis. It is also worth mentioning that all the codes and datasets used in this project are available on GitHub.

Chapter 1

Literature review

The objective of this thesis is to demonstrate that the use of new machine learning and natural language processing (NLP) techniques allows us to predict more precisely economical outcomes of crowdfunding projects. The first part of the literature review explains the roots of crowdfunding. The second part describes natural language processing tools, some of them that can be used directly in econometrical models. The last section describes a variety of machine learning models that can predict the outcome of a project.

1.1 Crowdfunding

1.1.1 Definition of crowdfunding

To define crowdfunding, we first need to understand that it comes from crowdsourcing. According to Kleemann et al. (2008), it is "when a profit-oriented firm outsources specific tasks essential for the making or sale of its product to the general public (the crowd) in the form of an open call over the internet." In the particular case of crowdfunding, we refer to Belleflamme et al. (2014): "Raising funds by tapping a general public (the crowd) is the most important element of crowdfunding. This means that consumers can provide input to the development of the product, in this case, in the form of financial help. How the interaction with the crowd takes place may, however, differ from crowdsourcing." The

different forms of crowdfunding (CF) all serve the same purpose: allocate funds from generally a large number of funders, "the crowd" to a fundee. The internet platforms act as intermediaries and take care of the transactions. The most commons forms of CF are equity/royalty-based, lending based, reward-based and donation-based (Belleflamme et al. (2015)).

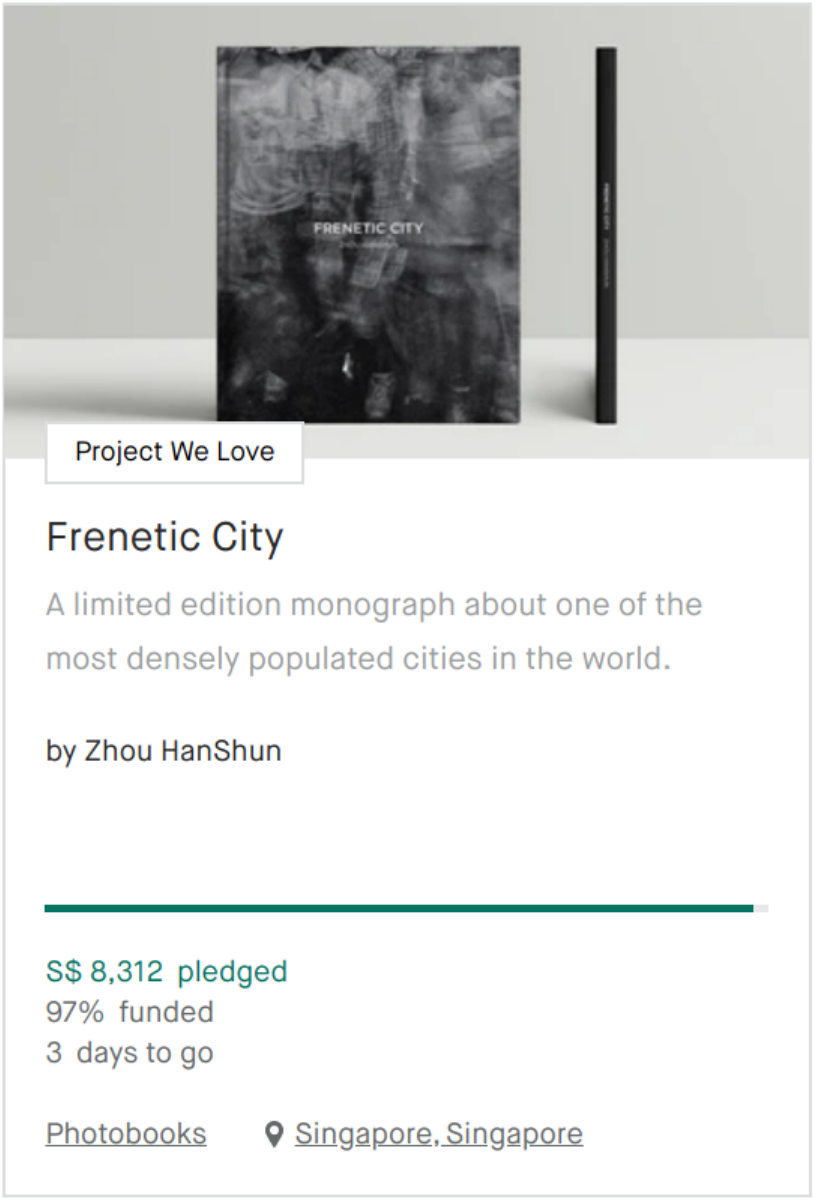
Equity and royalty-based CF are an alternative way for firms to raise venture capital. Equity-based CF trades funds for equity shares of the firm, whereas royalty-based CF exchanges funds for a percentage of the revenues or profits. Lending-based CF is a peer to peer lending for individuals or firms. Prosper.com is a well-known platform that allows individuals to lend money to other individuals with an interest rate fixed by the platform. Reward-based CF is a way to fund in exchange for non-monetary rewards. One of the most popular platforms is Kickstarter. The rewards take different forms. Most frequently, it is a pre-ordering of the good before the production started. Kickstarter hosts all kinds of projects, from tech products to artistic projects like plays, music albums or movies. The database we use in this research project comes from Kickstarter.

1.1.2 Specific elements about Kickstarter

Kickstarter is a multisided platform with the entrepreneurs (creators) and consumers (backers). When a creator launches a project, he sets a specific amount of money to raise (the goal) with a specific deadline. This platform works with an all-or-nothing mechanism. The website does not charge fees to the creator and refunds the backers if the campaign does not achieve the goal. When the campaigns achieve their goal, Kickstarter transfers the funds to the creator but keeps 5% as payment and charges the transaction fees directly to the creator. The payment transaction fees vary from 3 to 5%, depending on the country, as we can read on the website Kickstarter (2019a).

Projects can last from 1 to 60 days as of June 2011, but the maximum duration of projects was of 90 days during the first years of Kickstarter's existence. The duration of campaigns is the time between the launch of the campaign and the deadline, which

Figure 1.1: Screenshot of the Website



Note: A research on the website yields a series of small descriptions of this kind. The text shown is the Blurb and we can also see the amount already pledged and the remaining time to the campaign.

is decided by the campaigner at the launch and can't be modified afterward. The platform explains in a blog post on Kickstarter (2011) that the website decided to shorten the maximum duration of the projects. Kickstarter's analysts noticed a negative correlation between the duration of projects and the success rate. Kickstarter's analysts think there is a negative dependence between campaign length and success because, further deadlines make potential donors procrastinate on their pledge, and they potentially forget to donate.

Moreover, the platform is strictly reward-based, as it is prohibited to offer equity as a reward. Kickstarter (2019b).

1.1.3 Determinants of success

According to Chan et al. (2018), all the variables that impact the outcome of a project fall into three levels of effects: the geographical, the creator and the product level. The product level effects are related to the specific good or service of the project. The signals of quality, per example, have a positive impact on the outcome. Descriptions with fewer spelling mistakes and more compelling videos and photos increase donations. Moreover, each category of project exhibits its particularities, with success rates varying significantly between types of projects. Most of the creator level effects are latent variables such as degree of preparation, skills and reputation and can have a significant impact on the outcome of a project. Experience from previous projects can also change the result of a campaign. Geographical variables also influence crowdfunding campaigns. Even if there are little additional costs to contributing to a project from a distance, economic clusters and local networks can help the firm performance with the success spillover effect and increased information flow.

1.2 Natural Language Processing

1.2.1 Preprocessing

The recent democratization of machine learning techniques opens the door to new potential explanatory variables in models to predict the outcome of crowdfunding projects. Natural language processing, also known as NLP, allows integrating new sets of information that were not available. Netzer et al. (2019) use NLP to predict loan default on Prosper.com, a peer to peer lending platform. On this website, the "campaigner" tries to convince the crowd that he will not default on a loan. Similarly, the creator on Kickstarter tries to convince the crowd his product is worth it and that he will deliver it. The next section aims to explain NLP techniques usable to predict the outcome of CF projects. In this thesis, we analyze text from "Blurbs", a short description of the project to catch the attention of backers. To explain the concepts of NLP, we use a fictional sample of two Blurbs :

- Big and fun and unique project!
- Biggest project for fun!

In NLP terms, each of those blurbs, which are associated with different campaigns, is a "document". A "collection" is the aggregate of all the documents of a dataset; in this example, it is both Blurbs. The "dictionary" is a list of unique elements for each word used in the collection. In alphabetical order, the dictionary of our collection is:

1. and
2. big
3. biggest
4. for
5. fun

6. project

7. unique

Preprocessing has the objective of transforming noisy texts into usable data. As stated by Uysal and Gunal (2014), the four most common preprocessing techniques are tokenization, stop-word removal, lowercase conversion and stemming. The tokenization transforms natural language into numerical vectors by assigning a token (a number) to each present in the dataset. We then use the dictionary and assign a number to each unique term. We can use that tokenization to make a bag of words (BOW) representation of the text where each row is a document, and each column is a term. Table 1.1 is an example of the bag of words representation.

Table 1.1: Bag of Words Representation

Document	and	big	biggest	for	fun	project	unique
Big and fun and unique project	2	1	0	0	1	1	1
Biggest project for fun!	1	0	1	1	1	1	0

Stop-words are words that are used in any context like prepositions, articles, etc. Some researchers remove them from texts they analyze because those words don't bring additional information for text classification. Another preprocessing technique consists of changing all the letters to lower cases, so the classification algorithm does not consider the same word in upper or lower case as different. Stemming is another popular technique; it changes all the words into their root form. In our example, biggest would have the same token as big. The goal is to regroup words of similar meaning together. Those techniques remove some of the noise in the text before transforming the natural language to numerical vectors.

1.2.2 Term Frequency and Inverse Document Frequency

Term frequency (TF) and inverse document frequency (IDF) are ways to transform natural language into numerical vectors Zhang et al. (2011). TF-IDF is defined with the idea of weighting terms by using their occurrence.

$$w_{i,j} = tf_{i,j} \cdot \log\left(\frac{N}{df_i}\right) \tag{1.1}$$

Here the $w_{i,j}$ is the weight of the term i in the document j. df_i is the document frequency of the term i in the collection of all the documents and $tf_{i,j}$ is the term frequency of i in the document j. N is the number of documents in the collection. The intuition behind this weighting process is quite simple: if a word is frequent in a specific document but rare in the rest of the collection, it means it is an important word in this context. On the other hand, words that are very common in every document like "the", "a" or "and" are probably not very meaningful. Bag of words and TF-IDF yields very high dimensional outputs since each term has its column. This makes it hard to use it in econometric models, but it is usable in machine learning models. We use the fictional Blurb: "Big and fun and unique project". In this example, we have a total of two different Blurbs, and the information necessary to the tf-idf is in table 1.2. This process gives weight to uncommon words. In this example, "big" is the rarest word, so it is the term with the most significant weight in this document, even is "and" is there twice. We used the TF-IDF representation to train the latent Dirichlet allocation model for topic modelling. We explain the latent Dirichlet allocation in the next section.

Table 1.2: An Illustration of TF-IDF Computation

Blurb: "Big and fun and unique project!"

Variable	and	big	biggest	for	fun	project	unique
tf	2	1	0	0	1	1	1
N	2	2	2	2	2	2	2
df	3	1	1	1	1	2	1
w	-0.81	0.69	0	0	0.69	0	0.69

1.2.3 Latent Dirichlet allocation

The latent Dirichlet allocation (LDA) is a dimensionality reduction technique that regroups words into topics and then outputs the probability of each document to talk about a specific topic. The following section uses the work of Blei et al. (2003). In the article previously cited, they wrote: "The basic idea is that documents are represented as a random mixture over latent topics, where each topic is characterized by a distribution over words." So the LDA regroups words in clusters of topics; if some words are often used together in the same document, they are associated with the same topic. Once each has a probability of being part of each topic, the models output the probability of a document to assess a certain topic depending on the words present in a said document. The input of the LDA can be a bag of words representation or a TF-IDF.

Since LDA modelling uses a fixed number of topics, the said number of topics can be considered as a hyperparameter to be adjusted depending on the prediction task. To do so, we use the *perplexity* to evaluate the model. It is a commonly used metric for language modelling purposes. A lower perplexity means a better generalization from the model.

$$perplexity(D_{test}) = \exp\left(-\frac{\sum_{d=1}^{m} \log p(w_d)}{\sum_{d=1}^{m} N_d}\right) \tag{1.2}$$

In equation (2), $p(w_D)$ is the probability that a given word w appears in the document d. N_d is the number of appearance of the term in all documents. M is the number documents in the dataset. The perplexity is the equivalent of the inverse of the geometric mean of the likelihood of each word.

1.2.4 Other NLP high-level analysis

We also used the package Textblob [1] (Loria S., 2016) and textstat, which returns high-level information about the sentences. Textblob returns a sentiment score of the positivity in the sentence of the text, and it is trained on a data set of movie critique. It varies

[1] documentation at: https://textblob.readthedocs.io/en/dev/

between minus one and one, zero is a neutral description. Textstat [2] calculates readability indexes such as the gunning fog, flesh-Kincaid score and automated readability index that are widely used to evaluate the simplicity and ease of understanding of texts Kincaid et al. (1975). The **gunning-fog index** (also known as fog) uses a weighing of the average sentence per word and the ratio of complex words per sentence divided by the total number of words. It represents the necessary level of education to understand a given sentence. The **automated readability index** (ARI) is similar but is uses the average character per words and the average words per sentence. Finally, the **flesh-Kincaid** (also known as flesh) weights the average words per sentence and the average syllable per words. If all those readability indexes have one variable in common, they also all have one unique information that can help describe the text in another dimension. These scores can be used directly as input for any model.

1.3 Prediction with machine learning tools

The next section explains how to use machine learning models to replace the econometric models. This section also describes how to benchmark machine learning models against econometric models.

1.3.1 Goodness of fit

Kahovi (1995) compares popular re-sampling techniques and explains the process of model selection. An important element of model selection is the use of different metrics to compare the performance of the models. The accuracy of a model, in our specific case, a classifier is the number of correct predictions divided by the overall number of predictions.

[2] documentation at: https://pypi.org/project/textstat/

Table 1.3: Confusion Matrix

	Predicted True	Predicted False
Actual True	True Positive (T_p)	False Negative (F_n)
Actual False	False Positive (F_p)	True Negative (T_n)

Table 1.3 is known as a confusion matrix. In classification problems, different kinds of prediction errors can occur.

From the confusion matrix it is possible to compute many other performance metrics, such as the true negative rate and false negative rate and the accuracy.

$$\text{Accuracy}(1-Error) = \frac{T_p + T_n}{C_p + C_n} = P(C) \tag{1.3}$$

The root means square error (RMSE) is also a widely used to assess the performance of models in both econometric and machine learning.

$$\text{RMSE} = \sum_{i}^{N} (y_i - \hat{y_i})^2 \tag{1.4}$$

Where y_i is the actual value of the target variable, and $\hat{y_i}$ is the predicted value. It is also worth mentioning that in the case of a binary target variable, it is possible to obtain the accuracy from the RMSE and vice versa. Since the prediction error is either one or zero, and depends on the number of misclassification, the $RMSE = \sqrt{(1-accuracy)}$. The RMSE indicates the error of the model, so we want to minimize, while we want to maximize the accuracy.

$$\text{True Positive Rate} = \frac{T_p}{C_p} = P(T_n)$$

$$\text{True Negative Rate} = \frac{T_n}{C_n} = P(T_p)$$

The true positive rate (TPR), also known as sensitivity or recall, is the fraction of total positive that were classified as positive. Similarly, the true negative rate (TNR), also known as specificity is the fraction of negative that were correctly classified. The positive

predictive value (PPV) or precision is the fraction of total positive prediction that were correctly classified. The performance metrics defined above can be used to select a model depending on the prediction cost function, which is defined as follows:

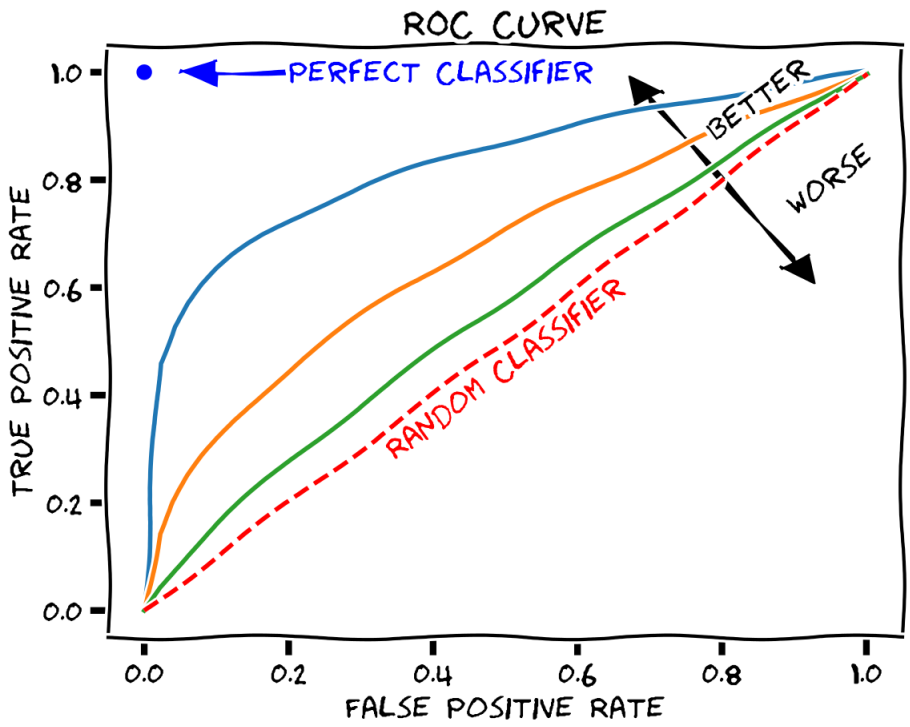
$$Cost = F_p \cdot C_{F_p} + F_n \cdot C_{F_n} \tag{1.5}$$

In the equation above, C_{F_p} is the cost of a false positive prediction, and C_{F_n} is the cost of a false negative prediction. The cost function is rarely explicitly formulated because it is mostly unknown in real-life situations. The general idea is that depending on the predictive task at hand, false-positive can be costlier than a false negative. In a credit risk problem, per example, we decide to lend or not to depend on the probability of default. In this case, a false negative costs the profit on the loan. A false positive, however, causes to lose the capital plus the interest. In this scenario, the PPV or TRN could be more pertinent metrics for model selection.

