HEC MONTRÉAL

A numerical study of the capability of decision trees and random forests to estimate probabilities

par

Marisa Radatz

Jean-François Plante
HEC Montréal
Directeur de recherche

Sciences de la gestion
(Spécialisation Data Science and Business Analytics)

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

May 2022

Résumé

Avec la popularité de l'intelligence artificielle, les algorithmes d'apprentissage automatique sont envisagés pour un nombre croissant de problèmes. En ce qui concerne la classification binaire, la plupart des algorithmes peuvent fournir une estimation de la probabilité qu'un événement se présente, mais il se peut qu'il n'y ait pas de résultats mathématiques pour garantir leur convergence. Après avoir examiné les résultats de convergence pour les arbres de classification et les forêts aléatoires dans la littérature, nous considérons un certain nombre de contextes différents dans lesquels les probabilités estimées sont utilisées et démontrons que certaines pourraient être affectées négativement par des estimations biaisées. Nous exécutons ensuite une simulation Monte Carlo approfondie inspirée de neuf ensembles de données pour évaluer numériquement la capacité de ces algorithmes d'apprentissage automatique à fournir des estimations appropriées des probabilités malgré le manque de résultats théoriques de convergence. Nous constatons que bien que les arbres et les forêts peuvent mieux performer en classification, leur capacité à estimer les probabilités dépasse rarement celle de la régression logistique, même lorsque la régression logistique est mal spécifié.

Mots-clés

Estimations de probabilité, Classification binaire, Arbres de décision, Forêts aléatoires, Régression logistique, Convergence, Simulations de Monte Carlo

Méthodes de recherche

Analyse expérimentale et simulations de Monte Carlo.

Abstract

With the rising popularity of artificial intelligence, machine learning algorithms are being considered for an increasing number of problems. Regarding binary classification, most algorithms can provide an estimate of the probability that an event will occur. However, there may be no mathematical results to guarantee their consistency. After reviewing convergence results for classification trees and random forests in the literature, we consider several different settings in which the estimated probabilities are utilized and demonstrate that some could be negatively impacted by biased estimates. We then run an extensive Monte Carlo simulation inspired by nine data sets to assess numerically the ability of those machine learning algorithms to provide appropriate estimates of the probabilities despite the lack of theoretical consistency results. We find that while trees and forests may perform better at classification, their ability to estimate probabilities rarely exceeds that of logistic regression, even when the logistic regression is misspecified.

Keywords

Probability Estimates, Binary Classification, Decision Trees, Random Forests, Logistic Regression, Consistency, Monte Carlo Simulations

Research methods

Experimental analysis and Monte Carlo simulations.

Contents

List of Tables

List of Figures

List of acronyms

Adj RF Adjusted random forest

Adj Tree Adjusted decision tree

AUC Area under the ROC curve

CART Classification and regression trees algorithm

LG Logistic regression

LGM Logistic regression-based probability model

RF Random forest

RMSE Root mean square error

ROC Receiver operating characteristic

TM Tree-based probability model

Acknowledgements

First and foremost I would like to thank my supervisor Jean-François Plante for guiding me throughout this process. Your continuous support, expertise, and availability made the experience very rewarding and motivating. Not only did I learn about statistical methods in more depth, but also about academia in general, shared through your insights and stories.

I am thankful to my family for giving me the necessary means in life to get to this stage of my academic career. Your unconditional support will always be appreciated.

I gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Natural Sciences and Engineering Research Council of Canada
Conseil de recherches en sciences naturelles et en génie du Canada
Canada

Introduction

Binary classification refers to supervised learning problems where the target variable is binary. Logistic regression is a classical model that will provide the estimated probability of an event based on covariates. When a logistic regression is correctly specified, traditional results on likelihood inference guarantee the consistency of the estimator (e.g. Casella and Berger, 2002).

For practical problems, however, the simplicity of classification trees and their straightforward interpretation often makes them a more attractive choice than logistic regression. The classification and regression trees algorithm (CART) of Breiman et al. (1984), for instance, is widespread and easily available in the *rpart* R package (Therneau et al., 2015) or as *PROC HPSPLIT* in SAS (SAS Institute Inc, 2015). An ensemble of trees may be used to improve the performance of single trees. Based on bagging, random forests (Breiman, 2001) present a robust alternative that typically performs very well for classification and regression problems.

With regard to classification, the target variable could represent an immutable state or the occurrence of an event. Examples of immutable state problems are the authenticity of a banknote, mineral characteristics of a rock, defectiveness of a mechanical part, or disease status. The development and assessment of classification algorithms typically assume that the binary target is fixed. In such situations, each subject has a true state and their true probability of being in a given state can only be zero or one. The fact that algorithms provide different values can be attributed to the uncertainty in the classification. Apart from immutable state target variables, there are numerous situations where

the target represents an event that could occur or not. Consider for instance the default on a loan, recidivism after an offense, customer churn, or propensity to respond to a survey. In these cases, each subject can have a true probability in $(0, 1)$ that represents a level of risk. A correctly specified logistic regression is a consistent estimator for that value. In general, when a logistic regression is well specified, no subject can have a probability as extreme as zero or one.

Today, classification trees and random forests are considered to be a superior alternative to logistic regression in different problems. For instance, Zhao et al. (2016) use random forests for proximity matching using propensity scores in a causal inference problem. Westreich et al. (2010) discuss employing decision trees for estimating propensity scores. Moreover, uplift modeling is a specific case of causal inference where the goal is to evaluate the effect of a marketing action. As mentioned in Sołtys et al. (2015), although a two-model approach may not be optimal in many applications, it can still be competitive in certain scenarios. This approach makes use of the probability estimates directly, predicted by two separate models. In a different field of application, Gelein et al. (2018) use machine learning methods to derive propensity scores to adjust for nonresponse in surveys.

Classification trees and random forests have been extensively studied. Their ability to rank subjects is partially supported by theoretical results and observed in practice. However, regarding the estimation of probabilities, even Breiman et al. (1984) warned that pruned trees could lead to poor probability estimates. During pruning, the goal is to prevent the algorithm from overfitting the data. The classification accuracy is maximized, leading to highly homogenous leaves. Although they focus on algorithm C4.5, Provost and Domingos (2003) and Provost and Domingos (2000) also mention that classification trees generally provide poor estimates of the probabilities due to their design. Margineantu and Dietterich (2003) consider a Laplace correction as well as modifications to the pruning mechanism in an attempt to improve probability estimates with finite samples. Besides pruning that is applied once a tree is built, pruning that occurs due to early stopping has a negative effect on the probability estimates. The tree stops growing early

because of the default parameter setting of the complexity parameter in the *rpart* package, requiring a minimum improvement in relative error to create a new split. The smaller the number of terminal nodes in a decision tree, the fewer individual probability estimates there are. This partly leads to the poor performance of decision trees with default settings in estimating probabilities.

In terms of model assessment, Monte Carlo simulations are a popular tool to evaluate whether a model performs reasonably well. In the context of causal inference, Setoguchi et al. (2008) compare the ability of classification trees and other methods to properly estimate a causal effect in seven different scenarios where the logistic regression is correctly specified. For survey methods, Buskirk and Kolenikov (2015) consider random forests as an alternative to logistic regression for propensity score weighting. In both cases, the simulations assess the final result of the method rather than the quality of the individual probability estimates. Lastly, Chawla and Cieslak (2006) run experiments with unbalanced data sets that aim to assess the quality of probability estimates given by decision trees and ensembles of those trees, specifically bagged decision trees. They compare the performance of the estimates given by a decision tree with regular leaf frequencies, estimates that have been smoothed with a Laplace correction, and estimates produced by bagged trees. The performance measures the authors employ all use the class label as the truth. Since they use data sets where the true class describes an immutable state, the class label as the true probability is appropriate.

In this paper, we run extensive simulations that focus on the estimation of the probabilities rather than on a specific method that uses those probabilities. To obtain more realistic and practically useful results, the simulated models are inspired by real data sets, nine of them in total. All of the data sets have a target variable that represents a possible event rather than an immutable state. Each data set is used to determine two generative models: one based on the assumption of a logistic regression, and a second one that features a structure akin to a tree, which makes it misspecified in regard to the logistic regression.

The next section reviews consistency results for trees and forests, followed by Chap-

ter 1, which summarizes methods that use probability estimates and how they interact with the probabilities. Chapter 2 introduces the different data sets and the generative models that are used in the simulation. Chapter 3 presents the Monte Carlo simulations and the results. The conclusion will highlight the key findings and directions for potential future research.

Literature review

The consistency of a supervised learning algorithm for classification can have different meanings. One is the consistency of the algorithm when it is a class target predictor and the other is the consistency when it is a probability estimator. By making assumptions about the data or modifying CART or the random forest methodology, different authors have derived consistency results for the class target predictor. This means that individuals are properly ranked by the algorithm with respect to their real probabilities. Other results show the consistency of the target predictor in a regression context, which has been discussed to be applicable to the consistency of probability estimators for binary classification by some authors (e.g. Malley et al., 2012). In general, the consistency of probability estimators is a different problem than the consistency of class target predictors. As Breiman (1996) mentions, the probability estimates obtained with the standard decision tree construction method "are poor estimates of the true class probabilities". Thus, the ranking induced by the estimates from trees and forests may be correct, while the probabilities themselves remain biased. The following review includes consistency results for regression and classification trees and forests which are achieved by either altering the construction and splitting mechanism for the trees or making assumptions about the data.

An important theorem that serves as the basis of consistency results for trees and random forests is that of Devroye et al. (1996). The authors establish consistency results for the probability estimators in partitioning algorithms. More specifically, they prove consistency for classification rules that partition the data space into disjoint cells and make a classification in each cell depending on the majority of the label of the variables.

A major assumption in this theorem is that the label information is not used to construct the partition, meaning that the target variable may not play a role in the creation of the tree. In addition, the partitions may change with the number of observations and may depend on the data points. In regards to the CART methodology used to construct decision trees, the theorem presented by Devroye et al. (1996) is not applicable, since CART uses the label information to create the splits based on information gain at each node. If the labels are ignored for the creation of the partition, the performance of the decision tree or random forest will be poor and the algorithm might not converge in practice. Nevertheless, the findings of Devroye et al. (1996) are helpful in deriving further theorems and expanding the results to averaged classifiers, such as the random forest.

In a technical report, Breiman (2004) discusses the consistency of the target predictor in a regression context where the random forest algorithm is simplified. His proof also implies consistency for the class target predictor in the binary classification case and consequently this implies the consistency for the probability estimator. He compares the simplified model to an adaptive nearest neighbor method with a smart distance measure. This concept was introduced by Lin and Jeon (2002), who show that random forests can be viewed as weighted layered nearest neighbors and therefore can make use of their consistency properties. Breiman alters how individual tree splits are made in the original CART methodology. Instead of predefining the variable $mtry$, which defines how many variables are randomly selected at each iteration to choose the best split from, each variable is associated with a probability of inclusion. The individual probabilities sum to one. Then, at each node, a variable to split on is randomly selected based on its associated probability. Depending on whether a strong or weak variable is selected, the split is either made at the midpoint of all values or on a randomly selected value, respectively. It is assumed that the expected value of the target variable only depends on the strong variables. In addition to this alteration, no bootstrapping of the training data is performed. Breiman's analysis yields that bootstrapping does not play a role in proving consistency. In fact, it is the random selection of the split variables at each node that is crucial in proving consistency. He shows that the difference between the true response variable and the

estimated response variable goes to zero as the sample size increases and that the rate of convergence only depends on the strong variables. Breiman notes that the assumption of a linear loss function, meaning the midpoint of a variable is the best cut for a strong variable, could be problematic in practice since it depends on a large sample size in the nodes. If this is not the case, the individual target variable value will be more important for making cuts and getting consistent results.