Another common way to assess prediction power and the robustness of the models, especially in binary cases, is the area under the Receiver Operating Characteristic (ROC) curve. Most models predict a continuous probability between zero and one for each observation. If the predicted probability is over the threshold 0.5, then the model predicts one, else it predicts zero. To plot the ROC curve, we compute the true positive rate and true negative rate for each threshold between zero and one. Figure 0.2 shows an example of a ROC curve.

The ROC curve allows evaluating the robustness of a model as the classification threshold varies. To change those multiple points to a single decision metric used in the model selection, we calculated the area under the ROC Curve (AUC). According to Bradley (1997), the AUC represents the probability that a random positive subject will be predicted higher than a random negative subject. Another widely used metric is the f-measure, which is the harmonic mean of the TPR and the PPV, but this metric is affected by the distribution of the classes Tharwat (2018).

Figure 1.2: Example of a ROC Curve



Note: The dotted line is a random classifier. Author: Martin Thoma, original image on this link.

1.3.2 The Bias Variance dilemma

Once we established which metric to use for model selection, it is essential to understand the bias and variance dilemma in the development and tuning of machine learning models. As explained by Geman et al. (1992), the bias of a model is related to its ability to predict accurately. When an estimator is, on average close to the actual value, it has low bias. The variance is about the sensitiveness of the model to the training data. A high variance model will accurately predict observations in the training set because it will react to the noise in the data and overfit. It will poorly generalize to other datasets since it uses noise to predict. When developing and/or tuning a machine learning model, the challenge is to balance both of those sources of error. The objective is to have a model that can both predict accurately and easily generalize to other datasets.

The choice of models is one of the first steps in the bias/variance arbitration. In general, simpler models, like linear or logistical regressions, will have higher bias but lower variance than complex machine learning algorithms. The same is true within different forms of the same kind of model. For example, a linear model with few linear variables versus a model with hundreds of variables will tend to have a lower bias and higher variance. The same principle applies to a machine learning model. A small neural network of one layer will typically have higher bias and lower variance than a more complex multi-layers network.

The number of observations in the training dataset has an impact on the bias/variance trade-off generalization. Simpler models will often require fewer observations, and fewer variables have a decent performance. Which model will be the best also depends on the dataset and the data generating process. When the data generating process follows a linear equation, the linear model might outperform sophisticated machine learning algorithms.

1.3.3 Goodness of Fit

It is possible to use cross-validation to assess the bias/variance dilemma and avoid overfitting. The idea is to randomly split the data into a training and a test set. The training dataset is used to train the model, and the test set is used to evaluate the performance of the model. According to Kahovi (1995) the test set is used to replicate real-world observations. The training set can be between one quarter and one-third of the dataset; the remaining observations are used to train and validate the model. There are a lot of decisions to make in the model design and model selection, and it is common to split the training dataset between a training and validation set. We then use the results on the validation set to design and select the model.

The k-fold cross-validation is an effective technique that consists of splitting the training set into k random mutually exclusive subsets. The next step is then to train and validate the model k times, each time using a different fold as a validation dataset and using the average result to choose the best model. Once again, as maintained by Kahovi (1995) ,

the k-fold cross-validation with a k between 10 and 20 is an efficient and reliable cross-validation technique.

1.3.4 LASSO and Ridge regularization

The least absolute shrinkage and selection operator is attributed to Tibshirani (1996). It is also known as LASSO or L_1 regularization. Ridge was formulated by Hoerl and Kennard (1970) is known as L_2 regularization. These two techniques are well-known tools for variable selection and bias reduction in machine learning. The least absolute shrinkage and selection operator (LASSO also called L_1) is a penalty applied on the regression coefficient. In the OLS context, the LASSO estimate is defined by the following optimisation problem :

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1.6}$$

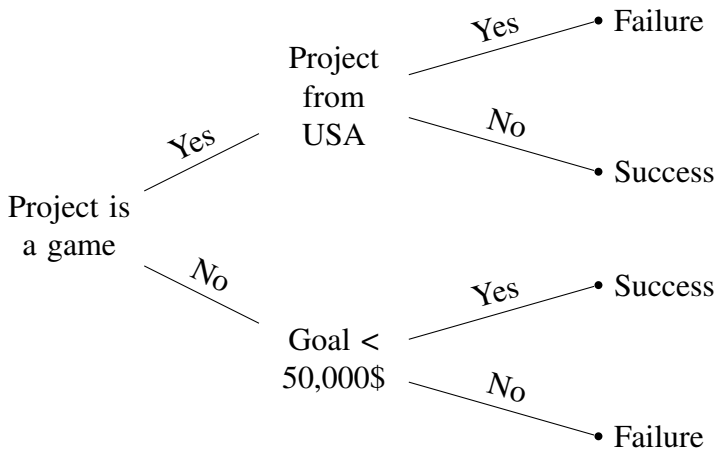
The Ridge constraint (also called L_2) is similar but uses the sum of the square of the coefficients. The only change is on the second term of the equation which is related to the constraint on the tuning parameter :

$$\min_{\beta} \sum_{i=1}^{n} (y_i - x_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{1.7}$$

In both cases, we define $\hat{\beta}$ as the coefficients estimates and $\hat{\lambda}$ as the shrinkage parameter that would result of the constraint. So depending of the L_1 or L_2 constraints we have respectively $\sum_j |\hat{\beta}_j| = \hat{\lambda}$ and $\sum_j \left(\hat{(\beta_j)^2} \right) = \hat{\lambda}$.

In each case, reducing the value of the parameter would reduce the values of the coefficients, but both of those techniques will yield different results. The LASSO will more drastically reduce the number of coefficients because it will quickly force their values toward zero. With both of those models, it is important to find the optimal value of the regularization parameter to allow for a parsimonious model without loosing too much prediction power. It is possible to use k-fold cross-validation to tune the regularization parameter.

Figure 1.3: Example of a Decision Tree Classifier



1.3.5 Trees

The decision tree classifiers (DTC) are non-parametric prediction tools. It is a powerful classifier, who is also at the center of other machine learning tools like Random Forests and Gradient Boosting Models. DTC can be described as « one of the possible approaches to multistage decision making » according to Safavian and Landgrebe (1991) where the idea is to compartmentalize a complex decision into simpler and smaller decisions. DTCs are composed of nodes and edges. Figure 0.1 shows an example of a decision tree classifier. Each leaf indicates the classification of the target variable, and each intermediate node is a split on a given feature.

According to Safavian and Landgrebe (1991), the design can be divided into three steps. First, we need to determine the optimal tree structure, then the feature selection rule at each internal node and finally, the decision rule to use at each internal node. Each of the aforementioned variables represents a different step in the tree design. There are four approaches of DTC design methods: top-down, bottom-up, hybrid and growing pruning. When it comes to the choice of decision rule strategy to split, the two most popular criteria are the Gini index or the entropy. The objective of any decision rule is to maximize the discrimination between classes. The Gini index is as followed:

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i) \tag{1.8}$$

In the equation above, $p(c_i)$ is the percentage of a class c_i in a given node. If the node is "pure", one of the $p(c_i) = 1$ and the Gini impurity index is null. The entropy criteria is as follow:

$$Entropy = \sum_{i=1}^{n} -p(C_i)\log_2(p(c_i)) \tag{1.9}$$

Once again, the objective is to minimize this formula. A purer node will yield a smaller entropy. A DTC uses one of those criteria and computes the chosen index for each node at each potential split. The DTC will choose the split that minimizes the sum of the chosen criteria. No splitting criteria are consistently better than the other (Raileanu and Stoffel (2004), so it is pertinent to try both.

1.3.6 Tree ensemble models

Ensemble Tree models combine multiple trees to generate a better model with generally lower variance. The three models we will be looking at are random forest and extremely-randomized trees (extra tree) and gradient boosting models (GBM). Both models generate M different trees and select the target variable by vote if it is a classification problem or average the prediction of each tree if the target variable is continuous.

The main hyper-parameters are:

$$K = \text{number of randomly selected independant variables to chose from}$$
$$n_{min} = \text{the minimum sample size for a splitting node}$$
$$M = \text{Number of trees}$$

Breiman (2001) defines a random forest as "a classifier consisting of a collection of structured tree classifiers.

$\{h(x,\Theta_k), k = 1, ...,\}$ where the $\{\Theta_k\}$ are independent and identically distributed random vectors and each tree casts a unique vote for the most popular class at input **x**. "
.

In other words, each tree is grown from a new randomly generated set. Each set is randomly drawn from with replacement from the original training dataset, and the trees are not pruned. Pruning a tree means removing the nodes of the tree that add little prediction power. An extension of the random forest also uses a random input selection (Forest-RI) for each tree.

Extremely-randomized trees, according to Geurts et al. (2006), are very similar to a random forest with the main exceptions that it uses the whole training sample to train the model. It randomly selects k input variables at each decision node and then selects a random cut off point. The GBM as stated by Friedman (2002) is an ensemble tree model with the main difference that the trees are trained sequentially. The model starts with a simple DTC and the additional trees are trained on with a reweighting of the observations. This reweighing gives more weight to the observations that were falsely predicted by the preceding trees.

1.3.7 Naive Bayes Algorithm

The Naive Bayes algorithm, as explained by Lewis (1998), is based on Bayes theorem.

$$P(C = c_k|X = x) = P(C = c_k)\frac{P(X = x|C = c_k)}{P(x)} \tag{1.10}$$

where

$$P(X = x) = \sum_{k'=1}^{e_c} P(X = x|C = c_{k'}) \bullet P(C = c_{k'}) \tag{1.11}$$

In the equations above, C is a random variable which represents one of the e_C classes of the target variable $(c_1, c_2, ..., c_k, ..., c_{e_C})$. X is a random variable which values are the vectors of the feature values of each document $x = (x_1, x_2, ..., x_j)$. Netzer et al. (2019)

used Naive Bayes to identify the words that are the most frequent in defaulted loans or repaid loans and then use those words as input for the other models.

1.3.8 Causality and Instrumental Variables

Ordinary least square models are often used to explain the effects of variables in economic phenomena, and under certain restrictive conditions, the coefficients can estimate causal effects of a variable x on a dependant variable y. In a social science context, however, omitted variables often make OLS estimators to be inconsistent in regard to causality. The two stages optimal least square model, according to Angrist and Imbens (1995), allows with an instrumental variable (IV) to estimate the average causal effect. The idea is to use an IV z to predict the value of x in a first-stage regression. The second stage of the regression uses the predictions of that variable $\hat{x}$ to predict y. In that second stage, the coefficient associated with $\hat{x}$ can estimate the average causal effect.

For an IV to be valid, and allow an unbiased causal effect, it needs to satisfy two conditions. First, z needs to have a significant impact on x. This is easily verifiable with an F test to know if the coefficient associated with the instrumental variable is significantly different from zero. The second condition is that the effect of z on y needs to be exclusively through the variation of $\hat{x}$. In other words, no omitted variables can affect both z and y, and this condition is not verifiable with statistical tests. Purely random shocks such as most random treatment variable with a placebo effect in a drug trial satisfy those conditions. In economic problems, random trial experiments are generally not realistic on a large scale, so researchers rely on other random shocks. Policy changes often satisfy those conditions and allow an unbiased estimation of the average causal effect. For example, Acemoglu and Angrist (2000) use variation in compulsory schooling laws to evaluate the causal impact of schooling in wages in the US. Oreopoulos (2006) also use a policy change in mandatory schooling laws to estimate the return of education.

Chapter 2

Data

The first section explains the source of the dataset. The second section of this chapter is an in dept explanatory data analysis of the dataset about the success of campaigns on the website, text descriptions of projects, duration of projects and serial creators.

2.1 Database source

The dataset used in this thesis comes from the website Web Robots[1], which specializes in web crawlers and data retrieval. The dataset is freely available on their website as part of one of their projects. They also made available a dataset with Indiegogo, which is another crowdfunding website.

Web Robots started scraping Kickstarter in April 2014, but the consistent and uninterrupted monthly scraping began in November 2015 (Web Robots, Kickstarter Datasets, 2019). Each dataset is a snapshot of the website on a specific day at a specific month. That means each picture has the projects that are still visible on the site, even if the campaign is over. We aggregated all those datasets and kept the most recent observation of each campaign to generate the final dataset.

[1] https://webrobots.io/

The variables available in the dataset are the start of the campaign, the deadline, the goal of the campaign, the total amount pledged, the number of donations, the Blurb, the geographical location of the project, the category of project, the name of the project and the name of the creator.

2.2 Explanatory data analysis

The first part is a description of the success metrics on the platform and the total frequency of projects per year. In the second part of this section, we make an analysis of the Blurb, the small text describing each project. In the third part of this section, we tale a look at the duration of the campaigns over time and how a policy change affected it. The last part contains stylized facts about the "serial creator", creators that made over five projects each.

2.2.1 Success variables and number of projects on the platform

In this section, we analyze the variable "goal", which is the monetary objective of the campaign. The project is successful if the goal is achieved before the deadline. The database has the variable state with the possible values cancelled and suspended, but we counted them as a failure. Success is the target variable for all the models. The frac-goal is the fraction of the goal that is achieved at the end of the campaign. We used backers-count and pledged (total amount pledged) to calculate mean-pledged, which we can use as a proxy for the mean price of the contribution. The table below shows the distribution of those variables. It is also worth mentioning that the currency varies from country to country and that the goal variable and the pledge variable are always in the same currency. The location variable captures the effect of the currency.

What we can see concerning the goal is that the distribution seemed skewed by extreme values since the mean is 37,260$, but the median is 5,000$, and even the third

Table 2.1: Success Metrics of Campaigns

Statistic	Goal	Success	Fraction of Goal	Backers Count	Pledged	Mean Pledged
count	295,394	295,394	295,394	295,394	295,394	264,732
mean	37,260	0.468	4.505	140	11,834	73
std	986,192	0.499	368.151	1051	109,741	152
min	0	0.000	0.000	0	0	1
25%	1750	0.000	0.011	3	60	25
50%	5000	0.000	0.400	20	1030	47
75%	15000	1	1.164	76	5344	83
max	1.0e8	1	155,320.390	219,382	20,338,986	20,573

Note: Fraction of goal is (total pledge amount/goal) and mean pledged is (pledged/backers count). It is not defined for the projects without contributors, that's why the count is different from the other variables. The maximum value of 155,320 is an outlier. This project is a documentary that had a goal of one pound sterling and £155,000 of pledge.

quartile (15,000$) is lower than the mean. Most projects have small achievable goals; the max is a hundred million, which does not seem an achievable goal. Since it's free to launch a campaign, a small minority of campaigns could be a random joke made by bored individuals, or it could be a publicity stunt. It's probably the case for the campaign with a hundred million dollar goal.

With a success rate of 47% and a median mean-pledged of 47 $, most projects are of a small scale. Here we purposely exclude the project without donations from the computation of the mean-pledged. We want to use that variable as a proxy for the average price of rewards in projects. As you can see in table 2.1 there are outliers; the problematic ones are concerning the "goal " variable. If many projects have a goal of 0$, that will make them easily successful and bias the real effect of "goal" on success. We will keep all observation in the result part, but we will assess it to make sure outliers do not disturb the results.

The data contains two variables for the category of projects. The main category has 15 possible values, and the sub-category has 169 possible values, and each sub-category only belongs to one main category. The "goal" is the funding objective that the campaigner sets at the beginning of the campaign and cannot change it in the process. Tables 2.2 shows the number of projects per year.

Table 2.2: Relative Frequency of Projects per Year

Year of Start Date	Nb Projects	Fraction of All Projects
2009	664	0.002
2010	6,012	0.020
2011	15,559	0.053
2012	25,910	0.088
2013	27,408	0.093
2014	44,424	0.150
2015	51,268	0.174
2016	37,661	0.127
2017	35,391	0.120
2018	31,448	0.106
2019	19,649	0.067

Note: The dataset starts in April 2008 and ends in August 2019. The first and last year are not complete.