Expanding on Breiman's work, other authors have discussed consistency results when making modifications to the algorithm used for the construction of trees and random forests. Biau et al. (2008) discuss the universal consistency of averaging rules. Random forests are an example of an averaged classifier and consist of a collection of randomized base tree classifiers. The classification forest outputs the majority vote of the class predicted by the individual trees. To prove that averaging classifiers are consistent if their base classifiers are, they first look at a purely random forest which was first considered by Breiman (2000). In this model, the individual trees are not split based on CART and an impurity measure. Instead, a random uniform selection of a leaf is made at each step of the construction. The split variable is then selected uniformly at random and the split itself is made according to a uniform random variable on the length of the selected split coordinate. This procedure is repeated k times and in the end, the majority vote for each class is taken in every leaf. The consistency applies to the binary class target predictor. The consistency of the classifier is equivalent to saying that the probability of error goes towards Bayes risk, which is the minimum between the probability that Y=1 given $\mathbf{x}$ and 1 minus this probability. In the binary classification setting, this also implies consistency of the probability estimator for this random splitting mechanism. The authors continue by proving how based on a consistent base learner, the voting classifier and averaging classifier are also consistent. Biau et al. (2008) further discuss that the standard Breiman's random forest classification algorithm, in which the trees are grown until each node contains one case, is not universally consistent.

In a different paper, Biau (2012) analyzes the random forest in a regression context and proves consistency of the target predictor. The author does an elaborate analysis of

the simple random forest proposed by Breiman (2004). Again, the splitting process is altered from the original CART method. For each node, a random variable is selected with a predefined probability, which could be determined with a second sample, and then a split is made on the midpoint of said variable. Biau (2012) shows that under a sparsity framework the rate of convergence of the algorithm only depends on the number of strong variables, which is very useful in high-dimensional regression when the number of variables can be much larger than the sample size.

Denil et al. (2013) take a different approach to prove consistency, as the authors analyze consistency in an online setting for multi-class classification tasks. The consistency refers to the class target predictor as well as the probability estimator. The authors adjust the conventional methodology for tree splitting by dividing the training data into a set of structure points and a set of estimation points. The structure points are used exclusively for the splitting process, i.e. for the split criterion and location of potential splits of the trees. The estimation points are used exclusively for making predictions, i.e. whenever a data point that is queried ends up in a leaf, said points are used to estimate the class probabilities. By splitting the points used for estimating and partitioning, the authors make use of the theorem proposed by Devroye et al. (1996) for partitioning algorithms and the consistency of the probability estimator is theoretically proven.

Other consistency results for random forests focus on assumptions on the data rather than the modification of the CART methodology when building the individual trees. Scornet et al. (2015) prove L2 consistency for the original random forest algorithm in an additive regression model context. The only difference to the original algorithm is that bootstrapping is replaced by subsampling, which doesn't play a role in proving consistency, as noted by Breiman (2004). The authors conclude that by controlling the variation of the regression function within each cell with either a good choice of the total number of leaves (when the trees aren't fully grown) or a good subsampling rate (when the trees are fully grown) L2 consistency can be ensured. However, n needs to be large enough so that the variation of the regression function within a cell of a random tree is small. Along with the additive regression requirement, all components must be continuous. The

authors make further assumptions for trees that are fully grown. According to Scornet et al. (2015), the presented theorems are the first consistency results for Breiman's forests given that all assumptions are satisfied. Since the consistency proof is in the regression setting, the consistency is valid for the predicted target itself, which is an average of the average target variable in each leaf of the trees.

Klusowski (2021) gives consistency results for decision trees in the additive regression context. The author expands on the paper of Scornet et al. (2015) and proves universal consistency for decision trees in the high dimensional additive regression model context. Hence, the consistency addressed refers to the predicted regression target variable. The standard CART methodology for tree construction is used with the minor adjustment that there can be only one observation remaining in any leaf. Other assumptions on the data are made in addition to it being expressed as an additive model. The author mentions that one of the issues with additive models is that they are not able to capture interaction effects between the predictor variables. This problem is solved by allowing for model misspecification. In general, it is difficult to incorporate interaction terms into a greedy modeling context (such as the standard CART) and thus it is unclear whether a general consistency theory can be developed. Both of the above proofs by Scornet et al. (2015) and Klusowski (2021) refer to the consistency of the target predictor in a regression setting. However, as Malley et al. (2012) examine, consistent target predictors in the regression setting supposedly give consistent probability estimators in the binary classification setting.

Malley et al. (2012) run a simulation study examining nonparametric models being used as probability machines. This is a slightly different approach to the standard classification algorithm and stems from the idea that consistent predictors in the regression setting will be consistent probability estimators for binary classification under the same conditions. They note that the conditional probability problem $P(Y = 1|\mathbf{x})$ is equal to the nonparametric regression problem $E(Y|\mathbf{x})$ and thus any algorithm that performs well for nonparametric regression also performs well for probability estimation for binary classification problems. Instead of applying the standard algorithm used for a classification random forest, the authors use the algorithm for regression random forests in the *ran-*

domForest R package. The deciding difference is the way in which the final outputs are predicted. In the classification case, a majority vote is taken in each terminal node and thus each leaf gives a class (0 or 1) as an output. Then, the final probability estimate given by the forest is obtained by taking the mean of these outputs. In the case of regression, the y values are averaged in each leaf and then averaged again over all trees. Malley et al. (2012) use four data sets to run simulations. For two of the data sets there are no"real" probability models and the target variables, which are appendicitis and diabetes, are immutable states. The results from the simulated data with available true probabilities show that the regression random forest does slightly better than the classification random forest for both data sets but it is not clear that either probability estimator is consistent based on the true and predicted probabilities.

In this paper, the focus lies on numerically assessing whether decision trees and random forests are consistent probability estimators when the target variable describes the occurrence of an event. The standard random forest algorithm for classification was used because it is most commonly applied in practice for these types of problems and the goal was to examine whether the standard classification algorithms are consistent probability estimators. As the literature has revealed, the consistency results for the regression random forests are closest to the algorithm used in practice. For classification trees, consistency results still rely on simplified versions of the algorithm. Thus, the objective is to see how classification trees and random forests perform empirically and whether they are consistent probability estimators.

Chapter 1

Methods using probability estimates

Various methods make use of the predicted probabilities in binary classification problems including model assessment methods, propensity scores, and uplift modeling. For some of these methods, the values of the probabilities are important and the selected algorithm must be a consistent probability estimator in order to get reliable outcomes from the chosen method. On the contrary, some methods are invariant to monotonic transformations of the estimated probabilities and a reliable ranking of the probabilities is sufficient. Several examples will be discussed and assigned to the appropriate scenario in this section.

1.1 Model assessment

Predicted probabilities are used in certain model assessment techniques. Consider for example receiver operating characteristic (ROC) curves and lift charts. The ROC curve is commonly used to measure the performance of a machine learning model in the binary classification setting. It plots the false positive rate versus the true positive rate achieved by the fitted machine learning model. The area under the curve (AUC) can then be calculated and used as a final measure. The larger this value, the better the model does in terms of classifying the target variable. One of the benefits of this performance measure is that it remains unchanged regardless of the selected classification threshold.

Every position along the curve corresponds to a different threshold and consequently the AUC takes all possible thresholds into account. Furthermore, it measures the ranking of the predictions, rather than the actual predicted values. Flach (2016) states that it is a normalized version of the Wilcoxon-Mann-Whitney sum of ranks test. Therefore, even if the actual predictions may be off by a large margin of error, the AUC will not be affected as long as the ranking of the predictions is good.

Another model assessment method is the lift chart. As per Blattberg et al. (2008), this tool allows companies to segment their customers into groups that classify which ones will be profitable when targeted in a marketing campaign. It does this by comparing the use of a machine learning model to target customers for marketing decisions versus randomly selecting them and obtaining the usual response rate. A graphical display of the lift shows the number or percentage of customers targeted on the x-axis and a ratio of the numbers of true responses on the y-axis. The point on the x-axis with the largest distance from the baseline (equivalent to a random guess) to the results from the model prediction has the highest lift score. Just as for the AUC, the lift chart is a ranking-based mechanism. It does not require the probability estimator to be consistent.

1.2 Propensity scores

Apart from their use in model assessment, probability estimates are utilized to create propensity scores in survey sampling methods. Rosenbaum and Rubin (1983) define propensity scores as the conditional probability of being assigned to a treatment given a group of observed covariates. They are used in survey sampling to imitate characteristics of randomized controlled trials, as the samples collected are non-random in nature and thus can have strong confounding bias. More specifically, they balance the individuals and create similar groups of treated and untreated subjects. In most cases, they are used for causal inference when the goal is to measure the effect of a treatment on a population. Austin (2011) summarizes four different methods in which propensity scores are used.

The first two are matching on the propensity score and stratifying on the propensity

score. These methods require the machine learning models to produce reliable rankings of class probabilities and not reliable class probabilities themselves. In matching, subjects will often be matched in a pair-wise fashion, where one untreated subject is matched with a treated subject that has the most similar propensity score. Based on a sample of matched individuals, the treatment effect can be estimated by subtracting the proportion of the untreated subjects experiencing a target variable event from the proportion of the treated subjects in a binary setting. Rosenbaum and Rubin (1983) describe taking the difference of the mean outcomes in a continuous target variable setting. Next, stratification on the propensity score is defined as ranking subjects based on their propensity scores and then separating the subjects into subsets. The subsets are created by using predefined thresholds of the propensity scores. There are common approaches to the creation of subsets, like dividing the subjects into five equal-sized groups using the propensity score. Just as for matching, this method relies on good rankings of the probabilities.

Other methods Austin (2011) discusses are the inverse probability of treatment weighting using the propensity score and covariate adjustment using the propensity score. In the first method, weights are created by using the inverse of the propensity score. Based on the weights, a new sample is created in which the distribution of measured covariates is independent of the treatment assignment. The idea is to weight the survey sample in a way that makes it representative of the true population. Thus, it becomes clear that in this scenario, a reliable probability estimate is needed. If the machine learning model is inconsistent for probability estimation, the weighting will be done incorrectly. In consequence, the estimated average treatment effect is unreliable. The last method discussed in the paper is covariate adjustment. Austin (2011) explains that in this method, a model is fitted on an indicator variable for the treatment status and the estimated propensity score to predict the outcome variable. Then, the treatment effect estimate is determined using the coefficient from the fitted regression model. It is crucial that the relationship between the propensity score and the outcome has been correctly modeled. The difference between the first three methods and the last is that they separate the design from the analysis of the study. However, as already emphasized, the last two methods both are sensitive to errors

in the values of the probability estimates used as the propensity scores.

1.3 Uplift modeling

In uplift modeling, the goal is to measure the causal effect that a treatment such as a marketing campaign has on customer behavior. Gutierrez and Gérardy (2017) review the different types of uplift modeling techniques that have been studied in the literature. The three main approaches for uplift modeling are the two-model approach, the class transformation method, and lastly modeling the uplift directly. In the two-model approach, there are two separate models which are based on the treatment and the control group respectively. The predictions of the control-based model are subtracted from the predictions of the treatment-based model to yield the uplift of an individual. The class transformation method creates a new target variable which is not only a binary variable indicating a positive response or not but rather a multiplication of responses and new indicators for whether an observation belongs to the treatment or control group. Details are explained by Gutierrez and Gérardy (2017). Lastly, modeling uplift directly takes into account the change in response between treatment and control groups directly in the model fitting process.

Several authors discuss modeling the uplift directly using the decision tree algorithm. Since it is the change in the class probability that is of interest in uplift modeling, using the direct modeling approach can yield better results. Rzepakowski and Jaroszewicz (2012) use decision tree algorithms based on CART and 4.5 and make adjustments to the splitting criteria and pruning methods. The approach is based on the idea of wanting to maximize the differences between the treatment and control class distributions. Hence, they use the Kullback-Leibler divergence, Euclidean distance, and the chi-squared divergence statistics that serve as splitting criteria. Other authors that propose methods adjusting the splitting criteria of decision trees to model uplift directly are Chickering and Heckerman (2000) and Radcliffe and Surry (2011).

The two-model approach has the advantage of being simpler than the other two meth-

ods and that any classification model can be implemented. Furthermore, Zaniewicz and Jaroszewicz (2013) state that if the uplift is strongly correlated with the target class or if the number of training observations is large enough, the models can give accurate probability estimates. Gutierrez and Gérardy (2017) write that it has been observed to perform well. However, Radcliffe and Surry (2011) show scenarios in which using two models causes the inability to capture certain uplift signals and this approach is outperformed by the other models such as direct uplift modeling. In general, since there are two separate probabilities in this approach, the effect of biased estimates can be even stronger than in a single classification model.

1.4 Cost and profit-based modeling

In settings where it is important to take into account the increase in revenue or the differing costs of a false positive versus a false negative, it is crucial to use consistent probability estimators. In the uplift modeling setting, two-stage modeling is a process in which two separate models are fit. First, a model is created to predict whether a customer that has received a treatment will purchase a product or not. Second, a model is fitted in order to predict the number of sales that will be made through each individual. Gubela et al. (2020) propose different types of two-stage uplift models. One such model includes applying a classification model to identify customers that will make a purchase such that the revenue exceeds zero and a regression model built on these buyers to predict the response value, such as the purchase volume. The use of consistent probability estimators is necessary to get the most reliable expected revenues.

Other settings that rely on consistent probability estimators are ones in which misclassification costs differ depending on the scenario. Zadrozny and Elkan (2001) describe that in the medical field, for example, the cost of prescribing a drug to a patient with an allergy is significantly higher than not prescribing a drug to a person without any allergies, given that alternative treatments are available. Another example is one-to-one marketing. The cost of not contacting a person who would respond positively to an offer is much higher

than the cost of contacting a person who does not take the offer. For these problems, the goal is to assign the class to an observation that will lead to the lowest expected cost. In the case that the cost is known in advance, this is a multiplication of the probability that an instance belongs to class 1 and the cost associated with predicting class 0 if the true class is class 1. Obtaining accurate expected costs is important in these cases.

1.5 Summary

Table 1.1 summarizes the discussed methods and classifies them into whether they are invariant to monotonic transformations of the probability estimates or whether they are methods that may be negatively affected by the bias (or in general, lack of consistency) of the probability estimates. The former is listed under the "Invariant" column and the latter under the "Affected by bias" column. It is important to choose an appropriate machine learning model for the latter methods to avoid the negative effects of biased estimates. The experiments described in this paper will highlight the differences between logistic regression, a method that is known to be a consistent probability estimator under reasonable assumptions, decision trees, and random forests. For classification problems, we could not locate proofs of consistency for the standard tree and random forest construction methods in the *rpart* and *randomForest* R packages when they are used as probability estimators. The next chapter will introduce the data sets used in the Monte Carlo simulations.

Table 1.1: Summary of methods using probability estimates.

Invariant	Affected by bias
ROC curve	Propensity score weighting
Lift chart	Two-model approach (uplift)
Propensity score matching	Two-stage modeling (uplift or other)
Propensity score stratification	Cost-based decisions

Chapter 2

Presentation of the data sets and true probability models

In order to assess the ability of tree-based methods to give reliable probability estimates, Monte Carlo experiments were conducted for nine different binary classification data sets. The data sets were used to establish two separate probability models. The first model called LGM was based on a logistic regression function. The second one, called TM, was based on a structure akin to a decision tree, making it a misspecified model for the logistic regression. The data sets were used to determine the "true models" by fitting either a logistic regression (for the LGM) or a decision tree (for the TM). The R *glm* and *rpart* functions were used to this end. For one of the data sets that did not include a target variable, the true probabilities were generated from manually determined equations rather than a fit. Details of the fit to determine the "true models" will be discussed in the corresponding data section.

To generate the probabilities for the LGM, a logistic regression was fitted to the original data set and then a bi-directional stepwise based on AIC was run. The variables that were selected during the stepwise process were then used to fit a logistic model whose estimated parameters were thereafter considered as the "true values". These lead in turn to true probabilities for all observations of the data set. For every iteration of our simula-

tion, new target variables were then generated as Bernoulli variables with a probability of success equal to those true probabilities that remained unchanged throughout the Monte Carlo experiment.

For the TM, the true probabilities were generated based on a decision tree rather than a logistic model, meaning that a logistic model would be misspecified in this scenario. The tree was fitted to the original data without pruning. The idea behind this strategy was to let the tree grow to a large enough size so that a larger amount of unique probabilities would be generated. In fact, it is noted in the literature that the pruning mechanism of trees and their size influence the probability estimates. Larger grown trees can be better for probability estimation, as discussed by Provost and Domingos (2003). Furthermore, Setoguchi et al. (2008) and Bauer and Kohavi (1999) found in their experiments that pruning mechanisms increased the bias of estimates versus their unpruned counterparts. Apart from the complexity parameter, the maximum depth parameter was adjusted according to the size of the data set. For example, if the data set had 14,927 observations, the maximum depth that could be set was 11. This would lead to a maximum of $2^{11} = 2048$ leaves, which means that the average number of observations per leaf corresponds to the default minimum of 7 observations per node. Once the decision tree was fitted to the data, it was deemed to be the data-generating process, or the "truth", and provided a "true probability" for each individual in the data set. Those true probabilities were then used in the Monte Carlo simulation to generate new target variables from a Bernoulli at each repetition of the simulation.

Table 2.1 summarizes each data set. The post-processing column shows the number of observations and variables that remained once the data sets were processed and ready to be used for the simulation experiments. The target variable is included in the "Variables" column. The following subsections provide detailed descriptions of each data set.

Table 2.1: Data set summaries.

Data set	Original		Post-processing	
	Observations	Variables	Observations	Variables
Census	31,997	651	6,000	11
Iowa recidivism	26,020	17	26,020	10
HMEQ	5,960	14	5,960	22
German Credit	1,000	21	1,000	21
Marketing Promotion	64,000	9	12,800	9
Bank Marketing	4,119	21	4,118	21
Churn	10,000	14	10,000	11
Internet Churn	72,274	11	14,916	10
Loan	9,578	14	9,578	14

2.1 Data set descriptions

2.1.1 Census data

Buskirk and Kolenikov (2015) compare random forest and logistic regression models in response propensity weighting and stratification. The propensity scores are used to correct for nonresponse biases and consequently ensure the validity of survey estimates. For the data in their experiment, the authors use information from the 2012 US Health Interview Survey. They keep the variables with no missing values in the original data and use two separate probability models to generate true probabilities. In this case, the data set does not contain a target variable. The models are generated by hand as possible data-generating processes. A similar process was followed for our version of these simulations, but our goal was to look at the quality of the probability estimates. We took data from the 2019 US Health Interview Survey[1] which includes variables from the Sample Adult Interview and Paradata files. Eleven predictor variables with no missing values in the original data and based on the selection of Buskirk and Kolenikov (2015) were chosen to create the new data set. More specifically, nine categorical variables, one binary, and

[1] https://www.cdc.gov/nchs/nhis/2019nhis.htm. Accessed: 2022-02-15

one numerical variable were selected. The target variable for this data set was whether an individual responded to the survey or not. Table 2.2, presented at the end of the chapter, shows a summary of the variables included in this experiment.

Most of the variables the authors use in their data set were recovered in the 2019 census data. However, some of the variables were excluded or replaced. The household telephone status (telstat) and the number of working cellphones in the household (wrkceln) were replaced by TELCEL_A and TELCURWRK_A. Furthermore, the overall functional limitation variable (alchronr2) was replaced with a general health question (PHSTAT_A) and the late sample adult interviews (lateinta) and employment status (wrkcata) variables were not included as they weren't available in the 2019 census data. A sample of 6,000 was taken from the original sample size of 31,997 for the simulations.

Several preprocessing steps were undertaken to prevent errors in the simulations. For TELCEL_A, TELCURWRK_A, and PHSTAT_A two of the categories (or survey answers) had a very low number of instances (less than 0.2% of the data). The categories indicated a respondent answering with "Don't Know" or "Refused" for the three survey questions. The rows with these two answers were therefore deleted. Furthermore, three of the categories for the education level variable were combined to create a sufficient number of entries for each category.

Once the preprocessing was completed, the true probabilities representing the response probabilities of the surveyees were created and target variables simulated. The probabilities were calculated using slight variations of the equations presented by Buskirk and Kolenikov (2015). The first equation, referred to in this paper as LGM, is a simple logistic regression interaction term model. For this experiment, the same coefficients as in the paper were used.

Logistic regression-based probability model (LGM)

$$P(Y=1|\mathbf{x}) = \frac{1}{1+e^{-1.63+0.028\,\text{age}+0.48(\text{sex=female})-0.57(\text{race=black})+0.32(\text{race=white})}} \tag{2.1}$$

The second model, referred to as TM, has more complex interactions between the variables and tree-like cuts. The coefficients differ slightly from the ones used by Buskirk and Kolenikov (2015). The adjustment was made to get a similar distribution of the probabilities compared to the LGM. Note that the census data did not come with a target variable like the other eight data sets. For this reason, the definition of the LGM and TM followed a different process. While a fit on the other eight data sets yields the true models, manually defined equations were utilized for the census data to generate the true probabilities.

$$\begin{gathered}
\text{Tree-based probability model (TM)} \\
P(Y=1|\mathbf{x}) = 0.01\left[\text{age} - 2\left(\frac{\text{age}}{50}\right)^5\right] - 0.07(\text{income} < \$35,000) - \\
0.06(\text{income} > \$100,000)(\text{education} < \text{high school}) \\
+0.1(\text{education} \in \{\text{high school, some college, bachelors}\}) \\
+0.2(\text{education} > \text{bachelors}) + 0.04(\text{sex=female}) \\
+0.02(\text{has cell phone})\{-0.04 + [0.04(\text{sex=female}) + 0.02(\text{sex=male})](51 - \text{age})\} \\
+0.08(\text{has landline})[0.06(\text{sex=female}) + 0.01(\text{sex=male})](\sqrt{\text{age}} - 5.5) \\
+0.06(\text{sex=male})
\end{gathered} \tag{2.2}$$

Once the true probabilities were obtained using these two probability models, the target variables were simulated for each of them. This was repeated 1,000 times (once for each repetition of our simulation) for all observations and for both probability models. The percentage of respondents was about 57%.