We can see a peak of projects in 2015, but it is important to notice some projects are not visible a few months after the completion, so that would explain the few numbers of projects in 2009 and 2010 before the website started web scraping. The website was also at its beginning, so there were very few projects on the platform.

2.2.2 Description of the Blurbs

Table 2.3 shows a short description of the Blurb, from which all the NLP features are extracted. This text is a brief description or catchphrase, usually between 16 and 25 words. The objective of this text is to catch the backer's attention. Here are two examples of Blurbs extracted from the dataset.

- A book of photography documenting the history of Christianity in the Chinese port city of Dalian, a former colony and capital of Manchuria.

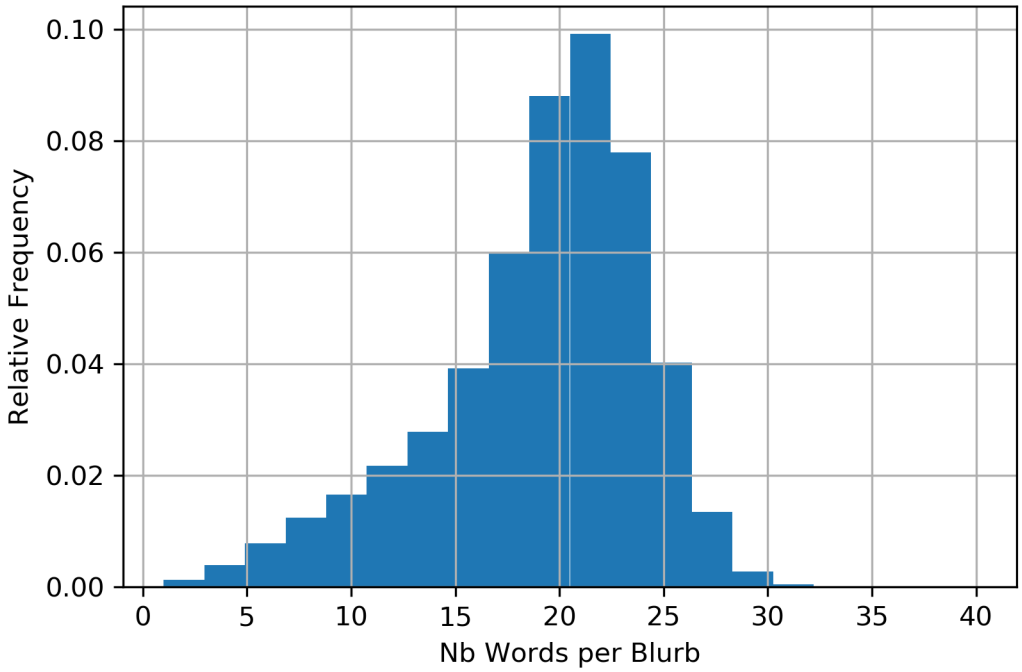
- 'Long time members of the CT ska scene in various aspects, newly formed band Steady Habits is looking to put their tracks on wax.'

Table 2.3: Numerical Description of the Blurb

Statistic	Nb Words	Nb Chars	Average Char per Word
count	295,394	295,394	295,394
mean	19.050	112.199	5.055
std	5.186	27.141	1.052
min	1.000	0.000	0.000
25%	16.000	101.000	4.478
50%	20.000	124.000	4.944
75%	23.000	132.000	5.476
max	85.000	150.000	135.000

Note: 25%, 50% and 75% are the values of the quartiles.

Figure 2.1: Distribution of the Number of Words per Blurb



For a more precise portrait of the Blurb, we plotted a histogram of the number of words in each text. For this plot, we excluded the observations with more than 40 words, because they were outliers.

We also computed the fog, ari, flesh score and sentiment analysis for each Blurb to describe the data in a different way. As mentioned in the literature review, the fog, ari

Table 2.4: Numerical description of NLP variables

Statistic	fog	ari	flesh-score	sentiment
mean	11.930	11.198	58.595	0.136
std	3.671	4.002	21.623	0.248
min	0.000	-16.300	-1,486.190	-1.000
25%	10.000	9.600	48.130	0.000
50%	11.810	11.400	60.310	0.100
75%	14.000	13.100	72.500	0.275
max	43.200	567.800	206.840	1.000

Note: The ari and flesh-score are not constrained scores and in small sentences they can produce extreme values, because of the subtractions and fraction in the formula. The sentiment analysis varies between one and minus one. A negative value means the sentiment is negative about the project.=

and flesh score are readability indices. A smaller value indicated a more straightforward text to understand, whereas a bigger value indicates the usage of more sophisticated or technical words. The sentiment feature indicates the degree of positivity, with a higher value indicating a more positive angle in the text.

2.2.3 Duration of projects

The project duration is one of the main choices when creators launch their campaign. The duration can vary between 1 and 90 days in the whole dataset, but as mentioned before, there was a policy change in 2011 to shorten the maximum length of campaigns to 60 days. The table below shows the distribution of the whole sample. The creator chose the campaign duration when he sets a deadline at the campaign launch. Once the campaign started, the duration cannot change. We thought about using the duration framework. We concluded that this framework is not appropriate since our dependant variable is set at the beginning of the campaign. Duration models are usually more appropriate to evaluate the probability of a change in the state of a dependant variable depending on the time elapsed, other independant variables and random shocks happening during the process. For example, the time before the death of a patient is unknown at the beginning of the process, contrarily to the duration of the campaign, which is set by the creator. The

negative dependence between the duration of campaigns and success can be explained by the fact that potential contributors procrastinate and have more chances to forget to donate if the deadline of the campaign is further away.

Table 2.5: Distribution of the Duration of Projects

Statistic	Before Policy Change	After Policy Change	Total
count	14,852	280,542	295,394
mean	42.623	33.026	33.509
std	19.443	11.534	12.237
min	1.000	1.000	1.000
25%	30.000	30.000	30.000
50%	35.000	30.000	30.000
75%	55.000	35.000	35.000
max	90.000	60.000	90.000

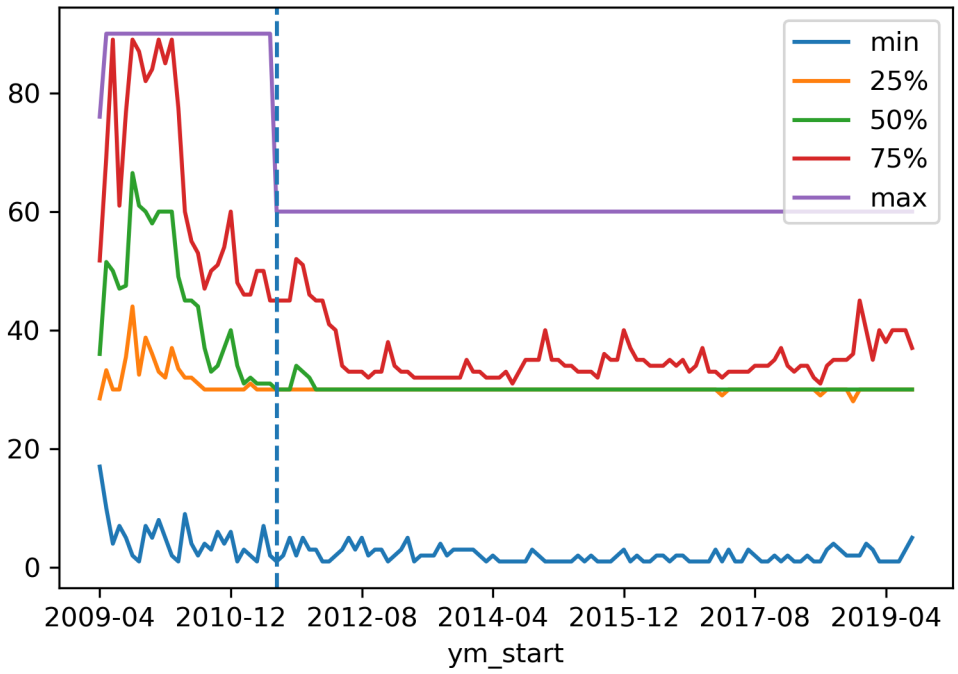
Note: The duration of the project is in days, like all subsequent analysis of duration. The first column has the statistics of the duration of projects before the policy change. The second column has the statistics after the policy change, and the total column is the aggregation of all the projects.

As we can see in table 2.5, the distributions of the durations of campaigns are different before and after the policy change. Before the policy change, the campaigns are, on average, ten days longer, and 25%. Moreover, the third quartile jumps from 55 days before the policy change to 35 days after the policy change. In total, the distribution is centred to 30 days, but there is still 25% of the sample with a duration of 35 days or more. For the policy change to be usable as an instrument, its shock must have a significant impact on the variable duration. To have another perspective on this issue, we plotted the average length of campaigns per month. The minimum, quartiles, maximum, mean and standard deviation are plotted for each month.

Before the policy change, there is a total of 14,852 projects in our sample. In total, 3414 projects lasted more than 60 days. That means 23% of the projects lasted more than 60 days before the policy change.

As we can see in figure 2.2.3, the distribution of duration is affected by the policy change. However, the duration of the project starts decreasing before that. A few elements could explain this phenomenon: first, since there were fewer projects in the platform at that

Figure 2.2: Average Duration of Projects Over Time



Note: Each point is a descriptive statistic of the average duration in days of all projects that started at a given month. 25%, 50% and 75% are the first, second and third quartile. The policy changed occurred at the blue vertical line on June 2011.

moment, the data are noisier and respond more to random shocks. Moreover, it is possible that the creator became aware of the negative correlation between project duration and success rate, and therefore made a shorter campaign. Even if the average campaign duration started decreasing before the policy change, there were enough 90 days projects to observe a change in the distribution of campaign duration. This context opens the door for an experimental setting.

2.2.4 Serial creators

When exploring the data, we noticed that some creators had many projects. The website's mission is to help the entrepreneur finance their first projects, so we were curious about

the thousands of project creators who had five projects and over. To understand if they were different from the other project creators, we made a more specific analysis of their case. At first, we wanted to know if they had a bigger success rate and what was the average length of their campaign. It is important to mention that if the creator makes a new creator account between projects, we cannot detect them. There is no reason to think creators do so. We think they can use previous projects' visibility as an argument to convince backers that they will properly deliver their products or services. However, a creator that fails in his first attempt might create new accounts to hide his past failure. In this case, we would not be able to detect them. However, it is unlikely that hidden serial creators represent a significant part of the sample. Detected creators with more than one project represent approximately 14 % of all the creators. Seventy percent of this 14% is from creators with only two projects. It is plausible that few creators keep trying and failing. However, it is unlikely that they represent an important part of the sample since they have no economic incentive to keep trying. Even successful serial creators are a small part of the sample.

As seen in table 2.6, the success rate increases with the experience of the creator and the duration of the project shortens. Moreover, the last columns shows the p-value of a t-test to evaluate if the average duration of projects is significantly different from one group to another. The small p-values indicate creators with more projects make projects that have a significantly shorter duration.

The following table shows the goal, frac-goal, mean pledge and months-per-project. The month-per-project variable is the average number of months between the beginning of a project and the beginning of the next project. Creators with more than five projects will probably spend less time on each project if they want to continue doing more projects.

The success rate and fraction of goal achieved increase as the number of projects increases. The median of the goal slowly decreases from 5,000 $ to a little bit over 2,000 $ for more than 20 projects. The mean pledged, and duration of campaigns also seems

Table 2.6: Average Success Rate and Duration of Creators in Function of Experience

Nb Projects per Creator	Nb Creators	Mean Success Rate	Mean Duration	p-value
(0, 1]	220988	0.413	34.3	0.000
(1, 2]	20765	0.505	33.3	0.000
(2, 3]	4845	0.613	32.3	0.000
(3, 4]	1751	0.679	31.8	0.000
(4, 5]	857	0.732	30.7	0.004
(5, 10]	1285	0.804	29.6	0.000
(10, 15]	244	0.833	27.4	0.000
(15, 20]	83	0.882	25.9	0.509
(20, 30]	58	0.895	23.1	0.038
(30, 40]	26	0.943	24.6	0.373
(40, 50]	5	0.991	20.0	0.540
(50, 150]	5	0.930	23.5	nan

Note: In this table, each point is a creator, like table 2.7. The first column indicates the total number of projects of a creator, and the second column is the number of creators with that experience. For example, 1285 creators made between 6 to 10 projects inclusively. The mean duration column is the mean of the average campaign length of each creator and not the mean of all the projects in that category of creators. We computed the mean success rate the same way. Finally, the last columns are the p-values of a t-tests to evaluate if each average duration of campaigns for each group is significantly different. A p-value smaller than 0.05 in the line i indicates that the group i and $i+1$ have different means with a certainty of a least 95%.

negatively correlated with the number of projects per creator. Serial creators apparently focus on small achievable projects. Moreover, to understand which kind of project serial creators make, we computed the relative frequency of different project categories and sub-categories. There are in total 15 categories and 169 sub-categories. The first column is the relative frequency of each category for the serial creator, and the second is the relative frequency of typical creators.

The top 3 categories are games, comics and arts, which are different from the typical top three, which is film & video, music and publishing. Serial creators might be different by their choices of parameters for their campaign, but also for the kind of projects they do.

Table 2.7: Median Goal, Fraction of Goal, Mean Pledge and Time Spent per Project in function of Experience

Nb Projects per Creator	Goal	Frac Goal	Mean Pledged	Months per project
(0, 1]	5,000	0.182	50.000	1.000
(1, 2]	5,350	0.763	46.385	5.000
(2, 3]	4,666	1.075	45.789	7.667
(3, 4]	4,000	1.300	46.227	8.000
(4, 5]	3,780	1.609	42.804	8.200
(5, 10]	3,958	2.231	41.882	6.714
(10, 15]	2,913	2.730	40.682	4.618
(15, 20]	2,568	3.960	43.513	3.882
(20, 30]	2,200	4.143	41.117	3.060
(30, 40]	2,234	4.847	47.736	2.293
(40, 50]	6,611	6.505	58.884	1.898
(50, 150]	4,166	1.553	40.137	0.736

Note: Like table 2.6, this analysis is at the creator level. Each column is the mean of the average value of the variable for each creator. The mean-pledged is used as a proxy for mean price. The months per project is the average number of months between the beginning of two campaigns of a creator. We set it by default as one for the group of creators with one project.

It is possible that serial creators are even more concentrated in sub-categories. Since they are 169 sub-categories, we will have a more specific idea of which kind of project they do.

Once again, serial creators are more concentrated in sub-categories. These categories are appropriate for iterative projects; for example, games can have extensions, or many versions and comics can have multiple volumes. Most creators want initial funding for an entrepreneurial project; the common usage of the website is not to have repeatedly produced new projects and use the platform as a selling platform. That would explain why serial creators are concentrated in iterative projects that are easier to do since they are adapted to their business model. To understand more specifically, which kind of projects are developed by both serial creators and common creators, we calculated the relative frequency of each sub-category. Since there is a total of 169 sub-categories, we only show the 15 most popular for both the serial creators and the common creators.

The most popular sub-category is by far the tabletop games, which is also one of the

Table 2.8: Market Shares of Project Categories: Serial vs Typical Creators Comparison

Project Category	Serial Creators	Typical Creators
games	0.265	0.108
comics	0.156	0.068
art	0.112	0.106
design	0.109	0.075
publishing	0.082	0.116
fashion	0.055	0.050
technology	0.050	0.099
film & video	0.048	0.117
music	0.041	0.115
crafts	0.023	0.029
photography	0.018	0.019
food	0.014	0.034
theater	0.012	0.031
dance	0.011	0.018
journalism	0.003	0.015

Note: This table is project level. The first columns are the market shares of project categories of all projects that have creators with 5 projects of more. The second column are the market shares of all the other projects.

most popular categories in the overall pool of projects. By looking at sub-categories, product design is also one of the most popular sub-categories in both samples. The distribution of projects is very different in the top categories, but it also seems much more concentrated in specific categories in the serial creator sample. To understand if it is really the case, we computed the Herfindahl-Hirschman index for both samples for the categories and sub-categories. The relative frequency of each category is similar to a market share of the projects on the website and we wanted to confirm if the difference was negligible or not.

As we can see, both categories and sub-categories are more concentrated in the serial creator sample. The main categories are overall more concentrated; it makes sense since there are fewer categories then main categories. The serial creators are more concentrated in specific categories, which overall are types of projects that are more easily iterative. This points to the fact that they are different kinds of entrepreneurs.