2.1.2 Iowa recidivism data

The Iowa recidivism data set includes information about criminal offenders being re-admitted to prison within a three-year period. The recidivism rate is around 33%. The data is provided by the Iowa Department State of Corrections under the Creative Commons

Attribution 4.0 International Public License[2]. Preprocessing for this data set required the removal of several variables that described the recidivism that occurred in more detail and could thus not be used as predictors. Additionally, the offense type variable was removed due to collinearity with the offense subtype variable, and the prison release year was removed due to collinearity with the recidivism reporting year. The difference between these years was three for every individual. For the race and offense classification variables, several categories were combined to create sufficient entries in every category. Moreover, one row was removed that contained the single occurrence of a category for the release type variable. Other rows containing missing values or single categories with a very low number of entries (less than 0.07% of the data set) were removed. The final data set used for the experiment is summarized at the end of the chapter in Table 2.3. The target variable describes whether recidivism within three years for an individual occurred.

2.1.3 HMEQ data

The Home Equity (HMEQ) data set contains information about bank customers and whether they defaulted on past home equity loans or were seriously delinquent[3]. The target variable includes 80% of individuals who repaid their loans and 20% who defaulted. In this data set, there was a significant amount of missing values for nine of the ten numerical variables. As a solution for these missing values, additional indicator variables were created for each of them, indicating a 1 if there was a missing value. The missing values were then imputed by the mean value of the non-missing values of each respective variable. For the two categorical variables, there were several missing values. These were treated as an additional category. The details of the variables are described in Table 2.4.

[2]https://data.iowa.gov/Public-Safety/3-Year-Recidivism-for-Offenders-Released-from-Pris/mw8r-vqy4. Accessed: 2022-03-04

[3]https://www.kaggle.com/ajay1735/hmeq-data. Accessed: 2022-02-10

2.1.4 German credit data

The German Credit data set (Dua and Graff, 2017)[4] is a collection of bank customer data that classifies a customer as either a "good" (70%) or "bad" (30%) credit risk. It contains 1,000 observations and 20 predictor variables, which include eleven categorical, two binary, and seven numerical variables. The binary variables V19 and V20 were treated as numerical values. For V4, two of the eleven categories were combined to ensure an adequate amount of entries in all categories. The variables of this data set are described in detail in Table 2.5.

2.1.5 Marketing promotion campaign data

The Marketing Promotion Campaign Uplift Modelling data set contains customer information and the historical use of discounts or Buy One Get One free promotions (Moro et al., 2014)[5]. The target variable is an indication of whether the customer bought a new offer or not. About 15% of the customers make a purchase after receiving an offer and 85% do not. Few preprocessing steps were performed for this data set. First, the offer variable was transformed from a categorical into a binary variable by changing the entries to one if they were "Discount" or "Buy One Get One" and zero otherwise. Second, a sample of the data was taken to conduct the experiment, which was 20% of the original number of observations. Table 2.6 gives a detailed description of the variables.

2.1.6 Bank marketing data

This data set comprises information about clients and direct marketing campaigns of a Portuguese banking institution[6]. The target variable describes whether a client subscribes to a term deposit or not. Around 11% of the clients subscribe and 89% do not. The data set that was used has 10% of the original observations. It has 4,119 observations and 20

[4]https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data). Accessed: 2022-03-02
[5]https://www.kaggle.com/datasets/davinwijaya/customer-retention. Accessed: 2022-01-24
[6]https://archive.ics.uci.edu/ml/datasets/bank+marketing. Accessed: 2022-03-09

predictor variables. For data preprocessing, the one entry of "illiterate" for the education variable was changed to "unknown" to ensure enough entries per category. For the default variable, the one row with the answer "yes" was removed and only the answers "no" and "unknown" remained. The default and contact variables were both treated as numerical values. The final data set used for the experiment is summarized in Table 2.7.

2.1.7 Churn data

The next data set used in these experiments is a churn data set for a bank indicating whether an individual left the bank or not[7]. In total, 20% of the individuals exit the company and 80% do not. A 20% churn rate appears to be rather large and therefore it seems reasonable to assume that the data could have been undersampled. The documentation of the data set does not refer to such preprocessing steps. The original data set has 10,000 observations and 13 predictor variables, including two categorical, three binary, and eight numerical variables. For this experiment, the row number, surname, and customer ID were removed. No additional preprocessing was required. The detailed description of the variables is shown in Table 2.8.

2.1.8 Internet churn data

The internet churn data set contains information about customers using different internet services including subscription type, duration, bills, and internet usage[8]. The target variable indicates whether a customer canceled the service. There is a 55% churn rate. As has been noted for the previous data set, it seems like the data could have been undersampled. Corresponding preprocessing steps were not included in the data set documentation. Since the original data set has over 72,000 observations, a sample of 15,000 was used for this simulation. Additional preprocessing steps included the removal of the id and download_avg variables. Download_avg was removed to prevent predictions of

[7]https://www.kaggle.com/datasets/mathchi/churn-for-bank-customers. Accessed: 2022-03-26
[8]https://www.kaggle.com/datasets/mehmetsabrikunt/internet-service-churn. Accessed: 2022-03-27

zero and one in the LGM and the id was redundant. Next, the rows with missing values for upload_avg were removed as they made up less than 0.6% of the observations. For the remaining_contract variable, a significant amount of entries had missing values. As a consequence, an indicator variable was added to the data set and the missing values were replaced by the mean of the non-missing data. The description of the variables used in the experiments is shown in Table 2.9.

2.1.9 Loan data

Finally, the loan data set contains information about people who borrowed money and whether they paid back the loan or not[9]. Around 16% of the people have not paid their loans back. No preprocessing was necessary for this data set as there were no missing values and no collinearity issues. Table 2.10 summarizes all variables.

[9]https://www.kaggle.com/itssuru/loan-data. Accessed: 2022-03-09

Table 2.2: Variable descriptions for the census data.

Name	**Description**	**Type**
RESPONSE	Response to the survey	Target variable
SEX_A	Sex: male or female	Binary
AGEP_A	Age	Numerical
EDUC_A	Education level	Categorical (8)
REGION	Region of residence	Categorical (4)
TELCEL_A	Do you have a working cellphone?	Categorical (7)
TELCULWRK_A	Is there at least one telephone inside your home that is currently working and is not a cell phone?	Categorical (7)
INCGRP_A	Total combined family income group	Categorical (7)
PHSTAT_A	Would you say your health in general is excellent, very good, good, fair, or poor?	Categorical (8)
RACEALLP_A	Race	Categorical (9)
RATCAT_A	Ratio of family income to poverty threshold	Categorical (9)
HISPALLP_A	Single and multiple race groups with Hispanic origin	Categorical (9)

Table 2.3: Variable descriptions for the Iowa recidivism data.

Name	**Description**	**Type**
Return to Prison	Occurrence of recidivism	Target variable
Recidivism Reporting Year	Fiscal year of reporting	Numerical
Main Supervising District	The judicial district supervising the offender for the longest time during the tracking period	Categorical (11)
Release Type	Reasoning for offender's release from prison	Categorical (12)
Race Ethnicity	Offender's race and ethnicity	Categorical (8)
Age At Release	Age group at release from prison	Categorical (5)
Sex	Sex: male or female	Binary
Offense Classification	Offense classification defines the type of penalty received	Categorical (9)
Offense Subtype	Further classification of the most serious offense for which the offender was placed in prison	Categorical (24)
Target Population	Indicates prisoners who are on parole	Binary

Table 2.4: Variable descriptions for the HMEQ data.

Name	**Description**	**Type**
BAD	Client defaulted on the loan	Target variable
LOAN	Amount of requested loan	Numerical
MORTDUE	Amount due on existing mortgage	Numerical
VALUE	Value of current property	Numerical
REASON	Reason for loan request	Categorical (3)
JOB	Job	Categorical (7)
YOJ	Years at current job	Numerical
DEROG	Number of major derogatory reports	Numerical
DELINQ	Number of delinquent credit lines	Numerical
CLAGE	Age of oldest tradeline in months	Numerical
NINQ	Number of recent credit lines	Numerical
CLNO	Number of credit lines	Numerical
DEBTINC	Debt-to-income ratio	Numerical
MISS_MORTDUE	Indicator variable for originally missing MORTDUE values	Binary
MISS_VALUE	Indicator variable for originally missing VALUE values	Binary
MISS_YOJ	Indicator variable for originally missing YOJ values	Binary
MISS_DEROG	Indicator variable for originally missing DEROG values	Binary
MISS_DELINQ	Indicator variable for originally missing DELINQ values	Binary
MISS_CLAGE	Indicator variable for originally missing CLAGE values	Binary
MISS_NINQ	Indicator variable for originally missing NINQ values	Binary
MISS_CLNO	Indicator variable for originally missing CLNO values	Binary
MISS_DEBTINC	Indicator variable for originally missing DEBTINC values	Binary

Table 2.5: Variable descriptions for the German credit data.

Name	Description	Type
V21	Customer is a “bad” credit risk	Target variable
V1	Status of existing checking account	Categorical (4)
V2	Duration in months	Numerical
V3	Credit history	Categorical (5)
V4	Purpose	Categorical (10)
V5	Credit amount	Numerical
V6	Savings account/bonds	Categorical (5)
V7	Present employment since	Categorical (5)
V8	Installment rate in percentage of disposable income	Numerical
V9	Personal status and sex	Categorical (4)
V10	Other debtors/guarantors	Categorical (3)
V11	Present residence since	Numerical
V12	Property	Categorical (4)
V13	Age in years	Numerical
V14	Other installment plans	Categorical (3)
V15	Housing	Categorical (3)
V16	Number of existing credits at this bank	Numerical
V17	Job	Categorical (4)
V18	Number of people being liable to provide maintenance for	Numerical
V19	Telephone	Binary
V20	Foreign worker	Binary

Table 2.6: Variable descriptions for the marketing promotion campaign data.

Name	**Description**	**Type**
conversion	Customer makes a purchase after the offer	Target variable
variable recency	Months since last purchase	Numerical
history	Dollar value of the historical purchases	Numerical
used_discount	Indication of whether a customer used a discount before	Binary
used_bogo	Indication of whether a customer used a buy one get one free (bogo) before	Binary
zip_code	Zip code as suburban/urban/rural	Categorical (3)
is_referral	Indicates if the customer was acquired from a referral channel	Binary
channel	Channels that the customer is using	Categorical (3)
promotion	Indication of whether a customer received either the bogo or the discount	Binary

Table 2.7: Variable descriptions for the bank marketing data.

Name	**Description**	**Type**
y	Subscription to a term deposit	Target variable
age	Age	Numerical
job	Job type	Categorical (12)
marital	Marital status	Categorical (4)
education	Education level	Categorical (7)
default	Has credit in default?	Binary
housing	Has housing loan?	Categorical (3)
loan	Has personal loan?	Categorical (3)
contact	Contact communication type	Binary
month	Last contact month of year	Categorical (10)
day_of_week	Last contact day of the week	Categorical (5)
duration	Last contact duration, in seconds	Numerical
campaign	Number of contacts performed during this campaign and for this client	Numerical
pdays	Number of days that passed by after the client was last contacted from a previous campaign	Numerical
previous	Number of contacts performed before this campaign and for this client	Numerical
poutcome	Outcome of the previous marketing campaign	Categorical (3)
emp.var.rate	Employment variation rate (quarterly indicator)	Numerical
cons.price.idx	Consumer price index (monthly indicator)	Numerical
cons.conf.idx	Consumer confidence index (monthly indicator)	Numerical
euribor3m	Euribor 3 month rate (daily indicator)	Numerical
nr.employed	Number of employees (quarterly indicator)	Numerical

Table 2.8: Variable descriptions for the churn data.

Name	**Description**	**Type**
Exited	Customer exited the bank	Target variable
CreditScore	Credit score	Numerical
Geography	Country	Categorical (3)
Gender	Gender: male or female	Binary
Age	Age	Numerical
Tenure	Number of years that the customer has been a client	Numerical
Balance	Bank balance	Numerical
NumOfProducts	Number of products a customer has purchased through the bank	Numerical
HasCrCard	Indicates whether a customer has a credit card	Binary
IsActiveMember	Indicates if the customer is active	Binary
EstimatedSalaray	Estimated salary	Numerical

Table 2.9: Variable descriptions for the internet churn data.

Name	**Description**	**Type**
churn	Customer canceled the service	Target variable
is_tv_subscriber	Indicates whether the customer has a TV subscription	Binary
is_movie_package_subscriber	Indicates whether the customer has a movie subscription	Binary
subscription_age	The number of years a customer has used the service	Numerical
bill_avg	Last three months billing average	Numerical
remaining_contract	The number of years remaining for a customer's contract	Numerical
service_failure_count	Number of times a customer has called for service failure in the last three months	Numerical
upload_avg	Last three months upload average (GB)	Numerical
download_over_limit	Amount of GB with which a customer exceeds their allowable download limit	Numerical
MISS_remaining _contract	Indicates whether there was a missing value in the original remaining contract variable	Binary

Table 2.10: Variable descriptions for the loan data.

Name	**Description**	**Type**
not.fully.paid	Individual has not repaid the loan	Target variable
credit.policy	Indicates whether the customer meets the credit underwriting criteria of LendingClub.com	Binary
purpose	Purpose of the loan	Categorical (7)
int.rate	Interest rate of the loan	Numerical
installment	Monthly installments owed by the borrower if the loan is granted	Numerical
log.annual.inc	The natural log of the self-reported annual income of the borrower	Numerical
dti	The debt-to-income ratio of the borrower	Numerical
fico	FICO credit score of the borrower	Numerical
days.with.cr.line	The number of days the borrower has had a credit line	Numerical
revol.bal	The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle)	Numerical
revol.util	The borrower's revolving line utilization rate (the amount of the credit line used relative to the total credit available)	Numerical
inq.last.6mths	The borrower's number of inquiries by creditors in the last 6 months	Numerical
delinq.2yrs	The number of times the borrower has been 30+ days past due on a payment in the last 2 years	Numerical
pub.rec	The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments)	Numerical

Chapter 3

Monte Carlo simulations

While some consistency results for decision trees and random forests as class target predictors exist, the consistency as probability estimators in the case of mutable states has not been theoretically proven for the algorithms used in practice. Therefore, we evaluate numerically the ability of the CART and random forest algorithms to provide good probability estimates. To this end, six machine learning models were selected, which include two logistic regression models, as they are proven to be consistent probability estimators when correctly specified, two decision trees, and two random forests. They are described in detail in the next section.

With nine different data sets, there were a total of eighteen true probability models; nine LGMs (that are well-specified logistic regressions) and nine TMs (that are misspecified models for a logistic regression and feature a structure that should be advantageous to trees and forests). For each of the eighteen probability models, the six machine learning models were trained and then used to give predictions for a validation set. This was done in a Monte Carlo simulation with 1,000 repetitions. While the predictor variables and the original sample size remained unchanged, a new target variable was created in each repetition based on the true probabilities from the LGM and TM. Training and validation sets were generated by randomly partitioning the data into 80% training and 20% validation set. This split was performed for each repetition.

Once the predictions for each machine learning model in the simulation were obtained, the performance was measured in terms of bias and RMSE for each individual true value. The bias was calculated by subtracting the true probability of an observation from the predicted probability for that observation. The RMSE was calculated for each observation over all iterations. More specifically, for each observation, the true probability was subtracted from the predicted probability and squared. Then the mean and square root were taken. Since the validation sets were randomly generated at each iteration, a given observation was not always selected in the validation set. Thus the sample size used for the RMSE calculation was not the same for every individual. It represents the total number of times an observation occurred in the validation sets throughout the simulation. In the next section, the six machine learning algorithms are described in detail.

3.1 Machine learning model selection

3.1.1 Logistic regression

The first model is a simple logistic regression model fit on all variables using the *glm* function in R with binomial family setting.

3.1.2 Logistic regression with interaction terms and step function

The second logistic regression model required additional steps. First, interaction terms for all the numerical variables in a data set were created. These terms were a single multiplication of two variables at a time. Certain interaction terms were removed if they caused rank-deficiency problems. Binary variables were included to create these terms except for the new indicator variables of the HMEQ data set since this would have led to computational issues. Multiplications of binaries times themselves were removed. A logistic regression model was fitted on the new data set including original variables and new interaction term variables. Then, a stepwise algorithm running in both directions was used as a variable selection strategy. The default settings of the R step function were kept.

3.1.3 Decision tree with default parameter settings

The decision tree model was applied using all default parameters of the R *rpart* function. The default complexity parameter (cp) of 0.01 requires a minimum improvement of 0.01 of the relative error to create a new split. The tree stops growing if no split can improve said error by at least 0.01 or when the number of observations in a node ($minsplit$) is less than 20. The $maxdepth$ parameter is set to 30 by default and determines the maximum depth of the tree. The root node has a depth of zero.

3.1.4 Decision tree with adjusted parameter settings

For the second decision tree model, three of the hyperparameters were adjusted. To this end, a grid search was performed. The values for the hyperparameter search included $minsplit$ set to 1, 2, 5, 10, 20, $maxdepth$ set to 3, 5, 10, 20, and cp set to -1 or 0.01. The different hyperparameter values were chosen so that the tree has the option to grow larger than the default cp permits, but not too large ($maxdepth$) so that there would still be many observations in the leaves. The area under the receiver operating characteristics (AUC) curve was calculated for every hyperparameter combination and the combination with the highest AUC was selected for the simulations. This AUC was selected as a performance measure to mimic the model selection process in a practical setting when the true probabilities are unknown.

3.1.5 Random forest with default parameter settings

The random forest model utilized the default parameters in the *randomForest* function of R. The model runs in classification mode if the target variable is a factor. The trees in the forest are grown to their maximum size in this setting. The default number of minimum observations in the terminal nodes is one ($nodesize$). The default number of trees grown in the forest ($ntree$) is 500, and the default value for the number of variables randomly sampled as candidates for each split ($mtry$) is the square root of the number of variables.

3.1.6 Random forest with adjusted parameter settings

For the adjusted random forest model, three of the hyperparameters were changed from their default values. A grid search was performed to determine the best combination. The parameters were *ntree* with the values 300, 500, 1000, *nodesize* with 1, 50, 100, 500, and *mtry* with 3,6, and 9. Since the trees in the forests grow to their maximum possible size, the *nodesize* values included larger values of up to 500. The AUC was calculated for every hyperparameter combination and the combination with the highest area was selected. Note that for the marketing data set, the mtry parameter 9 was removed from the hyperparameter search since there are only eight predictor variables in the data set. Both the adjusted tree and adjusted random forest have a small advantage over the other models because the hyperparameters were selected based on their performance on a validation set. They are implicitly the best of a collection of models (that were included in the grid search), thus this could slightly bias the performance to their advantage.

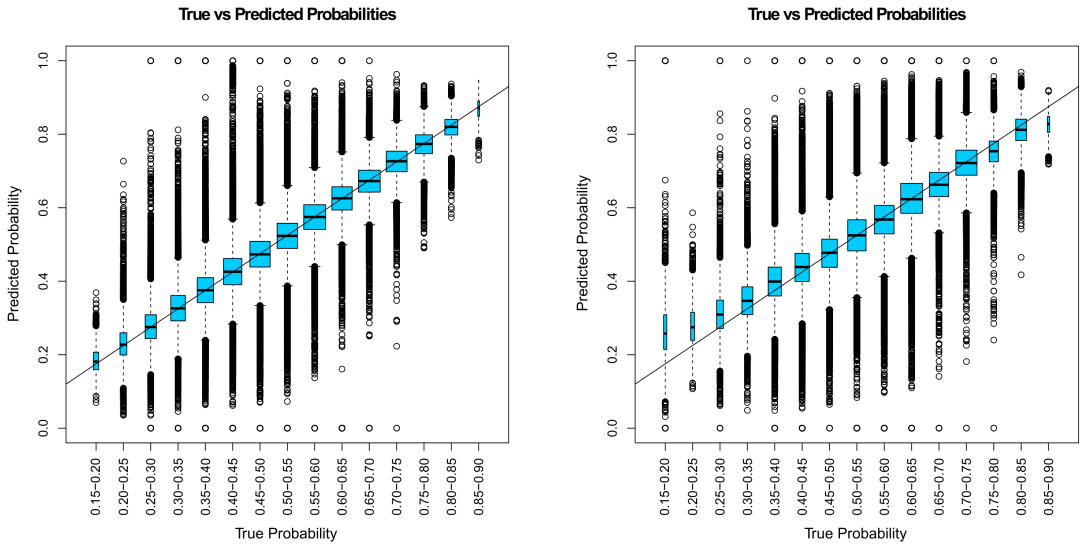
3.2 Results

In the following sections, the results of the six machine learning algorithms are presented for each data set. First, the generated probabilities from the LGM and TM are discussed. This is followed by a comparison of the bias and RMSE for all algorithms as well as their predictive behaviors. The selected hyperparameters for the tree-based algorithms are also presented for each data set. Lastly, the performances of the algorithms will be summarized over all nine data sets at the end of the chapter based on two different ranking systems.

3.2.1 Census data

The generated probabilities of the LGM and the TM followed very similar distributions. The distribution for the LGM had a mean, median, and standard deviation of 0.5622, 0.5710, and 0.1514 respectively. The probabilities ranged from 0.1550 to 0.8683.

For the TM, the mean, median, and standard deviation were 0.5815, 0.5986, and 0.1355 respectively. The probabilities ranged from 0.1694 to 0.8569. For both probability models, there is a similar amount of probabilities in the middle of the probability range. Figure 3.1 shows the true versus predicted probabilities of the logistic regression model for both the LGM and TM. The boxplots illustrate through varying widths that the number of entries that fall in the groups 0.35-0.40, 0.40-0.45, to 0.70-0.75 are almost the same. There are fewer entries for the groups that are closer to the extreme ends of the range, i.e. groups 0.15-0.20, 0.20-0.25, 0.80-0.85, and 0.85-0.90. All probability boxplots for the LGM and TM for the census data are presented in the appendix in Figure 1 and Figure 2, respectively. The hyperparameters selected by the grid search for the adjusted decision tree and adjusted random forest for both probability models are shown in Table 3.1.



(a) LGM (b) TM

Figure 3.1: Census data: true probabilities vs predicted probabilities for the logistic regression based on the LGM and TM. Individual values of all data points from 1,000 replicates are displayed.