Table 2.9: Market Shares of Project Sub-categories: Serial vs Typical Creators Comparison

Project Sub-category	Serial Creators	Typical Creators
games/tabletop games	0.211	0.055
design/product design	0.090	0.071
comics/comic books	0.066	0.021
fashion/accessories	0.037	0.018
art	0.035	0.026
art/illustration	0.035	0.018
games/playing cards	0.032	0.011
comics	0.031	0.010
comics/graphic novels	0.030	0.019
publishing/fiction	0.026	0.023
games/video games	0.021	0.037
design/graphic design	0.017	0.011
comics/webcomics	0.014	0.006
crafts	0.013	0.011
publishing/children's books	0.012	0.021
film & video/documentary	0.008	0.025
film & video/narrative film	0.008	0.019
fashion/apparel	0.007	0.016
technology/apps	0.005	0.019
music/country & folk	0.004	0.022

Note: This table follows the same structure as table 2.14, but the columns do not sum to one, since only the sub-categories with most projects are shown.

Table 2.10: Herfindahl-Hirschman Index for Categories and Sub-Categories: Comparison Typical vs Serial Creators

Statistic Name	Sub-categories	Main Categories
Nb Categories	169	15
Equal Shares HHI	0.006	0.067
Typical Creators HHI	0.019	0.090
Serial Creators HHI	0.067	0.136

Note: The HHI are calculated from market shares of each category and sub-categories.

Chapter 3

Methodology

One of the objectives of this paper is to benchmark the usage of machine learning tools in comparison to econometrics models. To do so, we run a series of models with different feature sets, both with the econometrics models and machine learning models. The first part of our result is the estimation of the causal impact of the duration of campaigns on the probability of success of projects. In the second part of our project, we use logistic regressions to understand the data-generating process and the impact of NLP variables on success. The third part is about the performance comparison of the econometric and machine learning models. Moreover, all the programs and databases used for the next section are available on GitHub.

3.1 Causal relationship between duration of campaign and success

We use the instrumental variable framework in a two-stage least square regression to estimate the average causal effect of the duration of a campaign on the probability of success.

3.1.1 Prediction of campaign length

The first step of the 2SLS has for objective to predict the duration of projects. The first linear regression contains all the explanatory variables used in the second stage plus the exogenous shock on the duration of projects. The shock was a policy change in June 2011 when the website shortened the maximum duration of the campaign from 90 to 60 days. To have a different perspective on the impact of the policy change on the campaign length, we also estimate a negative binomial regression with the same set of independent variables on the duration. The negative binomial regression is more appropriate to estimate discrete data, such as the duration of a campaign in days. We compare the marginal effect of the policy change in the negative binomial with the first stage regression effect to know if both estimates are consistent with each other.

3.1.2 Prediction of Success in the instrumental variable framework

In the second stage, we used the predicted duration and the square value of predicted duration to allow for non-linearity for the impact of duration on the probability of success. We use information related to geographic, temporal and product-level information about the campaign. The variables concerning the parameter of the campaigns are the natural logarithmic transformation of the goal, the campaign duration and the category of the project. There are also two geographical variables, such as the location and the relative importance of the location on the platform. Additionally, we use the total number of projects of the creator, a month indicator, a period and square period variable to allow for the quadratic effect of the trend and the growth of new campaign on the website.

3.2 Logistic regression as benchmark

The first logistic regression has the same independent variable as the second-stage linear model, except we use the empirical duration instead of the predicted duration of the first stage. We also use an interaction variable between the policy change and the duration of

the projects. The second logistic regression uses additional variables related to NLP. We used the word count, fraction of words with six letters or more, the ari, fog, flesh score and sentiment analysis. As mentioned in the literature review, the first three indicators are readability indexes that indicate the level of complexity of the sentences. The sentiment analysis returns a level of "positiveness" of the text.

The logistic regression is useful to understand the underlying data generating process between the dependant variables and the success of campaigns. We calculate the marginal effect of those variables to understand the relative importance of each variable. We also compute different performance metrics to understand how the models perform differently.

3.3 Machine learning methods

Similarly to the econometric models, we use different feature sets. The first dataset does not have any features based on text. The predictive variables are all based on the economic and geographical aspects of the projects. The two other datasets contain additional features extracted from the Blurb. Then we train these three datasets on machine learning models to compare the performance metric of each model and evaluate if the prediction power increased on the test set with the text features.

Our goal is to predict the success of a campaign, so we use the condition "converted pledged amount" >= "goal" to define a successful campaign. We generate the features from the variables that are available before or at the campaign launching. Since the models use NLP features, we exclude non-anglophone countries from the dataset. The countries left in the dataset are the United-States, Great-Britain, Canada, Australia, New-Zealand and Ireland. Table 3.1 shows the number of project per country.

3.3.1 Non-nlp features

The variables that do not depend on natural language processing can be divided into four categories: the campaign characteristics, the temporal variables, the geographical vari-

Table 3.1: Number of Projects per Country

Country	Project per country	Relative frequence
AU	7,345	0.025
CA	14,268	0.048
GB	32,030	0.108
IE	925	0.003
US	240,826	0.815

Note: The US represent most of the projects.

ables and the mix of both temporal and geographical variables. The variables used in the econometric models are used in machine learning models. However, we also added their square and cubic transformation when the variable is not a fraction or a binary variable. Each project also belongs to a sub-category of product or service. We include these features as factor variables to capture each category-specific effect.

The geographical variables are there to capture regional effects at the country, regional and municipal levels. As mentioned in the literature review, people tend to fund local projects, and some regions fund specific categories of projects. Since we had the information concerning the location of the projects and the launched date, we created a variable that is the count of projects already launched in a region. The objective is to capture the local trend and cycle. Each region may go by the same cycle of rapid growth when Kickstarter is popularized in an area, and then the number of projects stabilizes. The local market for a campaign might behave differently depending on when in the cycle the project is. The number of new projects each month can capture this effect, but the variation in the percentage of a number of projects can be a better representation of the growth in a project with respect to the population or economic size of the geographical reason. We compute those variables for all the possible geographic levels in our dataset, which are the city, region and countries, to capture all the possible effects in our machine learning models. We also calculate the ratio of the number of projects compared to the other geographical unit to understand the relative importance of a town in a country.

3.3.2 Second feature set

The second feature set has all the information from the dataset plus additional NLP features. The objective is to test if a "simple" NLP feature adds predictive power on the test set. We used the variables from the article "When Words sweat" (Netzer & Al., 2019) as a baseline and added other common NLP features. Some of them are from simple count from the Blurb. First, we generated the character count from each Blurb, the average number of characters per word and the average number of words per Blurb. These variables are to capture what is the optimal length of the Blurb. We also use sentiment analysis to output a value of positivity for each Blurb and different metrics to describe the complexity of the sentences such as the ari, fog and flesh score. Those metrics are supposed to capture the same effect, but some of them might be more powerful than others to predict the success of the campaign.

We also use the TF-IDF representation of the Blurb as input. To select the words we use as input, we train a naive Bayes model with only the TF-IDF features to predict the success. We then select the 400 words that are the most correlated with success or failures to use as a feature in my final model.

3.3.3 Third feature set

The LDA model output the topic probabilities of each Blurb for each topic. In our context, there were two hyperparameters for this unsupervised learning model. The first one is the input: we can use either the bag of words or the TF-IDF representation of the text. The second hyperparameter is the number of topics to use. We tried both inputs and ran the models using between 1 and 200 topics and plotted the perplexity at each point. We select the parameter combination with the elbow method on the perplexity. To train the LDA model and output the probability of each Blurb, we used the Python library pyLDAvis [1].

[1] https://pyldavis.readthedocs.io/en/latest/

3.3.4 Model training

We use the Statsmodels[2] and Linearmodels[3] libraries in Python for the econometric models, since it already had all the necessary statistical tests already implemented. We used the whole dataset to train the econometric models since these simple models with high bias have, in general, low variance and little chance of overfitting. We also wanted to compare the econometric approach, in which it is very common to train on the whole dataset, to the machine learning models.

With the three feature sets, we train all my models and store the performance metrics on the test set for each model on each feature set. In all the model training, the training set represents 80% of the sample, and the test set is 20% of the sample. We used regularized logistic regressions, decision tree, random forest, extremely randomized trees and gradient boosting model (GBM). To fit the regularized logistic regression, we used both LASSO and Ridge regularisation. For the tree-based models, we trained models by using both the Gini and the entropy criteria as a splitting rule to see if that would have a major impact on the performance metric. We use cross-validation to tune the regularized logistic regression. For all the models, except the GBM, we used the scikit-learn library [4] to train the models and predict the target variable. For the GBM model, we used the Light GBM library, [5] with which it is possible to use the same API as scikit-learn, but it has more option to tune the GBM.

The model that often outperforms the other ones in machine learning is the GBM. We used Bayesian optimization and the Python library Hyperopt [6], which is a discrete optimization library, to auto-tune the model. We first define all the possible values of each of the hyperparameter; then the package used a discrete optimization algorithm to select the best hyperparameters. We integrated the tune GBM directly in the rest of the pipeline to compare its performance to the other models.

[2]https://www.statsmodels.org/stable/index.html
[3]https://pypi.org/project/linearmodels/
[4]https://scikit-learn.org/stable/
[5]https://lightgbm.readthedocs.io/en/latest/
[6]https://github.com/hyperopt/hyperopt

Chapter 4

Empirical Results

This section contains three parts. First, we use a two-stage least square model with an instrumental variable to assess the causal relationship between the duration of the projects and the probability of success. Second, we use logistic regression as a benchmark for the machine learning models, and we evaluate the impact of the other variables on the success of the campaign. Thirdly, we compare the performance of machine learning models to the econometric approach.

4.1 Causal Effect of the Duration on the Success Rate

The following section explains how we used a policy change in reducing the maximum duration of projects from 90 days to 60 days as an instrumental variable. We used this exogenous shock on the duration of projects to evaluate the causality between the duration of projects and the probability of success of campaigns.

4.1.1 Simple OLS before and after policy change

Before we get started in our models concerning the causal relationship between duration and success, we need to access an issue concerning the correction of the coefficients' standard error. Some projects in the dataset are from the same creators, but we do not

observe the creators in any regression for computational reasons. Usually, we should use a cluster correction on the standard errors of the coefficients of all models to assess this problem. However, the routines used in Statsmodels do not offer this correction. Since most coefficients are strongly significant in our result section, and only a minority of projects are from serial creators, we are confident that the cluster correction would not make a material change in our results. We did run a logistic regression and filtered out projects that were not the first project of a creator to compare the coefficients of the regressions. The result of this regression is in the appendix in table 1. The coefficients of the base logistic model in table 4.8 have very similar coefficients. With all those elements, we are confident that our results are reliable even if we did not make any cluster correction on our standard errors.

Before using the two-stage OLS, we want to know if the duration has a direct impact on success. We also want to know if said impact changes before and after the policy. So we compute twice the same regression, but with different samples. The first one is the OLS regression with all the projects before the policy change, and the second one is with all the projects after the policy change.

As shown in table 4.1, duration has a significant impact of -0.0022 on the probability of success. Duration of campaigns typically between 30 and 60 days, as shown in table 2.5 in the data description section and the standard deviation is 12 days. That means duration typically has an effect of 2.6% on the probability of success with a shock of one standard deviation. To know if the policy had any effect on the impact of duration on success, we want to compare the coefficient of duration in the sample of the project after the policy change.

As seen in table 4.2, the coefficients of both regressions are of similar magnitude and sign, except for the growth-new-projects variables. This sample was before the website reached become mainstream, so that could explain why the effect of new projects is different in the first regression. The regression with the observation previous to the policy change might be slightly different from the rest of the sample partly because there are fewer observations, so the coefficients might be responding to noise in the data.

Table 4.1: Sample Before the Policy Change: OLS Regression

Model:	OLS	Adj. R-squared:	0.204
Dependent Variable:	success	AIC:	17144.0851
RMSE:	0.5850	BIC:	17805.7975
No. Observations:	14852	Log-Likelihood:	-8485.0
Df Model:	86	F-statistic:	45.28
Df Residuals:	14765	Prob (F-statistic):	0.00
R-squared:	0.209	Scale:	0.18463

Variable	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
intercept	1.0041	0.0387	25.9149	0.0000	0.9282	1.0801
log(goal)	-0.0673	0.0030	-22.2373	0.0000	-0.0733	-0.0614
duration	-0.0022	0.0002	-11.4968	0.0000	-0.0026	-0.0019
nth-project	0.0121	0.0025	4.8007	0.0000	0.0071	0.0170
period	0.0319	0.0024	13.5601	0.0000	0.0273	0.0365
period-square	-0.0008	0.0001	-10.5669	0.0000	-0.0009	-0.0006
ratio-nth-city-country	0.3062	0.0564	5.4271	0.0000	0.1956	0.4167
lfd-new-projects	0.0463	0.0161	2.8690	0.0041	0.0147	0.0779

Note: The variables location, month and project category are used as control variables but not shown. Specific analysis of the coefficients are made in subsequent models.

4.1.2 Policy change as exogenous shock

There are two necessary conditions for the policy change to be a valid instrumental variable. First, it needs to have a significant impact on the duration of the project. Secondly, the effect it has on success can only be through the duration of projects and the other variables in the regression. The question is, did the policy change affect the probability of success by any other mean than through the impact of project duration. Since this policy affected only the duration of the projects, no other influence mechanism evidently emerges. Moreover, the policy was announced with a blog post containing a data analysis about the duration of campaigns and pledge distribution. The website explained there was an empirical negative correlation between the length of the campaigns and the success rate, like we observe in the OLS regressions. Once this information was made public, a campaigner more prepared and informed could choose to make a shorter campaign. If the choice of campaign duration is correlated with the level of preparation of the creators,

Table 4.2: Sample After the Policy Change: OLS Regression

Model:	OLS	Adj. R-squared:	0.166
Dependent Variable:	success	AIC:	354523.3407
RMSE:	0.5707	BIC:	355440.7103
No. Observations:	280542	Log-Likelihood:	-1.7717e+05
Df Model:	86	F-statistic:	650.9
Df Residuals:	280455	Prob (F-statistic):	0.00
R-squared:	0.166	Scale:	0.20712

Variable	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
intercept	1.6882	0.0099	171.0269	0.0000	1.6688	1.7075
log(goal)	-0.0569	0.0006	-102.9060	0.0000	-0.0580	-0.0558
duration	-0.0049	0.0001	-63.3091	0.0000	-0.0050	-0.0047
nth-project	0.0198	0.0003	58.2455	0.0000	0.0191	0.0205
period	-0.0136	0.0002	-64.3032	0.0000	-0.0140	-0.0132
period-square	0.0001	0.0000	58.9336	0.0000	0.0001	0.0001
ratio-nth-city-country	0.2087	0.0113	18.4435	0.0000	0.1865	0.2309
lfd-new-projects	-0.0949	0.0054	-17.5026	0.0000	-0.1055	-0.0843

Note: Same regression as in table 3.1, except we use the projects that started after the policy change.

the latent variable "level of preparation" influences both the length and success rate of the campaign. This latent variable is the primary justification for an instrumental variable, but it does not change its validity since the additional information impacts only the duration of the projects.

4.1.3 First Stage

The instrumental variable needs to have a significant impact on the duration of the campaigns. The first stage of the two stages regression gives information about that. We used all the independent variables used in the second stage plus the policy change to estimate the duration of projects. Table 4.3 only shows the variables of interest. The coefficients associated with the other control variables are hidden.

Table 4.3: First Stage: OLS Regression

Model:	OLS	Adj. R-squared:	0.095
Dependent Variable:	duration	AIC:	2288447.4284
RMSE:	11.6385	BIC:	2289369.2861
No. Observations:	295394	Log-Likelihood:	-1.1441e+06
Df Model:	86	F-statistic:	362.1
Df Residuals:	295307	Prob (F-statistic):	0.00
R-squared:	0.095	Scale:	135.49

Variable	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
intercept	32.8875	0.1984	165.7678	0.0000	32.4987	33.2764
log(goal)	1.5086	0.0136	110.6595	0.0000	1.4819	1.5353
policy-change	-6.7036	0.1444	-46.4260	0.0000	-6.9866	-6.4206
nth-project	-0.4242	0.0086	-49.3980	0.0000	-0.4411	-0.4074
period	-0.1851	0.0051	-36.5680	0.0000	-0.1951	-0.1752
period-square	0.0012	0.0000	37.9878	0.0000	0.0012	0.0013
ratio-nth-city-country	-0.9673	0.2826	-3.4227	0.0006	-1.5213	-0.4134
lfd-new-projects	1.5806	0.1277	12.3760	0.0000	1.3303	1.8310

Note: The variable lfd-new-projects is the growth of projects on the platform. Ratio-nth-city-country is the ratio of (number of projects launched in city)/(total projects launched in the country). Period starts at one for the first month in the sample and increases of one each month. Period-square is the squared value of the period. Months, location and project category are included as control variables but are not shown. See table 2 in the appendix.

The policy change has a strong impact on the duration, decreasing the duration of projects by 6.7 days. To assess the possibility of a weak instrument, we performed an F-test to test the hypothesis that the coefficient is significantly different from 0. The F statistic was of 12,245, which is high enough to exclude a weak instrument. The coefficient associated with the logarithmic transformation of the goal is positive, which means creators with bigger objectives tend to have a longer duration of projects. This result is interesting because it shows creators have the wrong impulse to increase the duration when they have more ambitious goals. The coefficient of the nth-project is negative, which means more experienced creators tend to choose a shorter campaign. The variable ratio-nth-city-country represents the relative importance of the city in the Kickstarter scene and can also be a proxy for the relative importance of a city in the country. Projects based in bigger cities have shorter campaigns than the ones in small towns. Lastly, the lfd-new-

projects is the growth of a new project in a month. If there are a lot of new projects on the platform, creators make longer campaigns. Most creators probably don't know that longer campaigns are less successful. It is possible that a lot of newcomers and least prepared creators chose less wisely the duration of their projects. The second stage will allow us to know if the negative coefficient of duration on success is attributable to a causal relationship, or it's simply a correlation without causality.