The detailed bias and RMSE results of the Monte Carlo simulations are listed in Table 1 in the appendix. For both probability models, the logistic regression with step function performs best for both bias and RMSE with median values of 0.0000 and 0.0230 for the

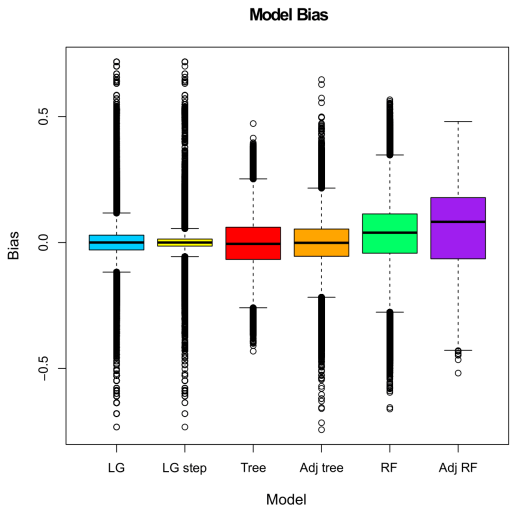
Table 3.1: Census data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(20, 3, -1)	(1000, 100, 3)
TM	(20, 5, -1)	(300, 500, 6)

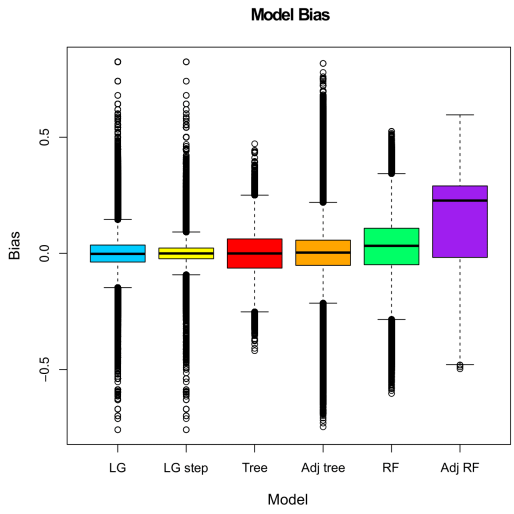
LGM and values -0.0003 and 0.0338 for the TM respectively. The simple logistic regression model follows closely. The next best are the decision tree models, with the adjusted tree performing better than the default tree for the LGM (median RMSE of 0.0683 versus 0.0801), but slightly worse for the TM (median RMSE of 0.0937 versus 0.0759 respectively). It is important to note that the decision tree with default parameters gives only a few unique values of probabilities as predictions in a single iteration. Thus, it is also insightful to look at the true probabilities versus predicted probabilities (see Figure 1 and Figure 2 in the appendix). For both probability models, the majority of the predictions of the default decision tree stay between 0.35 and 0.65. Lastly, the random forest models perform the poorest for the census data. Especially for the TM, the adjusted random forest has a very large median RMSE of 0.2505. Examining the true versus predicted probability graph reveals that the model has pushed a majority of the predictions to either zero or one.

3.2.2 Iowa recidivism data

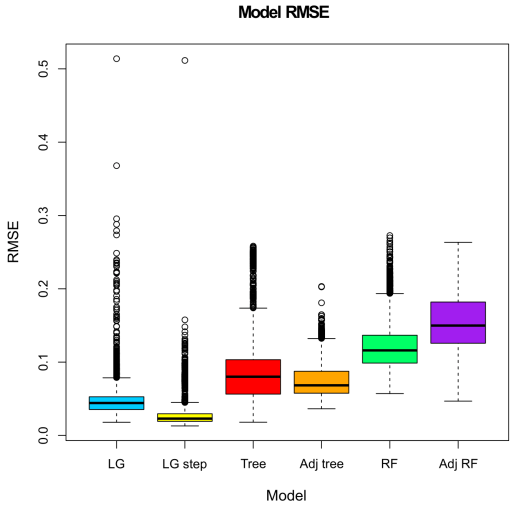
For the Iowa recidivism data set, the distribution of the LGM had a mean, median, and standard deviation of 0.3338, 0.3357, and 0.1410. The range was 0.0000 to 0.8058. For the TM, the mean, median, and standard deviation were 0.3338, 0.3333, and 0.1929. The probabilities ranged from 0 to 1. For the LGM, the middle probability groups are slightly more populated than the groups on the ends. In general, groups larger than 0.60 aren't heavily populated. There is a trend where the entries per group increase slightly until around 0.45 and then decrease thereafter. For the TM, the number of entries per



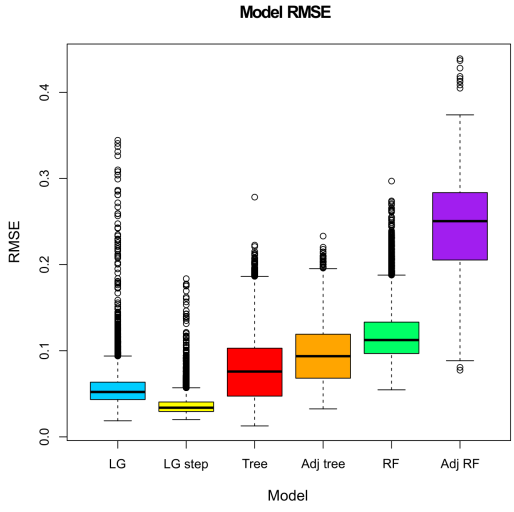
(a) LGM bias



(b) TM bias



(c) LGM RMSE

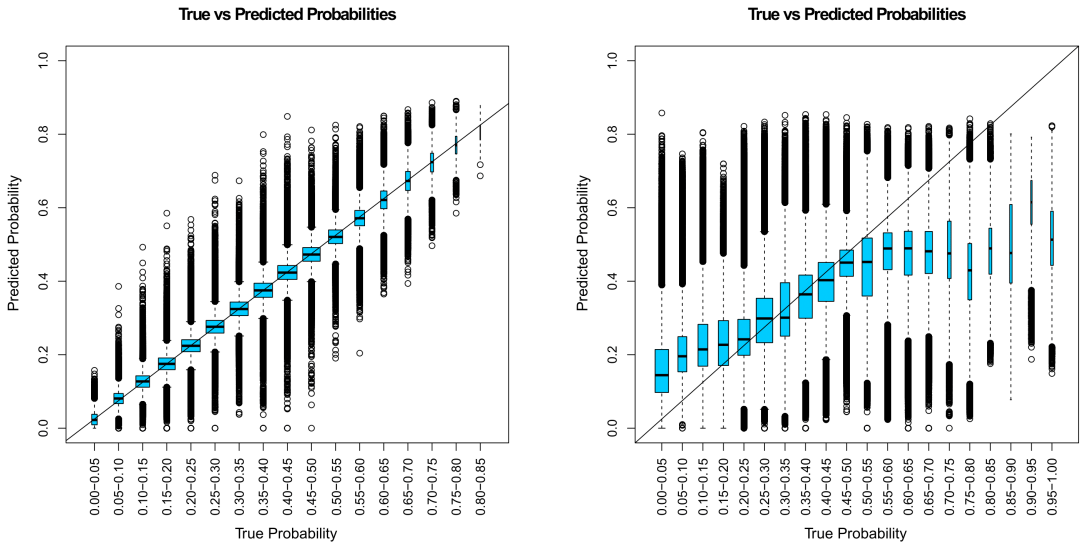


(d) TM RMSE

Figure 3.2: Census data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 1 of the appendix.

probability group fluctuates along the entire range. However, the probabilities are more skewed towards 0. Figure 3.3 shows these behaviors in the probability boxplots for the logistic regression. All of the remaining boxplots for the LGM and TM are presented in the appendix in Figure 3 and Figure 4, respectively. The hyperparameters selected by the

grid search for the adjusted decision tree and adjusted random forest for both probability models are shown in Table 3.2.



(a) LGM (b) TM

Figure 3.3: Iowa recidivism data: true probabilities vs predicted probabilities for the logistic regression based on the LGM and TM. Individual values of all data points from 1,000 replicates are displayed.

Table 3.2: Iowa recidivism data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(10, 5, -1)	(300, 100, 9)
TM	(10, 5, -1)	(1000, 50, 6)

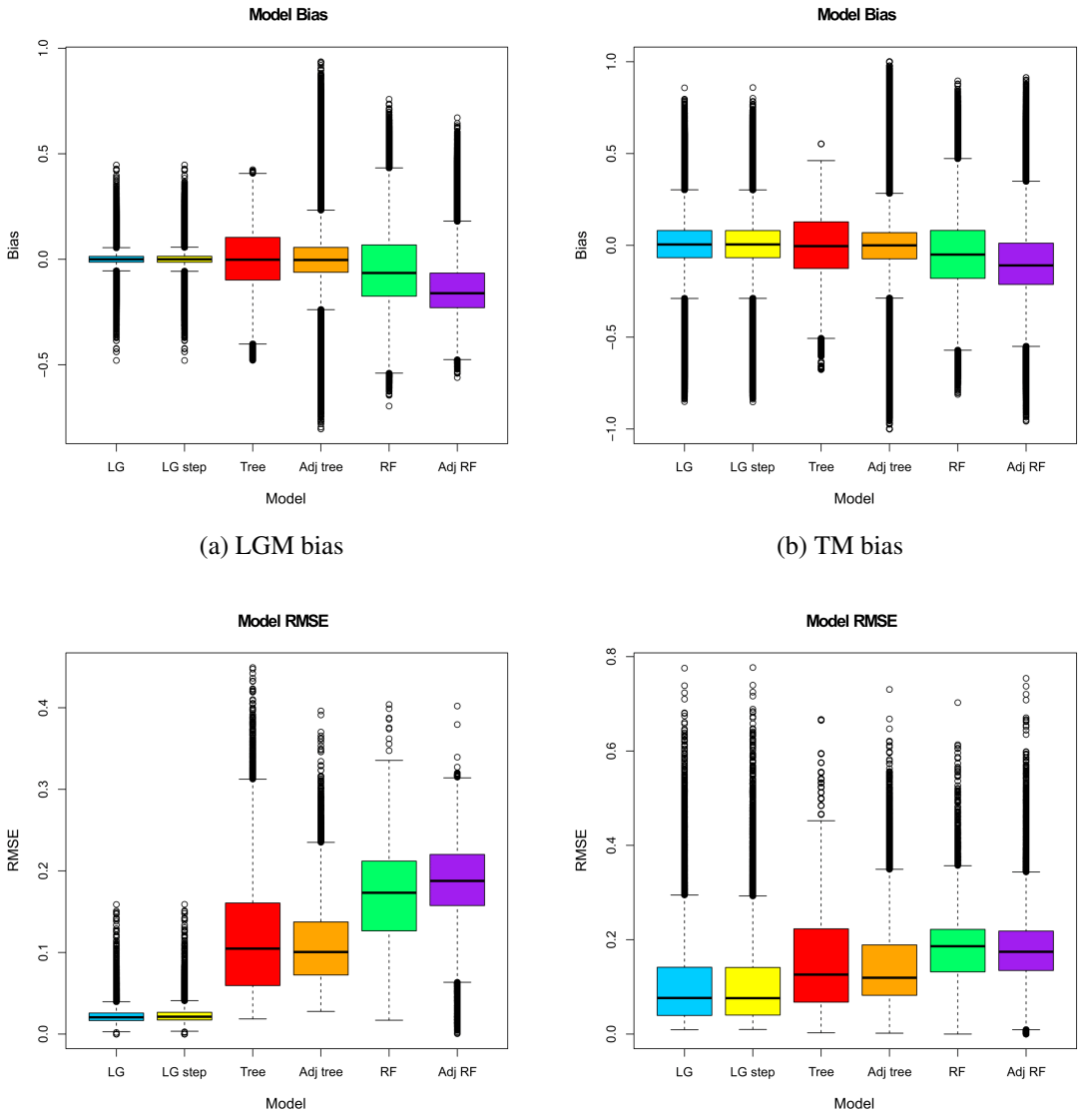
The boxplots in Figure 3.4 show clear differences for bias and RMSE for both probability models. Both logistic regression models perform almost the same. They have a median bias of 0.0040 each and median RMSE of 0.0765 and 0.0761 for the TM. The next best in both scenarios is the adjusted tree with median bias and RMSE of -0.0021 and 0.1193 for the TM. The adjusted decision tree is able to follow the general trend of the true probabilities well. However, there is more variance in the predictions versus the

logistic regression models (see Figure 4 in the appendix for further illustration). For the default decision tree, note that it has a lower median RMSE value than both random forest models for the LGM and TM but it is clear that the model is unable to give correct estimates. It predicts the same probability for all observations in every iteration. The reason why this behavior doesn't reflect in the bias and RMSE graphs is that the prediction is closer to the true probabilities for most of the data points. As mentioned, the middle groups are more populated and the default decision tree's prediction for those middle probability groups is closer to the true probability more often than the predictions of the random forests. For both random forest models, the predicted values for true probabilities within the first six probability groups are pushed to zero. This is why the forests perform poorly. The median bias and RMSE for the default random forest are -0.0598 and 0.1862 for the TM. For the adjusted random forest these values are -0.1124 and 0.1744, which is much higher than the logistic regression models.

3.2.3 HMEQ data

The distribution of the LGM for the HMEQ data set had a mean, median, and standard deviation of 0.8005, 0.9394, and 0.2806. The range was 0 to 0.9996. For the TM, the mean, median, and standard deviation were 0.8005, 0.9668, and 0.3047. The probabilities ranged from 0 to 1. For both of these probability distributions, it is clear that most of the generated probabilities were very close to one. Figure 5 and Figure 6 in the appendix show the true versus predicted probabilities of all machine learning algorithms for the LGM and TM respectively. The probability group sizes are indicated through varying boxplot widths. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.3.

In terms of bias and RMSE, the simple logistic regression outperforms the other five machine learning models for the LGM closely followed by the logistic regression with step function. The median RMSEs are at least five times as high for the tree-based models in this case. For the TM, although the bias of the adjusted decision tree and logistic



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.4: Iowa recidivism data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 2 of the appendix.

regression with step function has almost the exact same absolute values, the quantiles are closer together for the adjusted tree, which is advantageous in terms of predictions. This is well reflected in the RMSE median values, with the adjusted tree having a median of 0.0202 and the logistic regression step model having a median of 0.0390. Interestingly, the

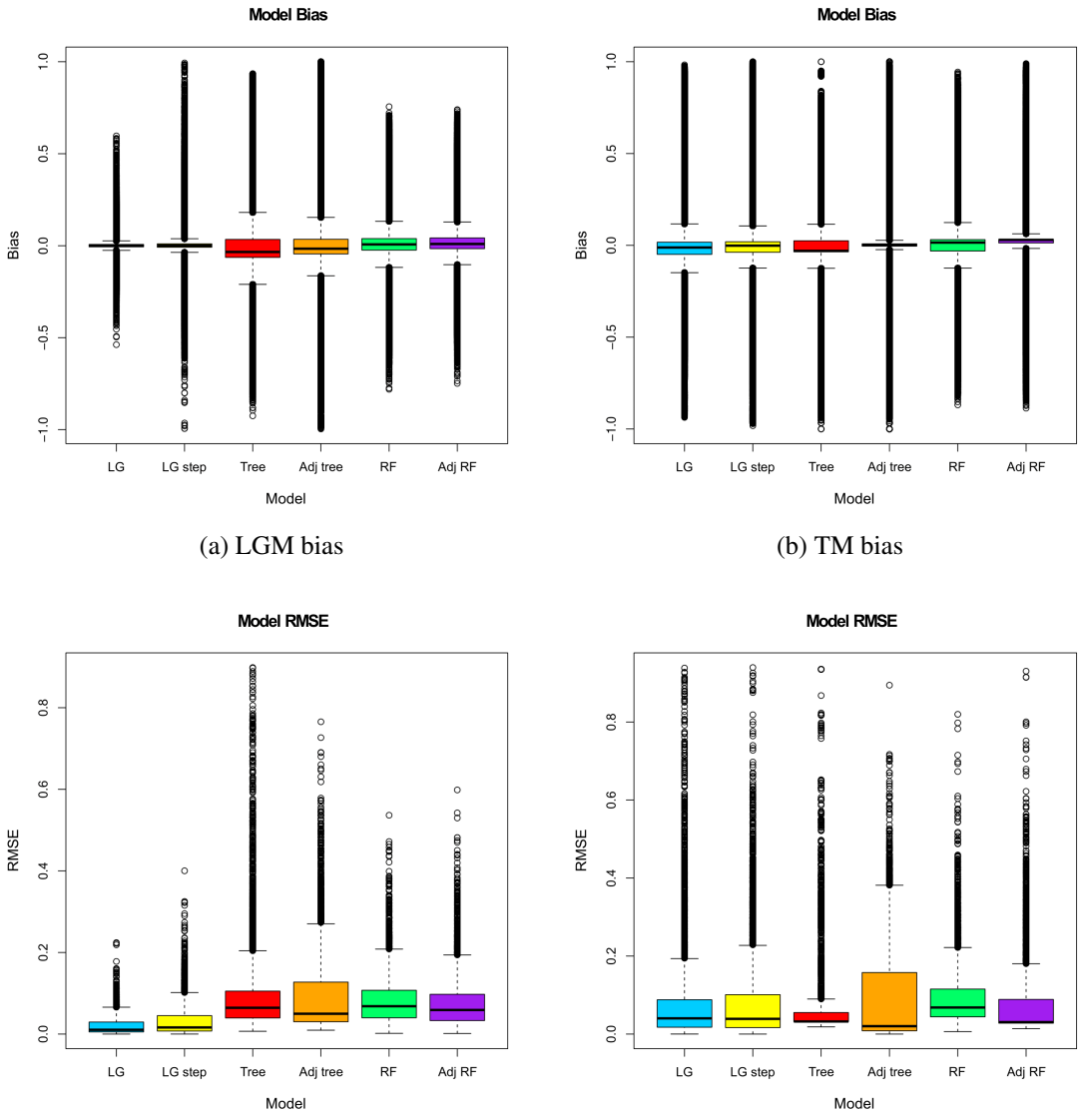
Table 3.3: HMEQ data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(5, 5, -1)	(1000, 1, 3)
TM	(20, 10, -1)	(300, 50, 3)

adjusted random forest has a median bias of 0.0299, which is almost 30 times as high as the absolute median bias values of the two best-performing models. However, in terms of RMSE it comes second in the ranking. This can be explained by the predictive behavior seen in Figure 6 in the appendix. While the median value of the predictions is further away from the true probability for the most populated probability group, the variance of the predictions is very low in comparison to the logistic regression models and default random forest. Note that the adjusted decision tree has the largest range between the 25th and 75th percentile due to the extreme variance of predictions for every observation.

3.2.4 German credit data

For the German credit data set the mean, median, and standard deviation of the LGM were 0.3000, 0.2206, and 0.2480 respectively. The probabilities ranged from 0.0019 to 0.9511. For the TM, these statistics were 0.3000, 0.2458, and 0.2879. The range was 0 to 1. The probability distributions indicate that most of the data is between 0 and 0.2 for both of the models. However, a notable difference is that for the LGM the probabilities are distributed among the groups in a way that the number gradually decreases: 144 probabilities between 0-0.05, 140 between 0.05-0.10, 98 between 0.10-0.15, 85 between 0.15-0.20, to finally one entry in the group 0.95-1. The TM generates probabilities that are not distributed following a trend. For the group between 0-0.05 there are 340 probabilities, for 0.05-0.10 there are 32, for 0.10-0.15 there are 41, for 0.15-0.20 there are 28, and 0.20-0.25 there are 118. Figure 8 in the appendix shows the true versus predicted probabilities of the six algorithms for the TM and indicates the probability group sizes through the



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.5: HMEQ data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 3 of the appendix.

boxplot widths. Figure 7 in the appendix shows the true and predicted probabilities for the LGM. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.4.

The bias plot for the LGM in Figure 3.6 indicates that the adjusted tree outperforms

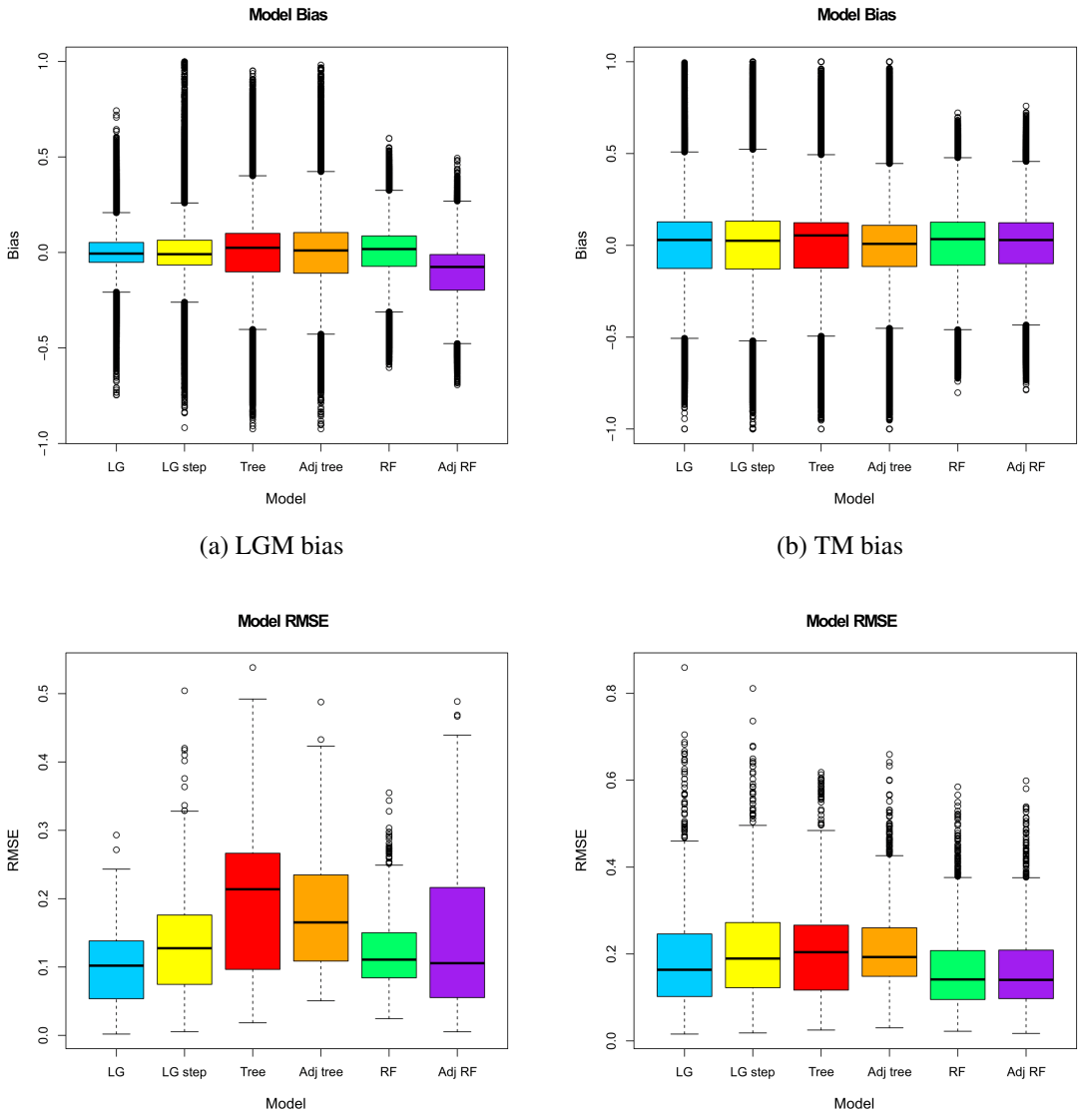
Table 3.4: German credit data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(5, 3, -1)	(1000, 100, 3)
TM	(20, 5, -1)	(1000, 1, 6)

the simple regression model in terms of the median (0.0010 versus -0.0078 for the logistic regression). However, it has a bigger range between the 25th and 75th percentile. Analyzing the RMSE graph paints a different picture. In this case, the simple logistic regression function performs the best with a median value of 0.1020. The adjusted random forest has the next best median value of 0.1056 followed by the default random forest. This is explained by the fact that the median predicted values of the adjusted decision tree are closer to the true probabilities for the first few probability groups that are most heavily populated. Nevertheless, the variance of those predictions is much higher for the adjusted tree than it is for both of the random forests. This is why they have significantly lower RMSEs. For the TM, the difference between the different machine learning models is more difficult to assess as they have similar 25th and 75th percentiles. The lowest median for the bias is 0.0106 for the adjusted tree followed by the adjusted random forest with a median of 0.0258. The lowest RMSE values are those of the random forests with medians of 0.1405 and 0.1414 for the adjusted random forest and default random forest respectively. The next best model is the logistic regression with a median of 0.1637.