4.1.4 Negative Binomial Regression

We also wanted to use a type of regression that is adapted to discrete data. So to compare the results of the first stage, we ran both a Poisson regression and a negative binomial. Those regressions yielded very similar results in terms of performance and coefficient values. We only show the negative binomial regression in table 3.4, since it has less restrictive on the variance of the variables in the dataset than Poisson regression.

Table 4.4: Negative Binomial on Projects Duration

Model:	NegativeBinomial	Pseudo R-squared:	0.014
Dependent Variable:	duration	AIC:	2253735.7381
RMSE:	11.6229	BIC:	2254668.1918
No. Observations:	295394	Log-Likelihood:	-1.1268e+06
Df Model:	86	LL-Null:	-1.1423e+06
Df Residuals:	295307	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	4.0000		

Variable	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
intercept	3.4562	0.0057	602.2146	0.0000	3.4450	3.4675
log(goal)	0.0453	0.0004	114.4627	0.0000	0.0446	0.0461
policy-change	-0.1718	0.0041	-41.6724	0.0000	-0.1799	-0.1638
nth-project	-0.0159	0.0003	-60.3214	0.0000	-0.0164	-0.0153
period	-0.0053	0.0001	-36.2829	0.0000	-0.0056	-0.0050
period-square	0.0000	0.0000	37.6769	0.0000	0.0000	0.0000
ratio-nth-city-country	-0.0330	0.0081	-4.0585	0.0000	-0.0490	-0.0171
lfd-new-projects	0.0376	0.0037	10.2661	0.0000	0.0304	0.0448

Note: The signs of the coefficients are consistent with the linear model. We used the same independent variables as the regression in Table 4.3

Table 4.5: Average marginal effects of variables on duration in Negative Binomial

Variable	dy/dx	std err	z	P> |z|	[0.025	0.975]
ln(goal)	1.4337	0.013	110.355	0.000	1.408	1.459
choc	-5.4337	0.134	-40.637	0.000	-5.696	-5.172
nth-project	-0.5017	0.008	-59.156	0.000	-0.518	-0.485
period	-0.1676	0.005	-37.070	0.000	-0.176	-0.159
period-2	0.0011	2.86e-05	38.513	0.000	0.001	0.001
ratio-nth-city-country	-1.0450	0.258	-4.054	0.000	-1.550	-0.540
lfd-new-projects	1.1882	0.115	10.322	0.000	0.963	1.414

Note: The marginal effects are consistent with the coefficients of the first stage regression.

We also computed the average marginal effect of the policy change on the duration, which is -5.3974. That means the policy change reduced by more than five days the campaign length of the median observation. The results of average marginal effect in table 4.5 are consistent with the coefficient of the first stage regression since the impact of the policy change estimated in both regressions is of the same sign and similar magnitude.

4.1.5 Second Stage

The second stage shows that the predicted duration variable is significant and has a negative impact on the probability of success. The table 4.6 shows the results of the second stage regression with a linear relationship between the duration and success of projects. We used the same explanatory variables as in the first stage and used the predicted values of duration as independent variables. The policy change variable is implicit in the predicted value of duration.

The coefficients of the other regressors are consistent with the theory mentioned in the literature review. Smaller goals have more chances of success. Nth-project is positively correlated with success, which indicates a return of experience. There are approximately 1.3% more chances of success per project. The concave quadratic function of periods indicates there was a peak in terms of the success rate of projects at some point in the sample. The ratio-nth-city-per-country is positive, which means projects from important

Table 4.6: Linear Duration: Second Stage

Dep. Variable:	success	R-squared:	0.0376
Estimator:	IV-2SLS	Adj. R-squared:	0.0373
No. Observations:	295394	F-statistic:	5.524e+04
RMSE:	0.5907	P-value (F-stat)	0.0000
Time:	19:54:52	Distribution:	chi2(86)
Cov. Estimator:	robust		

Variable	Parameter	Std. Err.	T-stat	P-value	Lower CI	Upper CI
intercept	1.9861	0.0314	63.327	0.0000	1.9246	2.0475
duration	**-0.0198**	**0.0010**	**-20.665**	**0.0000**	**-0.0216**	**-0.0179**
log(goal)	-0.0352	0.0015	-22.810	0.0000	-0.0382	-0.0322
nth-project	0.0130	0.0006	21.269	0.0000	0.0118	0.0142
period	-0.0134	0.0004	-37.232	0.0000	-0.0141	-0.0127
period-2	7.876e-05	2.278e-06	34.567	0.0000	7.429e-05	8.322e-05
ratio-nth-city-country	0.2085	0.0379	5.4982	0.0000	0.1342	0.2828
lfd-new-projects	-0.0709	0.0057	-12.396	0.0000	-0.0822	-0.0597

Note: The duration is the predicted duration from the first stage regression. The other independent variables are the same except for the policy change, which is absent. See table 3 in the appendix.

cities have more chances of success than projects from smaller cities. A more important growth of projects on the platform has a negative impact on success. That could mean additional competition on the website makes it harder for projects to be funded if creators are growing more quickly than the traffic of contributors on the website. Unfortunately, we do not have data about the monthly traffic on the website.

The duration of projects has an average causal effect of 0.02, which is very economically significant because projects can last, to this day, between 1 to 60 days. With the duration used this way in the model, we assume a linear relationship between the duration and success. If we interpret this coefficient, we should choose an optimal duration of one day, since the coefficient is negative. This conclusion is not satisfactory economically speaking, and the relationship between campaign length and success is probably not strictly linear. However, since there are very few projects that last less than 30 days, we think that this negative relation is due to longer projects that fail and not very short projects that are very successful. However, we ran the second stage regression with both

the instrumentalized duration and squared duration. The results are in table 5 in the appendix. We did not use those results for two reasons. First, the routine we used in the python package did not allow us to use the instrumentalized variable's squared value. So we had no way efficient way to correct the standard error of the coefficients of the second stage. Moreover, it is not clear that the two-stage model predicts a valid causal effect if we use the squared value of the instrumentalized variable. We were unable to find an example in the literature that supported that. We think the linear second stage gave us interesting insight concerning the causal effect of duration on the probability of success. Finally, the 2SLS has an accuracy of 0.67 and an AUC of 0.75. The overall performance is still pretty poor, but the objective of these models is to estimate causal relationships.

Finally, the 2SLS has an accuracy of 0.67 and an AUC of 0.75. The overall performance is still pretty poor, but the objective of these models is to estimate causal relationships.

Table 4.7: Performance of the Second Stage Regression

Performance Metric	Value
Accuracy	0.671
AUC	0.735
RMSE	0.574

Note: Those are the performance metrics on all the observations since we typically do not split the datasets between training and test set in econometrics.

4.2 Econometric Models Coefficients and Performance

The following section assesses the econometric models used to predict the success of the projects. The interpretation of the coefficients allows us to understand better the impact of the variables on the probability of success. Moreover, we use NLP variables in the logistic models to understand if those variables can increase the prediction power and help

us understand the data generating process. Finally, we evaluate the performance of the models, which we can compare to machine learning models in subsequent sections.

4.2.1 Basic logistic model

Table 4.8 shows the coefficients associated with the variables to predict the probability of success of campaigns. The variables duration and period respectively have the square value to allow for quadratic relations. The units vary for these variables. The variables category, location and month-creation, are dummy encoded variables. The reference category for product categories is film and video. The location of reference is California, and July is the month of reference. We picked those categories as the reference category because they are the most frequent.

All variables, except for a few control variables, are significantly different from zero. The goal, duration and period variables all have a negative and significant impact on the probability of success. The number of projects of the creator has a positive and significant impact on success. Months of creation, location and category all have a significant impact on the project outcome. The next table shows the marginal effect of a standard deviation increase of the independent variables on the probability of success of the project. We calculate the marginal effect at the median.

Moreover, as mention in chapter 2, outliers could be cause for concern if many projects have a goal of 0$ or even 1$. These campaigns would be a systematic success and make an inverse relationship between goal and success in the data even if this relation is not present in the rest of the observation. We kept outliers in all of our models, but to make sure these outliers did not disproportionately impact the coefficients, we ran a model without the outliers. We did a logistic regression and excluded the observations that had a goal in the top 95% or bottom 95%. The bottom 5% was 300$, and the top 95% was 65,00$. The results of the regression are in the table 4 of the appendix. The variable goal's

Table 4.8: Logistic Regression with non-NLP features

Model:	Logit	Pseudo R-squared:	0.132
Dependent Variable:	success	AIC:	283561.5799
RMSE:	0.5715	BIC:	284453.6511
No. Observations:	236315	Log-Likelihood:	-1.4169e+05
Df Model:	85	LL-Null:	-1.6333e+05
Df Residuals:	236229	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	7.0000		

Variables	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
intercept	3.5766	0.0392	91.2704	0.0000	3.4998	3.6534
log(goal)	-0.2881	0.0031	-91.7136	0.0000	-0.2942	-0.2819
duration	-0.0191	0.0004	-48.5638	0.0000	-0.0198	-0.0183
nth-project	0.2620	0.0047	55.5578	0.0000	0.2528	0.2712
period	-0.0076	0.0002	-43.0962	0.0000	-0.0080	-0.0073
ratio-nth-city-country	1.4297	0.0649	22.0363	0.0000	1.3025	1.5568
lfd-new-projects	-0.4519	0.0275	-16.4586	0.0000	-0.5057	-0.3981
main-cat-art	-0.4676	0.0196	-23.8848	0.0000	-0.5059	-0.4292
main-cat-comics	0.4780	0.0274	17.4597	0.0000	0.4243	0.5317
main-cat-crafts	-1.2131	0.0322	-37.6439	0.0000	-1.2763	-1.1500
main-cat-dance	0.2070	0.0425	4.8674	0.0000	0.1236	0.2903
main-cat-design	0.3522	0.0213	16.5600	0.0000	0.3105	0.3939
main-cat-fashion	-0.7323	0.0217	-33.7392	0.0000	-0.7749	-0.6898
main-cat-food	-0.4007	0.0220	-18.2149	0.0000	-0.4439	-0.3576
main-cat-games	-0.1548	0.0189	-8.1865	0.0000	-0.1918	-0.1177
main-cat-journalism	-1.3993	0.0437	-32.0547	0.0000	-1.4849	-1.3137
main-cat-music	-0.0288	0.0169	-1.7031	0.0885	-0.0618	0.0043
main-cat-photography	-0.6193	0.0312	-19.8401	0.0000	-0.6805	-0.5581
main-cat-publishing	-0.7280	0.0177	-41.0808	0.0000	-0.7627	-0.6933
main-cat-technology	-0.8907	0.0208	-42.8646	0.0000	-0.9314	-0.8500
main-cat-theater	0.8532	0.0327	26.0918	0.0000	0.7891	0.9172

Note: This regression is very similar the second stage regression in the causality section, except we use the actual duration as independent variable. The objective here is to set a performance baseline for further models and analyse the impact of the independent variables on the success of campaigns. The categories of projects are shown to demonstrate the impact on success. Months and location are still control variables, as for all the models.

coefficient is almost the same in the regression with and without outliers, like all the other coefficients. We conclude that outliers do not bias the estimation of the coefficient, and

we can use those observations in our models.

Table 4.9: Average Maginal Effect Calculated at the Median of Each Regressor

Variable	median	std	dy/dx	Pr(>|z|)	std_margeff
log(goal)	8.517	1.691	-0.071	0.000	-0.120
duration	30.000	12.237	-0.005	0.000	-0.058
nth-project	1.000	2.582	0.065	0.000	0.167
period	75.000	27.801	-0.002	0.000	-0.052
ratio-nth-city-country	0.005	0.092	0.353	0.000	0.032
lfd-new-projects	0.028	0.229	-0.111	0.000	-0.026

Note: The column median is the median value of the variable. The column dy/dx indicates the marginal effect of a one-unit shock on the median observation. The std-margeff column indicates a shock of one standard deviation on the median observation.

Except for the category variable, the strongest effect is the number of projects per creator, which is of almost 7% for an additional project. The duration has an impact of - 0.5 %, which can be economically significant since it would make a - 1.5 % between a project of 30 and 60 days. The category seems to be one of the strongest predictors, with theatre being one of the most successful categories and journalism, one of the least. The location also has a significant impact on the probability of success, with marginal effect ranging from -0.23 to 0.40.

4.2.2 Logistic model with policy change interaction variable

In continuity with the second stage regression, we wanted to include an interaction variable between the policy change variable and the duration variable. With this interaction variable, we want to verify if the policy change affected the impact of duration on the success of project.

As shown in Table 4.9, the interaction variable between policy change and duration is significantly different from zero and is negative. That implies that longer campaigns are even worst after the policy change. We computed the average marginal effect of both variables. The average marginal effect duration is -0.003, and the average marginal effect

Table 4.10: Policy Change and Duration Interaction Variable: Logistic Regression

Model:	Logit	Pseudo R-squared:	0.133
Dependent Variable:	success	AIC:	283347.2762
RMSE:	0.5705	BIC:	284249.7203
No. Observations:	236315	Log-Likelihood:	-1.4159e+05
Df Model:	86	LL-Null:	-1.6333e+05
Df Residuals:	236228	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	7.0000		

Variable	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
intercept	3.5324	0.0393	89.8854	0.0000	3.4554	3.6094
log(goal)	-0.2844	0.0031	-90.3037	0.0000	-0.2906	-0.2782
duration	-0.0136	0.0005	-25.1033	0.0000	-0.0147	-0.0125
policy-change*duration	-0.0075	0.0005	-14.6931	0.0000	-0.0085	-0.0065
nth-roject	0.2591	0.0047	54.9596	0.0000	0.2499	0.2684
period	-0.0066	0.0002	-34.3979	0.0000	-0.0070	-0.0062
ratio-nth-city-country	1.3846	0.0650	21.3110	0.0000	1.2573	1.5120
lfd-new-projects	-0.4847	0.0275	-17.5924	0.0000	-0.5387	-0.4307

Note: This regression has an interaction variable between the policy change and the duration of the project. The objective is to capture if the policy change affected the impact of duration on success.

of the interaction variable is -0.002. Both effects are of similar magnitudes, that result confirms that longer campaigns got even worst. Maybe it is because people were more aware of the negative correlation between duration and success. That means that only unprepared creators that did not even make a simple Google research made campaigns longer than 30 days.

4.2.3 Logistic model with NLP variables

To evaluate if econometric models can use NLP variables, we added a few variables extracted from the blurb. The word count is negative and significantly different from 0, but all the other NLP coefficients are positive. The distribution of the NLP variables can help us understand if the effect is economically significant. Like in the previous table, we compare the average marginal effect of the independent variables in the success to understand

which variable has more impact.

Table 4.11: With NLP Variables: Logistic Regression

Model:	Logit	Pseudo R-squared:	0.135
Dependent Variable:	success	AIC:	353433.0443
RMSE:	0.5690	BIC:	354407.8824
No. Observations:	295394	Log-Likelihood:	-1.7662e+05
Df Model:	91	LL-Null:	-2.0416e+05
Df Residuals:	295302	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	7.0000		

Variable	Coef.	Std.Err.	z	P> |z|	[0.025	0.975]
word-count	-0.0143	0.0013	-11.1907	0.0000	-0.0167	-0.0118
frac-six-letters	0.3574	0.0540	6.6181	0.0000	0.2515	0.4632
ari	0.0328	0.0025	13.0888	0.0000	0.0279	0.0377
fog	0.0124	0.0019	6.3532	0.0000	0.0086	0.0162
flesh-score	0.0029	0.0004	6.7241	0.0000	0.0021	0.0038
sentiment	0.1029	0.0167	6.1646	0.0000	0.0702	0.1357
intercept	3.1281	0.0627	49.8681	0.0000	3.0051	3.2510
log(goa)	-0.2948	0.0028	-104.0288	0.0000	-0.3004	-0.2893
duration	-0.0191	0.0004	-54.2320	0.0000	-0.0198	-0.0184
nth-project	0.2545	0.0042	60.9565	0.0000	0.2463	0.2627
period	-0.0079	0.0002	-49.2817	0.0000	-0.0082	-0.0076
ratio-nth-city-country	1.4093	0.0581	24.2731	0.0000	1.2955	1.5231
lfd-new-projects	-0.4461	0.0246	-18.1604	0.0000	-0.4943	-0.3980

Note: All the variables from the precedent logistic regressions are included here with the addition of NLP related variables. We include those to understand their effect on success.

The coefficients of this regression are consistent with the other models estimated previously. The negative correlation indicates that shorter blurbs are correlated with higher success. The frac-six-letter variable is the fraction of words in the blurb that have six letters or more. The fog, ari and flesh score are indicators of the simplicity of the text. As indicated in the literature review, they depend on the average letters per word, syllables per word, word per sentence and complex word per sentence. A higher value of those indicators means a more sophisticated language.

Table 4.12: Average Maginal Effect of NLP variables on the Median Regressor

Variables	median	std	dy/dx	Pr(>|z|)	std_margeff
word-count	20.000	5.186	-0.004	0.000	-0.019
frac-six-letters	0.364	0.135	0.084	0.000	0.011
ari	11.400	4.002	0.008	0.000	0.032
fog	11.810	3.671	0.003	0.000	0.012
flesh-score	60.310	21.623	0.001	0.000	0.015
sentiment	0.100	0.248	0.027	0.000	0.007

Note: The column median is the median value of each variable. The column dy/dx indicates the marginal effect of a one-unit shock on the median observation. The std-margeff column indicates a shock of one standard deviation on the median observation. An increase in fog, ari and flesh score implies the usage of more complex words. An increase in sentiment indicates a more positive text. Frac-six-letter is the fraction of words with six letters or more.

To evaluate if the marginal effects of the NLP variable are economically significant, we computed the marginal effect of one standard deviation. Table 4.12 shows the median, standard deviation of said variables. It also shows the impact of one standard deviation shock of the variables on the probability of success of the projects. The NLP variables have a weak effect, with the stronger one of approximately 2.5% for one standard deviation. The word count and flesh score also have an impact of more than one percent in absolute value for a one standard deviation shock. The marginal effect of the NLP variables informs us about the underlying mechanism that increases the success of projects. Smaller blurbs with more technical and longer words, combined with a positive tone, increases the chances of success. The addition of NLP variables causes little increase in performance, with less than one percent of difference with the NLP variables.

As we can see in table 4.13, the performance metrics of the logistic regressions are very similar to those from the 2SLS model. It is important to notice that we trained the logistic regression on a train set that represented 80% of the sample, and the test set is the 20% remaining, as we usually do in machine learning models. However, since we want to compare the approaches of econometric versus machine learning, we also computed the performance metrics as we would do in a usual econometric project. We trained the models on the whole dataset and measured the performance metrics on the same dataset

Table 4.13: Performance Metrics of the Logit Regressions

Model	Accuracy	AUC	RMSE
Logistic Base	0.673	0.738	0.572
Logistic Policy Change	0.674	0.739	0.571
Logistic NLP	0.676	0.740	0.570

Note: The first model is from table 3.7, the second from table 3.9 and the third from table 3.10. Those results are from the logistic regressions are train with a train set that represents 80% of the sample, and the performance metrics are evaluated on the remaining observations.

for all the logistic regression. So we tried both approaches with the same three logistic regression to see if the performance would differ significantly. We find the econometric approach yields a very small increase of less than 0.01 for all the metrics and models. This result is consistent with what we expected since logistic regressions have a very low variance. This increase is due to both the fact the models have more data to train on, and also they predict only on data on which they were trained.

4.3 Machine learning model performance

The next section will compare the performance of ML models to predict the success of campaigns. We first assess the performance of the ML models with three different feature sets. Each subsequent dataset has more features than the preceding one. This exercise will answer three questions. First, we want to know which is the most performing model for each dataset. Second, we want to know if the ranking is consistent across all feature sets. Third, this process will tell us what the marginal impact of the additional features on the performance of the models is. Once we assessed the performance of the models for each dataset, we want to compare the overall performance of the best model to the econometric models. The test set is 20% of the total sample. We always use the same test and training set for all models.

4.3.1 Base feature set

The base feature set does not have any variables based on NLP. All the features depend on the characteristics of the project in terms of timing, categories, goal, etc., as explained in the methodology section.

Table 4.14: Performance Metrics: ML Models Without NLP Features

Model	Accuracy	AUC	RMSE
logistic reg LASSO	0.734	0.812	0.516
logistic reg Ridge	0.734	0.814	0.516
tree crit gini	0.675	0.674	0.570
tree crit entropy	0.673	0.672	0.572
random forest gini	0.736	0.817	0.513
extra trees entropy	0.738	0.819	0.511
LGBM	**0.765**	**0.852**	**0.485**

Note: Each row is a different model with the same dataset. No features in that dataset are from NLP.

The only machine learning models with lower performance in for few metrics are decision trees. In this prediction task, they have a similar accuracy but a lower AUC of 0.7, compared to the econometric models. This result makes it the least performing models for the area under the curve.

The regularized logistic regression outperforms the econometric models—with an increase in accuracy of 0.067 and an increase of AUC of 0.07. The random forest and extra tree models have similar performance than the regularized logistic regression. The runner-up is the extra-tree model. Finally, GBM is the best model for every metric. The differences for the accuracy and AUC are respectively of 0.027 and 0.033 difference with the second-best model, the extra tree. The overall difference with the econometric models is important; the performance increases of approximately 14% in accuracy and of 16% in AUC. The RMSE decrease of 0.03 of the LGBM compared to the econometric models is also noticeable and consistent with the accuracy increase.

4.3.2 With NLP indicators

The next models were run with additional features containing the NLP variables such as the number of words, average density, etc. and the scores used in the logistic model. We also use other count variables for word type such as adverb, verb, noun, etc. as features.

Table 4.15: Performance Metrics: ML Models with Simple NLP Features

Model	Accuracy	AUC	RMSE
logistic reg LASSO	0.737	0.816	0.513
logistic reg Ridge	0.738	0.817	0.512
tree crit gini	0.670	0.669	0.574
tree crit entropy	0.665	0.664	0.579
random forest gini	0.741	0.821	0.508
extra trees entropy	0.750	0.832	0.500
LGBM	**0.768**	**0.854**	**0.482**

Note: Each row is a different model with the same dataset. This dataset has the same features from the previous models in section 4.2.1, plus NLP features like the readability indexes, the word counts, etc.

The ranking of the performance of the model is very similar, and most models have a modest performance increase of less than 0.01 in AUC and accuracy. The extra tree model, however, has the biggest increase of performance of 0.012 in accuracy, 0.013 in AUC and a decrease of 0.011 in RMSE. Overall the GBM is still the best model, but its performance increases due to the additional features are less important. Finally, decision tree classifiers are the least adapted to the NLP features since their accuracy decreases.

4.3.3 Topic features and bag of words

The final dataset has 400 features with the bag of words representation of the text. We also add 82 topic features, which represent the probability that each blurb concern one of the 82 topics the latent Dirichlet's allocation model created. We used the TF-IDF representation of the text to train the topic features.

The overall performance increases of the models follow a similar narrative as the previous addition of NLP features. They all have a modest increase of less than 0.01, but

Table 4.16: Performance Metrics: ML Models with Every NLP Features

Model	Accuracy	AUC	RMSE
logistic reg LASSO	0.743	0.822	0.507
logistic reg Ridge	0.744	0.824	0.506
tree crit gini	0.669	0.667	0.576
tree crit entropy	0.666	0.665	0.578
random forest gini	0.743	0.822	0.507
extra trees entropy	0.756	0.838	0.494
LGBM	**0.771**	**0.858**	**0.479**

Note: Each row is a different model with the same dataset. This dataset has the same features from the previous models in section 4.2.2 plus more complex NLP features like the bag of words representation and the topic modelling trained on the TF-IDF representation of the text.

the extra tree model has one of the biggest performance increase with the NLP features. If this model did not beat the GBM with the basic feature set for accuracy, it had the biggest increase with the addition of the NLP features. Its accuracy and AUC both increased by 2,4%, and its RMSE decreased by 3.4% compared to the without NLP features.

To assess if NLP variables can help us in this setting, we take a closer look at the best performing model and analyze the performance increase between the dataset without NLP features and the final dataset. The overall performance increase AUC with the GBM is 0.7%, which smaller than the 2,65% increase observed by Netzer & Al Netzer et al. (2019) in a similar context.

4.3.4 Comparison of the most performing ML Models with the Econometric Models

The previous section showed the increase of between each model with additional features. As shown previously, the overall prediction power increase is important. However, the pertinence of NLP uncertain because it only showed a small increase in performance.

The total improvement compared with the logistic regression is of approximately 14% for the accuracy, 16% for the AUC and RMSE for the LGBM model. Traditional econometric models are mostly limited to linear relationships, except for the features on which

Table 4.17: Percentage increase of the GBM compared to the econometric models

Model	Accuracy	AUC	RMSE
2SOLS	14.9%	16.7%	-16.5%
Logit Base	14.5%	16.3%	-16.2%
Logit NLP	14.3%	16.2%	-16.1%

Note: Each of those cells is the GBM relative increase for a given performance metric (columns) compared to a given model (rows). Per example, the 14.9% at the first row and first column means the GBM has an accuracy 14.9% better than the 2SLS model.

we explicitly apply quadratic or cubic transformations or the interaction variable we defined. The ability of machine learning models to "learn" by themselves non-linear relationships and handle a very large amount of data allows them to more flexibly approximate and account for the performance increase we observe in this dataset.

Conclusion

This thesis combines an econometrical analysis and machine learning predictions to understand crowdfunding, an alternative to the traditional ways of financing new business ventures. First of all, we uncover the existence of serial creators, people using the website receptively. These creators differ from the other website users by making generally shorter campaigns with smaller goals. They make projects concentrated in certain categories like publishing and games, probably because these categories easily permit iterative projects.

Furthermore, we demonstrate a causal relationship between the duration of campaigns and the probability of success. With the usage of a two-stage least square regression and a policy change as an instrumental variable, we find that an additional day of campaign length has a causal negative impact of approximately -2 % on the probability of success. Moreover, we estimate the relative importance of the dependant variable with logistic regression and notice that the main success drivers are the experience of the creator and the category of project. We estimate that the average marginal effect of an additional project increases the probability of success by approximately 6%. The marginal effects of the NLP variables in the logistic regression indicate that Blurbs with fewer words, but a more sophisticated vocabulary have more chances of success. Finally, we benchmark the prediction power of the models mentioned above with multiple machine learning models and the usage of NLP features. The performance increase varies depending on the metric observed. The best model is the gradient boosting model that exhibits a 14% increase in accuracy and a 16% increase in AUC compared to the logistic regression. The overall increase in performance brought by machine learning is substantial and illustrates the

power of those models to predict economic phenomena.

Finally, it would be interesting to compare Kickstarter to another crowdfunding platform such as Indiegogo. Serial creators could be alternating between both platforms; it would be interesting to identify if it is the case. Moreover, a popular phenomenon in the online retail market is the so-called "drop shipping". Entrepreneurs make targeted advertising campaigns for products they buy in bulk on an online retail platform like Alibaba. Finally, they ship them directly to the consumer. It would be interesting to see if some of the serial creators are drop shippers who use crowdsourcing websites as a selling platform. It is possible some creators make a new account between each project to stay undercover. Another interesting approach would be to analyze the pictures and videos describing the campaigns to extract other quality signals from the creators and quantify the impact of those signals on the success of projects.

Bibliography

Acemoglu, D. and Angrist, J. (2000). How Large Are Human-Capital Externalities? Evidence from Compulsory Schooling Laws. *NBER Macroeconomics Annual*, 15:9–59. Publisher: The University of Chicago Press.

Angrist, J. D. and Imbens, G. W. (1995). Two-Stage Least Squares Estimation of Average Causal Effects in Models with Variable Treatment Intensity. *Journal of the American Statistical Association*, 90(430):431–442. Publisher: Taylor & Francis.

Belleflamme, P., Lambert, T., and Schwienbacher, A. (2014). Crowdfunding: Tapping the right crowd. *Journal of Business Venturing*, 29(5):585–609.

Belleflamme, P., Omrani, N., and Peitz, M. (2015). The economics of crowdfunding platforms. *Information Economics and Policy*, 33:11–28.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(January):993–1022.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Chan, C. S. R., Park, H. D., Patel, P., and Gomulya, D. (2018). Reward-based crowdfunding success: decomposition of the project, product category, entrepreneur, and location effects. *Venture Capital*, 20(3):285–307.

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

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4:1–58.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67.

Kahovi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artifcial Intelligence*, 14(2):1137–1145.

Kickstarter (2011). Shortening the Maximum Project Length. Library Catalog: www.kickstarter.com.

Kickstarter (2019a). Fees for Canada — Kickstarter. Library Catalog: www.kickstarter.com.

Kickstarter (2019b). Our Rules — Kickstarter. Library Catalog: www.kickstarter.com.

Kincaid, J., Fishburne, R., Rogers, R., and Chissom, B. (1975). Derivation Of New Readability Formulas (Automated Readability Index, Fog Count And Flesch Reading Ease Formula) For Navy Enlisted Personnel. Research Branch Report, Institute for Simulation and Training, University of Central Florida, Florida.

Kleemann, F., Voß, G. G., and Rieder, K. (2008). Un(der)paid Innovators: The Commercial Utilization of Consumer Work through Crowdsourcing. *Science, Technology & Innovation Studies*, Vol. 4(No. 1):22.

Lewis, D. (1998). *Naive (Bayes) at forty: The independence assumption in information retrieval | SpringerLink*, volume 1398. ECML: European Conference on Machine Learning, claire nédellec, céline rouveirol edition.

Netzer, O., Lemaire, A., and Herzenstein, M. (2019). When Words Sweat: Identifying Signals for Loan Default in the Text of Loan Applications. *Journal of Marketing Research*, 56(6):960–980.

Oreopoulos, P. (2006). Estimating Average and Local Average Treatment Effects of Education when Compulsory Schooling Laws Really Matter. *American Economic Review*, 96(1):152–175.

Raileanu, L. E. and Stoffel, K. (2004). Theoretical Comparison between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93. Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 1 Publisher: Kluwer Academic Publishers.

Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*.

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society*, 58(1):267–288.

Uysal, A. K. and Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112.

Zhang, W., Yoshida, T., and Tang, X. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765.

Appendix A – Results of the quadradic duration

Regression

Standard error estimation problems related to multiple projects with the same creators (section 4.1.1 in the result section)

In the regression shown in Table 1 we used the outliers of the goal variable. We will show in section 4.2.1 that outliers do not affect significantly the results of our models.

Table 1: Logistic Regression Without Serial Creators

Dep. Variable:	Success	**No. Observations:**	193265
Model:	Logit	**Df Residuals:**	193180
Method:	MLE	**Df Model:**	84
Date:	Sat, 11 Jul 2020	**Pseudo R-squ.:**	0.1234
Time:	16:29:42	**Log-Likelihood:**	-1.1556e+05
converged:	True	**LL-Null:**	-1.3183e+05

	coef	**std err**	**z**	**P$>$ |z|**	**[0.025**	**0.975]**
intercept	4.1799	0.043	96.707	0.000	4.095	4.265
ln_goal	-0.3086	0.004	-87.091	0.000	-0.316	-0.302
duration	-0.0180	0.000	-41.751	0.000	-0.019	-0.017
period	-0.0100	0.000	-50.689	0.000	-0.010	-0.010

ratio_nth_city_country	1.5903	0.071	22.426	0.000	1.451	1.729
lfd_new_projects	-0.5094	0.030	-17.169	0.000	-0.568	-0.451
main_cat_art	-0.5210	0.021	-24.237	0.000	-0.563	-0.479
main_cat_comics	0.2856	0.033	8.669	0.000	0.221	0.350
main_cat_crafts	-1.3333	0.036	-36.650	0.000	-1.405	-1.262
main_cat_dance	0.1279	0.047	2.747	0.006	0.037	0.219
main_cat_design	0.3208	0.024	13.580	0.000	0.275	0.367
main_cat_fashion	-0.7615	0.024	-32.004	0.000	-0.808	-0.715
main_cat_food	-0.3450	0.023	-14.810	0.000	-0.391	-0.299
main_cat_games	-0.3902	0.022	-17.858	0.000	-0.433	-0.347
main_cat_journalism	-1.4804	0.048	-31.134	0.000	-1.574	-1.387
main_cat_music	-0.0772	0.018	-4.268	0.000	-0.113	-0.042
main_cat_photography	-0.6719	0.034	-19.796	0.000	-0.738	-0.605
main_cat_publishing	-0.8038	0.019	-41.690	0.000	-0.842	-0.766
main_cat_technology	-0.9877	0.023	-42.999	0.000	-1.033	-0.943
main_cat_theater	0.8413	0.035	23.964	0.000	0.772	0.910
location_AU	-0.7188	0.036	-19.903	0.000	-0.790	-0.648
location_CA	-0.4992	0.027	-18.576	0.000	-0.552	-0.447
location_GB	-0.5630	0.022	-25.831	0.000	-0.606	-0.520
location_IE	-1.0947	0.097	-11.272	0.000	-1.285	-0.904
location_US_AK	-0.3076	0.109	-2.812	0.005	-0.522	-0.093
location_US_AL	-0.9291	0.075	-12.454	0.000	-1.075	-0.783
location_US_AR	-0.9006	0.095	-9.456	0.000	-1.087	-0.714
location_US_AZ	-0.6331	0.043	-14.640	0.000	-0.718	-0.548
location_US_CO	-0.2565	0.038	-6.832	0.000	-0.330	-0.183
location_US_CT	-0.2953	0.060	-4.924	0.000	-0.413	-0.178
location_US_DC	-0.0304	0.055	-0.554	0.580	-0.138	0.077
location_US_DE	-0.3281	0.121	-2.709	0.007	-0.566	-0.091

location_US_FL	-0.9045	0.029	-30.832	0.000	-0.962	-0.847
location_US_GA	-0.7510	0.036	-20.656	0.000	-0.822	-0.680
location_US_HI	-0.2395	0.081	-2.963	0.003	-0.398	-0.081
location_US_IA	-0.5761	0.081	-7.148	0.000	-0.734	-0.418
location_US_ID	-0.4939	0.078	-6.299	0.000	-0.648	-0.340
location_US_IL	-0.1451	0.030	-4.764	0.000	-0.205	-0.085
location_US_IN	-0.6824	0.052	-13.078	0.000	-0.785	-0.580
location_US_KS	-0.7407	0.082	-9.022	0.000	-0.902	-0.580
location_US_KY	-0.6611	0.068	-9.771	0.000	-0.794	-0.529
location_US_LA	-0.3183	0.059	-5.369	0.000	-0.435	-0.202
location_US_MA	0.2038	0.035	5.847	0.000	0.136	0.272
location_US_MD	-0.4221	0.049	-8.653	0.000	-0.518	-0.326
location_US_ME	0.0508	0.077	0.660	0.509	-0.100	0.202
location_US_MI	-0.5130	0.039	-13.093	0.000	-0.590	-0.436
location_US_MN	0.0007	0.043	0.016	0.988	-0.083	0.084
location_US_MO	-0.5403	0.046	-11.668	0.000	-0.631	-0.450
location_US_MS	-1.0565	0.116	-9.109	0.000	-1.284	-0.829
location_US_MT	-0.0837	0.091	-0.915	0.360	-0.263	0.095
location_US_NC	-0.5199	0.039	-13.396	0.000	-0.596	-0.444
location_US_ND	-0.4774	0.161	-2.973	0.003	-0.792	-0.163
location_US_NE	-0.6410	0.101	-6.367	0.000	-0.838	-0.444
location_US_NH	-0.1939	0.090	-2.151	0.031	-0.371	-0.017
location_US_NJ	-0.4686	0.045	-10.403	0.000	-0.557	-0.380
location_US_NM	-0.3153	0.074	-4.238	0.000	-0.461	-0.169
location_US_NV	-0.6829	0.057	-12.030	0.000	-0.794	-0.572
location_US_NY	0.2397	0.020	11.716	0.000	0.200	0.280
location_US_OH	-0.5857	0.037	-15.815	0.000	-0.658	-0.513
location_US_OK	-0.7563	0.072	-10.440	0.000	-0.898	-0.614

location_US_OR	0.1036	0.037	2.824	0.005	0.032	0.175
location_US_PA	-0.2662	0.033	-7.994	0.000	-0.331	-0.201
location_US_RI	0.0558	0.086	0.648	0.517	-0.113	0.225
location_US_SC	-0.6630	0.064	-10.289	0.000	-0.789	-0.537
location_US_SD	-0.7264	0.147	-4.940	0.000	-1.015	-0.438
location_US_TN	-0.3034	0.039	-7.707	0.000	-0.381	-0.226
location_US_TX	-0.5331	0.026	-20.219	0.000	-0.585	-0.481
location_US_UT	-0.2195	0.044	-5.031	0.000	-0.305	-0.134
location_US_VA	-0.4869	0.043	-11.378	0.000	-0.571	-0.403
location_US_VT	0.5339	0.087	6.126	0.000	0.363	0.705
location_US_WA	0.0176	0.032	0.545	0.586	-0.046	0.081
location_US_WI	-0.4023	0.052	-7.810	0.000	-0.503	-0.301
location_US_WV	-0.5090	0.117	-4.360	0.000	-0.738	-0.280
location_US_WY	-0.1546	0.152	-1.017	0.309	-0.453	0.143
month_creation_1	0.2269	0.025	9.065	0.000	0.178	0.276
month_creation_10	0.1744	0.024	7.359	0.000	0.128	0.221
month_creation_11	0.1026	0.025	4.061	0.000	0.053	0.152
month_creation_12	-0.2034	0.031	-6.629	0.000	-0.264	-0.143
month_creation_2	0.1989	0.024	8.206	0.000	0.151	0.246
month_creation_3	0.3068	0.023	13.208	0.000	0.261	0.352
month_creation_4	0.1625	0.024	6.704	0.000	0.115	0.210
month_creation_5	0.1394	0.024	5.876	0.000	0.093	0.186
month_creation_6	0.1403	0.025	5.714	0.000	0.092	0.188
month_creation_8	-0.1591	0.025	-6.254	0.000	-0.209	-0.109
month_creation_9	0.1181	0.025	4.711	0.000	0.069	0.167

First stage Regression (table 4.3 in the result section)

Table 2: First stage regression

Model:	OLS	Adj. R-squared:	0.095
Dependent Variable:	duration	AIC:	2288447.4284
Date:	2020-07-13 22:00	BIC:	2289369.2861
No. Observations:	295394	Log-Likelihood:	-1.1441e+06
Df Model:	86	F-statistic:	362.1
Df Residuals:	295307	Prob (F-statistic):	0.00
R-squared:	0.095	Scale:	135.49

Variable	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
intercept	32.8875	0.1984	165.7678	0.0000	32.4987	33.2764
ln_goal	1.5086	0.0136	110.6595	0.0000	1.4819	1.5353
choc	-6.7036	0.1444	-46.4260	0.0000	-6.9866	-6.4206
nth_project	-0.4242	0.0086	-49.3980	0.0000	-0.4411	-0.4074
period	-0.1851	0.0051	-36.5680	0.0000	-0.1951	-0.1752
period_2	0.0012	0.0000	37.9878	0.0000	0.0012	0.0013
ratio_nth_city_country	-0.9673	0.2826	-3.4227	0.0006	-1.5213	-0.4134
lfd_new_projects	1.5806	0.1277	12.3760	0.0000	1.3303	1.8310
month_creation_1	0.5000	0.1063	4.7042	0.0000	0.2917	0.7084
month_creation_10	0.1469	0.1008	1.4575	0.1450	-0.0506	0.3444
month_creation_11	0.4354	0.1079	4.0363	0.0001	0.2240	0.6469
month_creation_12	2.4341	0.1313	18.5357	0.0000	2.1767	2.6915
month_creation_2	0.6588	0.1035	6.3644	0.0000	0.4559	0.8617
month_creation_3	0.1800	0.0991	1.8176	0.0691	-0.0141	0.3742
month_creation_4	0.2917	0.1039	2.8080	0.0050	0.0881	0.4954
month_creation_5	-0.0962	0.1009	-0.9535	0.3403	-0.2938	0.1015
month_creation_6	-0.0792	0.1047	-0.7565	0.4493	-0.2844	0.1260
month_creation_8	-0.2828	0.1073	-2.6348	0.0084	-0.4931	-0.0724

month_creation_9	0.5027	0.1068	4.7096	0.0000	0.2935	0.7120
main_cat_art	-0.2827	0.0952	-2.9707	0.0030	-0.4692	-0.0962
main_cat_comics	0.4430	0.1258	3.5213	0.0004	0.1964	0.6896
main_cat_crafts	0.1385	0.1466	0.9448	0.3448	-0.1488	0.4258
main_cat_dance	-0.2565	0.2015	-1.2726	0.2032	-0.6514	0.1385
main_cat_design	0.5815	0.1029	5.6537	0.0000	0.3799	0.7831
main_cat_fashion	-0.1830	0.1038	-1.7639	0.0777	-0.3864	0.0203
main_cat_food	0.2926	0.1055	2.7742	0.0055	0.0859	0.4994
main_cat_games	-1.1928	0.0913	-13.0655	0.0000	-1.3718	-1.0139
main_cat_journalism	1.2853	0.1848	6.9546	0.0000	0.9231	1.6475
main_cat_music	1.5845	0.0830	19.1007	0.0000	1.4219	1.7471
main_cat_photography	0.3350	0.1503	2.2295	0.0258	0.0405	0.6296
main_cat_publishing	0.9159	0.0856	10.6943	0.0000	0.7480	1.0838
main_cat_technology	0.9341	0.0942	9.9184	0.0000	0.7495	1.1187
main_cat_theater	-0.7337	0.1407	-5.2132	0.0000	-1.0096	-0.4579
location_AU	0.5460	0.1516	3.6014	0.0003	0.2488	0.8431
location_CA	0.9920	0.1150	8.6234	0.0000	0.7666	1.2175
location_GB	0.1897	0.0919	2.0639	0.0390	0.0096	0.3698
location_IE	0.8662	0.3980	2.1763	0.0295	0.0861	1.6462
location_US_AK	-0.5370	0.4751	-1.1303	0.2584	-1.4682	0.3942
location_US_AL	0.4473	0.2992	1.4947	0.1350	-0.1392	1.0338
location_US_AR	0.6725	0.3821	1.7600	0.0784	-0.0764	1.4214
location_US_AZ	0.5439	0.1768	3.0771	0.0021	0.1975	0.8903
location_US_CO	-0.3562	0.1642	-2.1696	0.0300	-0.6780	-0.0344
location_US_CT	0.8354	0.2607	3.2042	0.0014	0.3244	1.3464
location_US_DC	0.2988	0.2395	1.2479	0.2121	-0.1705	0.7682
location_US_DE	1.2730	0.4896	2.6002	0.0093	0.3134	2.2325
location_US_FL	1.1065	0.1184	9.3473	0.0000	0.8745	1.3385

location_US_GA	0.7596	0.1529	4.9693	0.0000	0.4600	1.0591
location_US_HI	0.6951	0.3559	1.9528	0.0508	-0.0025	1.3927
location_US_IA	0.5056	0.3355	1.5071	0.1318	-0.1519	1.1630
location_US_ID	0.5518	0.3246	1.7000	0.0891	-0.0844	1.1879
location_US_IL	0.6549	0.1323	4.9494	0.0000	0.3956	0.9143
location_US_IN	0.7378	0.2185	3.3771	0.0007	0.3096	1.1660
location_US_KS	0.3481	0.3411	1.0204	0.3075	-0.3205	1.0167
location_US_KY	1.4100	0.2809	5.0197	0.0000	0.8595	1.9606
location_US_LA	0.6323	0.2584	2.4473	0.0144	0.1259	1.1387
location_US_MA	-0.0324	0.1532	-0.2113	0.8326	-0.3327	0.2679
location_US_MD	1.0553	0.2079	5.0762	0.0000	0.6478	1.4627
location_US_ME	-0.1630	0.3446	-0.4730	0.6362	-0.8384	0.5124
location_US_MI	0.4719	0.1643	2.8720	0.0041	0.1498	0.7939
location_US_MN	-0.0158	0.1868	-0.0843	0.9328	-0.3819	0.3504
location_US_MO	0.6273	0.1990	3.1528	0.0016	0.2373	1.0173
location_US_MS	0.7824	0.4486	1.7440	0.0812	-0.0969	1.6617
location_US_MT	-1.0025	0.4036	-2.4836	0.0130	-1.7936	-0.2113
location_US_NC	0.4186	0.1665	2.5133	0.0120	0.0921	0.7450
location_US_ND	0.9500	0.7081	1.3417	0.1797	-0.4378	2.3378
location_US_NE	1.0394	0.4174	2.4900	0.0128	0.2212	1.8576
location_US_NH	0.3329	0.3721	0.8948	0.3709	-0.3963	1.0621
location_US_NJ	1.1021	0.1903	5.7923	0.0000	0.7292	1.4751
location_US_NM	-0.0283	0.3224	-0.0876	0.9302	-0.6601	0.6036
location_US_NV	-0.2771	0.2218	-1.2493	0.2115	-0.7119	0.1576
location_US_NY	-0.2081	0.0895	-2.3262	0.0200	-0.3834	-0.0328
location_US_OH	0.6308	0.1571	4.0161	0.0001	0.3229	0.9386
location_US_OK	0.1781	0.2968	0.6001	0.5484	-0.4036	0.7599
location_US_OR	-0.8247	0.1585	-5.2031	0.0000	-1.1354	-0.5140

location_US_PA	0.7982	0.1435	5.5637	0.0000	0.5170	1.0793
location_US_RI	-0.0212	0.3931	-0.0539	0.9570	-0.7916	0.7492
location_US_SC	0.2464	0.2693	0.9147	0.3603	-0.2815	0.7742
location_US_SD	-0.7610	0.6283	-1.2112	0.2258	-1.9925	0.4705
location_US_TN	0.0686	0.1756	0.3903	0.6963	-0.2757	0.4128
location_US_TX	0.3063	0.1122	2.7307	0.0063	0.0864	0.5261
location_US_UT	-0.7419	0.1851	-4.0076	0.0001	-1.1047	-0.3790
location_US_VA	0.6713	0.1799	3.7305	0.0002	0.3186	1.0240
location_US_VT	-0.8412	0.3763	-2.2355	0.0254	-1.5788	-0.1037
location_US_WA	-0.5317	0.1396	-3.8099	0.0001	-0.8052	-0.2582
location_US_WI	0.3712	0.2174	1.7072	0.0878	-0.0549	0.7973
location_US_WV	1.9427	0.5041	3.8535	0.0001	0.9546	2.9308
location_US_WY	0.9792	0.6678	1.4662	0.1426	-0.3297	2.2881

Second stage Regression (table 4.6 in the result section)

Table 3: Second stage Regression

Dep. Variable:	success	**R-squared:**	0.0376
Estimator:	IV-2SLS	**Adj. R-squared:**	0.0373
No. Observations:	295394	**F-statistic:**	5.524e+04
Date:	Mon, Jul 13 2020	**P-value (F-stat)**	0.0000
Time:	21:37:27	**Distribution:**	chi2(86)
Cov. Estimator:	robust		

	Parameter	**Std. Err.**	**T-stat**	**P-value**	**Lower CI**	**Upper CI**
intercept	1.9861	0.0314	63.327	0.0000	1.9246	2.0475
ln_goal	-0.0352	0.0015	-22.810	0.0000	-0.0382	-0.0322
nth_project	0.0130	0.0006	21.269	0.0000	0.0118	0.0142
period	-0.0134	0.0004	-37.232	0.0000	-0.0141	-0.0127
period_2	7.876e-05	2.278e-06	34.567	0.0000	7.429e-05	8.322e-05
ratio_nth_city_country	0.2085	0.0379	5.4982	0.0000	0.1342	0.2828
lfd_new_projects	-0.0709	0.0057	-12.396	0.0000	-0.0822	-0.0597
main_cat_art	-0.1056	0.0041	-25.700	0.0000	-0.1136	-0.0975
main_cat_comics	0.1269	0.0051	25.029	0.0000	0.1170	0.1369
main_cat_crafts	-0.2406	0.0060	-40.255	0.0000	-0.2523	-0.2289
main_cat_dance	0.0560	0.0086	6.5372	0.0000	0.0392	0.0727
main_cat_design	0.0947	0.0045	21.196	0.0000	0.0860	0.1035
main_cat_fashion	-0.1545	0.0043	-35.525	0.0000	-0.1630	-0.1459
main_cat_food	-0.0797	0.0044	-18.031	0.0000	-0.0884	-0.0710
main_cat_games	-0.0358	0.0040	-8.8671	0.0000	-0.0438	-0.0279
main_cat_journalism	-0.2581	0.0075	-34.552	0.0000	-0.2727	-0.2434
main_cat_music	0.0123	0.0039	3.1578	0.0016	0.0047	0.0200
main_cat_photography	-0.1171	0.0065	-18.013	0.0000	-0.1299	-0.1044
main_cat_publishing	-0.1429	0.0037	-38.333	0.0000	-0.1502	-0.1356

main_cat_technology	-0.1575	0.0040	-39.205	0.0000	-0.1653	-0.1496
main_cat_theater	0.1641	0.0054	30.563	0.0000	0.1536	0.1747
month_creation_1	0.0609	0.0044	13.793	0.0000	0.0523	0.0696
month_creation_10	0.0440	0.0042	10.514	0.0000	0.0358	0.0521
month_creation_11	0.0306	0.0046	6.6760	0.0000	0.0216	0.0395
month_creation_12	-0.0044	0.0062	-0.7163	0.4738	-0.0165	0.0077
month_creation_2	0.0564	0.0044	12.912	0.0000	0.0478	0.0650
month_creation_3	0.0650	0.0041	15.814	0.0000	0.0569	0.0730
month_creation_4	0.0408	0.0044	9.3112	0.0000	0.0322	0.0494
month_creation_5	0.0317	0.0042	7.5342	0.0000	0.0234	0.0399
month_creation_6	0.0300	0.0044	6.8613	0.0000	0.0215	0.0386
month_creation_8	-0.0375	0.0046	-8.2139	0.0000	-0.0464	-0.0285
month_creation_9	0.0421	0.0045	9.2861	0.0000	0.0332	0.0509
location_AU	-0.1255	0.0078	-16.030	0.0000	-0.1409	-0.1102
location_CA	-0.0770	0.0058	-13.321	0.0000	-0.0883	-0.0656
location_GB	-0.0956	0.0054	-17.667	0.0000	-0.1062	-0.0850
location_IE	-0.1676	0.0205	-8.1950	0.0000	-0.2077	-0.1275
location_US_AK	-0.0385	0.0199	-1.9315	0.0534	-0.0776	0.0006
location_US_AL	-0.1562	0.0121	-12.892	0.0000	-0.1799	-0.1324
location_US_AR	-0.1521	0.0156	-9.7625	0.0000	-0.1826	-0.1215
location_US_AZ	-0.1074	0.0073	-14.675	0.0000	-0.1217	-0.0930
location_US_CO	-0.0562	0.0071	-7.9262	0.0000	-0.0701	-0.0423
location_US_CT	-0.0435	0.0110	-3.9521	0.0001	-0.0650	-0.0219
location_US_DC	-0.0152	0.0104	-1.4630	0.1435	-0.0355	0.0051
location_US_DE	-0.0651	0.0217	-3.0037	0.0027	-0.1075	-0.0226
location_US_FL	-0.1615	0.0051	-31.798	0.0000	-0.1715	-0.1515
location_US_GA	-0.1355	0.0064	-21.258	0.0000	-0.1480	-0.1230
location_US_HI	-0.0565	0.0156	-3.6200	0.0003	-0.0870	-0.0259

location_US_IA	-0.1074	0.0140	-7.6943	0.0000	-0.1348	-0.0801
location_US_ID	-0.0736	0.0132	-5.5630	0.0000	-0.0995	-0.0476
location_US_IL	-0.0209	0.0056	-3.7199	0.0002	-0.0318	-0.0099
location_US_IN	-0.1201	0.0091	-13.236	0.0000	-0.1378	-0.1023
location_US_KS	-0.1442	0.0141	-10.259	0.0000	-0.1718	-0.1167
location_US_KY	-0.1262	0.0121	-10.400	0.0000	-0.1500	-0.1024
location_US_LA	-0.0634	0.0111	-5.7008	0.0000	-0.0852	-0.0416
location_US_MA	0.0575	0.0066	8.7052	0.0000	0.0446	0.0705
location_US_MD	-0.0654	0.0088	-7.4051	0.0000	-0.0827	-0.0481
location_US_ME	-0.0046	0.0147	-0.3137	0.7537	-0.0335	0.0242
location_US_MI	-0.1041	0.0070	-14.942	0.0000	-0.1177	-0.0904
location_US_MN	0.0010	0.0081	0.1246	0.9009	-0.0148	0.0168
location_US_MO	-0.0984	0.0084	-11.718	0.0000	-0.1148	-0.0819
location_US_MS	-0.1851	0.0181	-10.236	0.0000	-0.2205	-0.1497
location_US_MT	-0.0167	0.0170	-0.9823	0.3259	-0.0499	0.0166
location_US_NC	-0.1011	0.0071	-14.245	0.0000	-0.1150	-0.0872
location_US_ND	-0.1016	0.0295	-3.4442	0.0006	-0.1594	-0.0438
location_US_NE	-0.1220	0.0173	-7.0687	0.0000	-0.1558	-0.0882
location_US_NH	-0.0680	0.0155	-4.3863	0.0000	-0.0984	-0.0376
location_US_NJ	-0.0847	0.0082	-10.344	0.0000	-0.1008	-0.0687
location_US_NM	-0.0613	0.0137	-4.4680	0.0000	-0.0882	-0.0344
location_US_NV	-0.1229	0.0092	-13.327	0.0000	-0.1410	-0.1048
location_US_NY	0.0411	0.0038	10.687	0.0000	0.0336	0.0486
location_US_OH	-0.1007	0.0067	-14.947	0.0000	-0.1139	-0.0875
location_US_OK	-0.1600	0.0122	-13.122	0.0000	-0.1840	-0.1361
location_US_OR	0.0123	0.0067	1.8217	0.0685	-0.0009	0.0255
location_US_PA	-0.0399	0.0062	-6.4417	0.0000	-0.0520	-0.0277
location_US_RI	0.0330	0.0161	2.0506	0.0403	0.0015	0.0645

location_US_SC	-0.1263	0.0109	-11.591	0.0000	-0.1477	-0.1050
location_US_SD	-0.1934	0.0251	-7.6972	0.0000	-0.2426	-0.1441
location_US_TN	-0.0610	0.0076	-8.0537	0.0000	-0.0758	-0.0461
location_US_TX	-0.0988	0.0048	-20.756	0.0000	-0.1081	-0.0894
location_US_UT	-0.0370	0.0079	-4.6812	0.0000	-0.0524	-0.0215
location_US_VA	-0.0806	0.0076	-10.594	0.0000	-0.0955	-0.0657
location_US_VT	0.1023	0.0155	6.6038	0.0000	0.0719	0.1327
location_US_WA	0.0018	0.0060	0.2928	0.7697	-0.0100	0.0135
location_US_WI	-0.0653	0.0093	-7.0577	0.0000	-0.0835	-0.0472
location_US_WV	-0.1082	0.0212	-5.0970	0.0000	-0.1497	-0.0666
location_US_WY	-0.0057	0.0288	-0.1985	0.8427	-0.0621	0.0507
duration	-0.0198	0.0010	-20.665	0.0000	-0.0216	-0.0179

Impact of outliers on the coefficient of goal

Table 4: Logit Regression Without the Outliers

Dep. Variable:	success	**No. Observations:**	211851
Model:	Logit	**Df Residuals:**	211765
Method:	MLE	**Df Model:**	85
Date:	Sat, 11 Jul 2020	**Pseudo R-squ.:**	0.1160
Time:	15:19:35	**Log-Likelihood:**	-1.2961e+05
converged:	True	**LL-Null:**	-1.4663e+05

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
intercept	3.4543	0.046	75.410	0.000	3.365	3.544
ln_goal	-0.2612	0.004	-63.682	0.000	-0.269	-0.253
duration	-0.0197	0.000	-47.596	0.000	-0.021	-0.019
nth_project	0.2622	0.005	51.985	0.000	0.252	0.272
period	-0.0079	0.000	-43.099	0.000	-0.008	-0.008
ratio_nth_city_country	1.4958	0.068	21.877	0.000	1.362	1.630
lfd_new_projects	-0.4037	0.029	-13.944	0.000	-0.460	-0.347
main_cat_art	-0.5020	0.021	-24.376	0.000	-0.542	-0.462
main_cat_comics	0.4086	0.028	14.522	0.000	0.353	0.464
main_cat_crafts	-1.3231	0.035	-37.401	0.000	-1.392	-1.254
main_cat_dance	0.2349	0.044	5.352	0.000	0.149	0.321
main_cat_design	0.3398	0.022	15.272	0.000	0.296	0.383
main_cat_fashion	-0.7562	0.023	-33.383	0.000	-0.801	-0.712
main_cat_food	-0.3393	0.023	-14.754	0.000	-0.384	-0.294
main_cat_games	-0.1844	0.020	-9.246	0.000	-0.223	-0.145
main_cat_journalism	-1.4660	0.046	-31.660	0.000	-1.557	-1.375
main_cat_music	-0.0530	0.017	-3.034	0.002	-0.087	-0.019
main_cat_photography	-0.6502	0.032	-20.279	0.000	-0.713	-0.587
main_cat_publishing	-0.7580	0.018	-41.356	0.000	-0.794	-0.722
main_cat_technology	-0.9634	0.022	-43.452	0.000	-1.007	-0.920
main_cat_theater	0.8916	0.034	26.049	0.000	0.825	0.959

location_AU	-0.6734	0.034	-19.537	0.000	-0.741	-0.606
location_CA	-0.4646	0.025	-18.222	0.000	-0.515	-0.415
location_GB	-0.5379	0.021	-26.019	0.000	-0.578	-0.497
location_IE	-0.9641	0.093	-10.360	0.000	-1.147	-0.782
location_US_AK	-0.1476	0.101	-1.468	0.142	-0.345	0.050
location_US_AL	-0.8403	0.070	-12.021	0.000	-0.977	-0.703
location_US_AR	-0.8101	0.090	-9.004	0.000	-0.986	-0.634
location_US_AZ	-0.6059	0.040	-15.134	0.000	-0.684	-0.527
location_US_CO	-0.2014	0.035	-5.706	0.000	-0.271	-0.132
location_US_CT	-0.2925	0.057	-5.120	0.000	-0.404	-0.181
location_US_DC	-0.0263	0.052	-0.507	0.612	-0.128	0.075
location_US_DE	-0.3967	0.110	-3.593	0.000	-0.613	-0.180
location_US_FL	-0.8700	0.027	-31.902	0.000	-0.923	-0.817
location_US_GA	-0.6863	0.034	-20.172	0.000	-0.753	-0.620
location_US_HI	-0.3126	0.076	-4.110	0.000	-0.462	-0.164
location_US_IA	-0.5825	0.075	-7.769	0.000	-0.730	-0.436
location_US_ID	-0.3999	0.072	-5.572	0.000	-0.541	-0.259
location_US_IL	-0.1372	0.029	-4.792	0.000	-0.193	-0.081
location_US_IN	-0.6716	0.048	-13.886	0.000	-0.766	-0.577
location_US_KS	-0.6627	0.076	-8.683	0.000	-0.812	-0.513
location_US_KY	-0.6329	0.062	-10.178	0.000	-0.755	-0.511
location_US_LA	-0.3274	0.056	-5.884	0.000	-0.437	-0.218
location_US_MA	0.2560	0.033	7.726	0.000	0.191	0.321
location_US_MD	-0.3348	0.045	-7.458	0.000	-0.423	-0.247
location_US_ME	-0.0096	0.073	-0.132	0.895	-0.152	0.133
location_US_MI	-0.4583	0.036	-12.675	0.000	-0.529	-0.387
location_US_MN	0.0490	0.040	1.225	0.221	-0.029	0.127
location_US_MO	-0.4828	0.043	-11.132	0.000	-0.568	-0.398
location_US_MS	-1.0113	0.105	-9.677	0.000	-1.216	-0.807
location_US_MT	-0.0144	0.084	-0.172	0.864	-0.179	0.150

location_US_NC	-0.4807	0.037	-13.117	0.000	-0.553	-0.409
location_US_ND	-0.5709	0.159	-3.596	0.000	-0.882	-0.260
location_US_NE	-0.7194	0.093	-7.696	0.000	-0.903	-0.536
location_US_NH	-0.2481	0.081	-3.052	0.002	-0.407	-0.089
location_US_NJ	-0.4554	0.042	-10.825	0.000	-0.538	-0.373
location_US_NM	-0.2931	0.070	-4.167	0.000	-0.431	-0.155
location_US_NV	-0.6154	0.051	-12.120	0.000	-0.715	-0.516
location_US_NY	0.2284	0.019	11.740	0.000	0.190	0.267
location_US_OH	-0.5504	0.035	-15.933	0.000	-0.618	-0.483
location_US_OK	-0.7849	0.068	-11.537	0.000	-0.918	-0.652
location_US_OR	0.1431	0.034	4.186	0.000	0.076	0.210
location_US_PA	-0.2534	0.031	-8.174	0.000	-0.314	-0.193
location_US_RI	0.0857	0.085	1.012	0.312	-0.080	0.252
location_US_SC	-0.6480	0.061	-10.695	0.000	-0.767	-0.529
location_US_SD	-0.7918	0.141	-5.630	0.000	-1.067	-0.516
location_US_TN	-0.3154	0.037	-8.513	0.000	-0.388	-0.243
location_US_TX	-0.4929	0.025	-19.892	0.000	-0.541	-0.444
location_US_UT	-0.1127	0.040	-2.818	0.005	-0.191	-0.034
location_US_VA	-0.4521	0.040	-11.388	0.000	-0.530	-0.374
location_US_VT	0.5152	0.084	6.100	0.000	0.350	0.681
location_US_WA	0.0282	0.030	0.936	0.349	-0.031	0.087
location_US_WI	-0.3492	0.047	-7.398	0.000	-0.442	-0.257
location_US_WV	-0.7062	0.115	-6.152	0.000	-0.931	-0.481
location_US_WY	-0.0226	0.140	-0.162	0.871	-0.296	0.251
month_creation_1	0.1752	0.024	7.341	0.000	0.128	0.222
month_creation_10	0.1641	0.022	7.377	0.000	0.120	0.208
month_creation_11	0.0856	0.024	3.620	0.000	0.039	0.132
month_creation_12	-0.1928	0.029	-6.674	0.000	-0.249	-0.136
month_creation_2	0.2004	0.023	8.789	0.000	0.156	0.245
month_creation_3	0.2652	0.022	12.127	0.000	0.222	0.308

month_creation_4	0.1459	0.023	6.422	0.000	0.101	0.190
month_creation_5	0.1394	0.022	6.299	0.000	0.096	0.183
month_creation_6	0.1141	0.023	4.981	0.000	0.069	0.159
month_creation_8	-0.1471	0.024	-6.218	0.000	-0.194	-0.101
month_creation_9	0.1400	0.023	5.962	0.000	0.094	0.186

Quadratic Duration; Second Stage

To allow for a less restrictive relationship between duration and success, we added the squared value of the predicted duration as another explanatory variable in the regression shown in Table 4.6.

Table 5: Quadratic Duration; Second Stage: OLS Regression

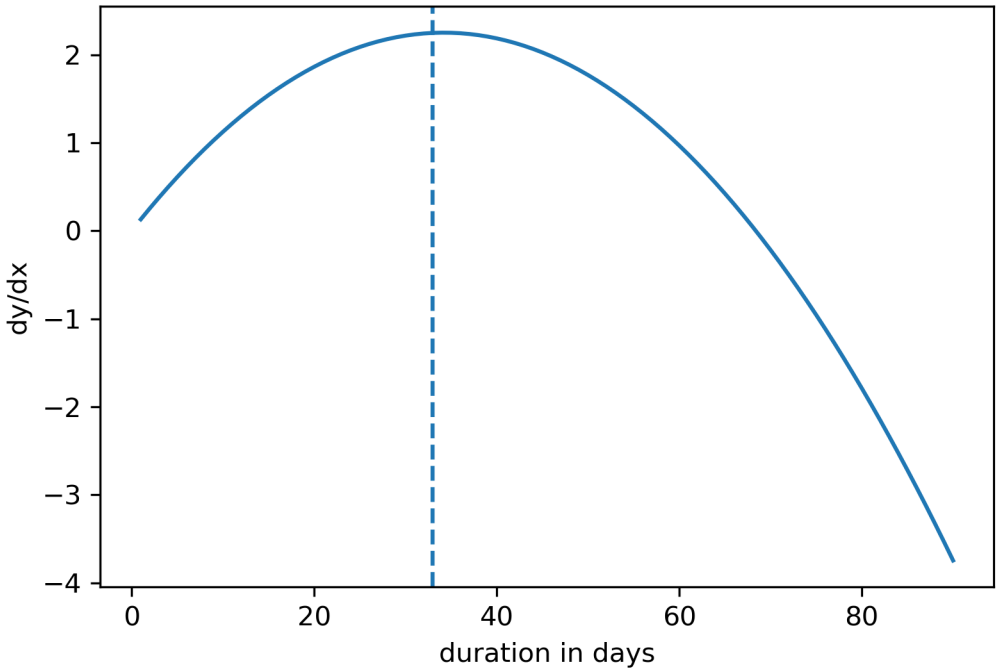
Model:	OLS	Adj. R-squared:	0.162
Dependent Variable:	success	AIC:	375666.9680
Date:	2020-03-24 20:14	BIC:	376599.4217
No. Observations:	295394	Log-Likelihood:	-1.8775e+05
Df Model:	87	F-statistic:	655.0
Df Residuals:	295306	Prob (F-statistic):	0.00
R-squared:	0.162	Scale:	0.20879

Variable	Coef.	Std.Err.	t	P> |t|	[0.025	0.975]
intercept	-0.7379	0.0609	-12.1208	0.0000	-0.8572	-0.6186
log(goal)	-0.0715	0.0016	-46.0108	0.0000	-0.0745	-0.0684
predicted-duration	0.1317	0.0031	42.3154	0.0000	0.1256	0.1378
predicted-duration-square	-0.0019	0.0000	-50.5701	0.0000	-0.0020	-0.0019
nth-project	0.0383	0.0007	54.4909	0.0000	0.0369	0.0396
period	-0.0095	0.0003	-28.2877	0.0000	-0.0101	-0.0088
period-square	0.0001	0.0000	24.5301	0.0000	0.0000	0.0001
ratio-nth-city-country	0.2340	0.0111	21.0328	0.0000	0.2122	0.2558
lfd-new-projects	-0.0901	0.0053	-17.0670	0.0000	-0.1005	-0.0798

Note: This is almost the same regression as table 4.6. The difference is we allow for campaign duration to have a quadratic effect on the success of the campaign. The standard error of the coefficients are not corrected by the number of variables in the second stage.

With coefficients of the duration and square duration, we were able to plot the marginal impact of the campaign length on the probability of success. The results indicated that the maximum of the function is at 33 days, which would be the optimal duration of projects.

Figure 1: Impact of Duration and Duration-square on Success of Projects



Note: The x axis is the value of duration, the y axis is the impact of duration on the success. The blue vertical axis indicates the peak of the quadratic function at 33 days.