3.2.5 Marketing promotion campaign data

For the marketing promotion campaign data, the probabilities of the LGM had mean, median, and standard deviation values of 0.1430, 0.1335, and 0.0592, respectively with a range of 0.0394 to 0.4581. The probabilities of the TM had a mean, median, and standard deviation of 0.1430, 0.1200, and 0.1198. The range was 0 to 0.7500. The true probabilities generated by the LGM were such that there was a general steady decline in the



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.6: German credit data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 4 of the appendix.

number of entries per probability group. There are exceptions for the first two probability groups. The first group with probabilities between 0-0.05 has only 270 entries (out of the 12,800 total observations) and the second has 2,954. The third has 4,013 entries and afterward, there is a steady decrease. For the TM, the entries per group do not follow a

general trend. However, the fluctuations are not as extreme as in the German Credit data. Figure 9 and Figure 10 in the appendix show the true versus predicted probabilities for all the machine learning algorithms for the LGM and TM respectively. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.5.

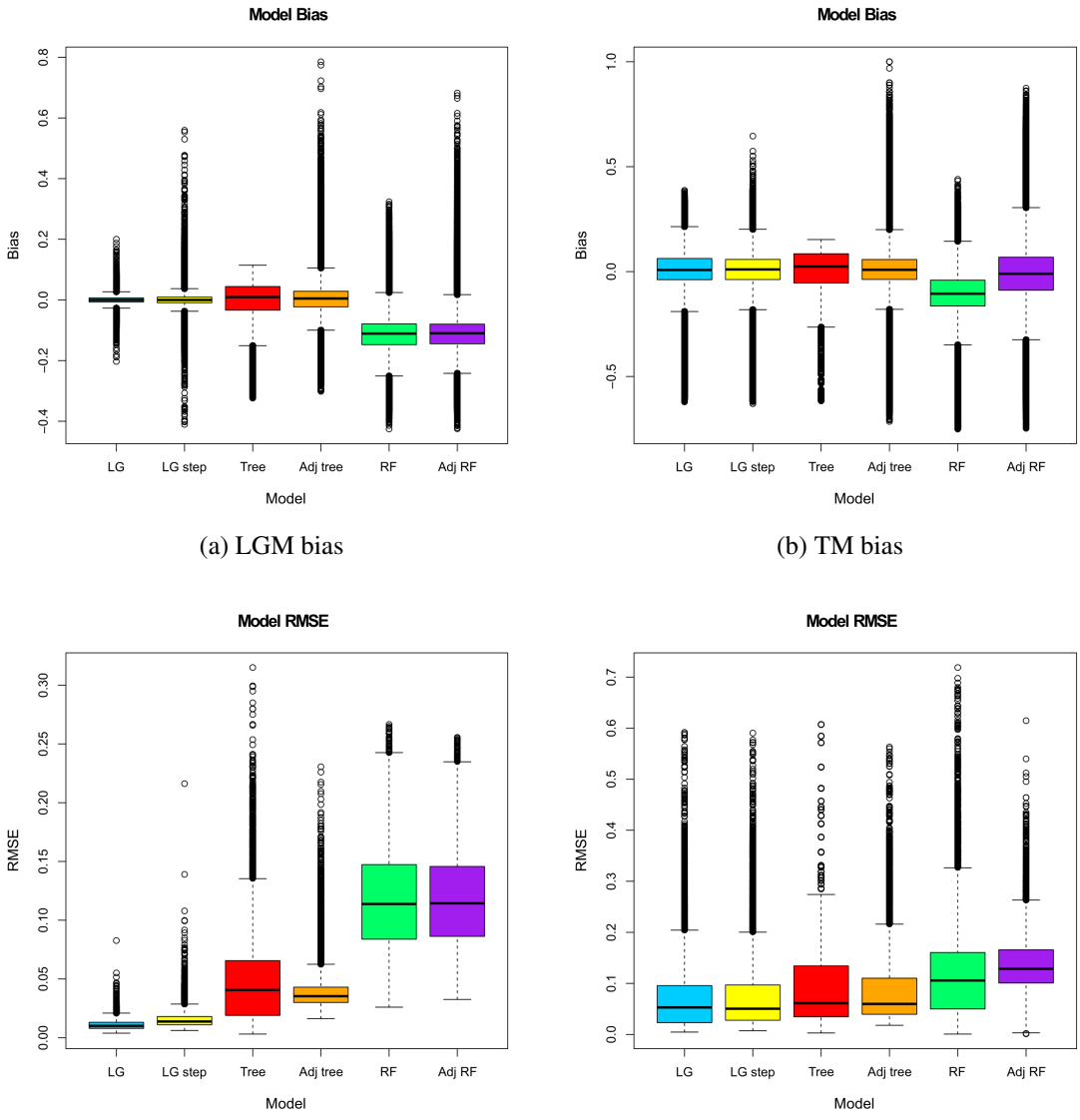
Table 3.5: Marketing promotion campaign data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(20, 3, -1)	(300, 50, 6)
TM	(20, 5, -1)	(1000, 1, 6)

The LGM shows the clear superiority of the logistic regression models in terms of bias and RMSE. The adjusted decision tree is the third-best model. Both random forest models perform the poorest. For the TM, the difference in performance between the logistic regression models and the adjusted decision tree isn't as large as in the LGM. The median biases are not too far from each other. The median values are 0.0068, 0.0097, and 0.0032 respectively for the logistic regression, logistic regression with step function, and adjusted decision tree. The RMSE plot gives a more detailed indication of the performance since predictions that are further away from the true probabilities are penalized more. Here, the lowest median RMSE values are the ones from the logistic regression models with 0.0529 and 0.0504. The adjusted tree is not far behind with 0.0598. Both random forest models perform the worst with median values of 0.1057 and 0.1286 for the default random forest and adjusted random forest respectively.

3.2.6 Bank marketing data

The probabilities generated from the LGM had mean, median, and standard deviation values of 0.1095, 0.0220, and 0.2034, respectively with a range of 0.0002 to 0.9999. The probabilities of the TM had a mean, median, and standard deviation of 0.1095, 0.0005,



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.7: Marketing promotion campaign data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 5 of the appendix.

and 0.2396. The range was 0 to 1. Hence the bulk of the probabilities was very close to zero, especially for the TM. For the LGM, 68.0% of the probabilities were below 0.05 and for the TM this percentage was 74.5%. This is partly the reason why the biases of the different machine learning models are all very close in median values as well as their

25th and 75th percentiles. Figure 11 and Figure 12 in the appendix show the true versus predicted probabilities for all the machine learning algorithms for the LGM and TM respectively. The boxplot widths indicate the true probability group sizes. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.6.

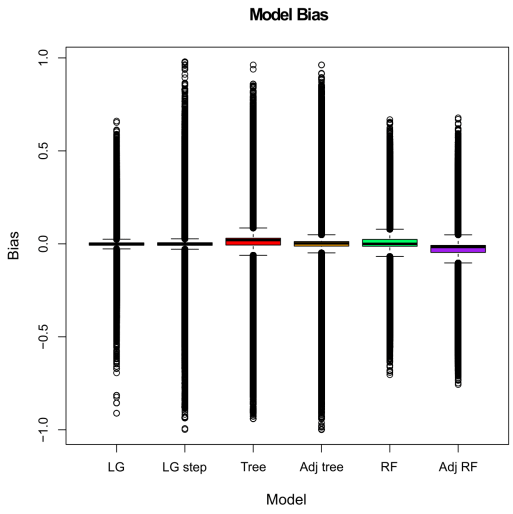
Table 3.6: Bank marketing data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(20, 5, -1)	(500, 100, 3)
TM	(20, 5, -1)	(1000, 1, 9)

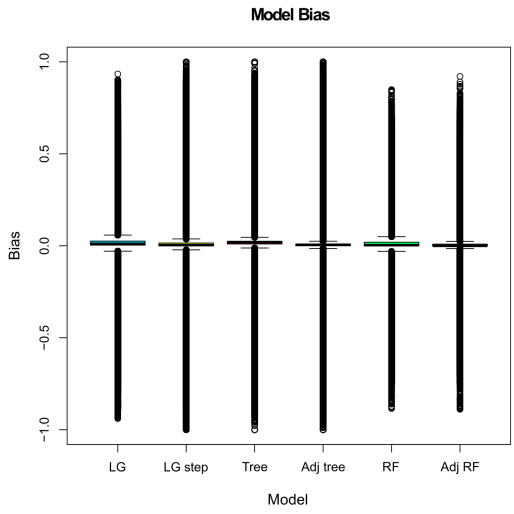
For the LGM, the adjusted tree and random forest have the lowest absolute value of the median bias of approximately 0.0016 (Figure 3.8). The next best bias is that of the logistic regression with step function, which has an absolute median bias of 0.0018. In fact, both logistic regression models, the adjusted tree, and the default random forest have almost the same absolute median bias values. An easier distinction can be made for the RMSE. In this case, the logistic regression models have the lowest median values of 0.0105 and 0.0113 followed by the adjusted tree with a median of 0.0173. For the TM, the adjusted random forest is closest to zero with a median bias of -0.0005 followed by the logistic regression step model with a median of 0.0033. The lowest median RMSE is that of the adjusted tree with 0.0076 followed by the logistic regression with step function with 0.0123.

3.2.7 Churn data

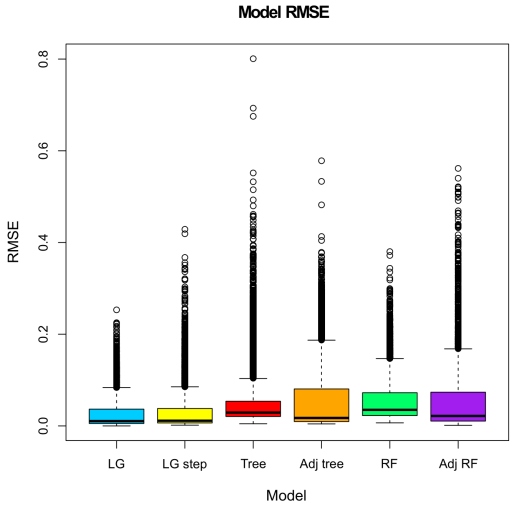
The probabilities generated from the LGM had a mean, median, and standard deviation of 0.2037, 0.1532, and 0.1633, respectively with a range of 0.0114 to 0.9333. The probabilities of the TM had a mean, median, and standard deviation of 0.2037, 0.0712, and 0.2788. The range was 0 to 1. Figure 13 and Figure 14 in the appendix show the



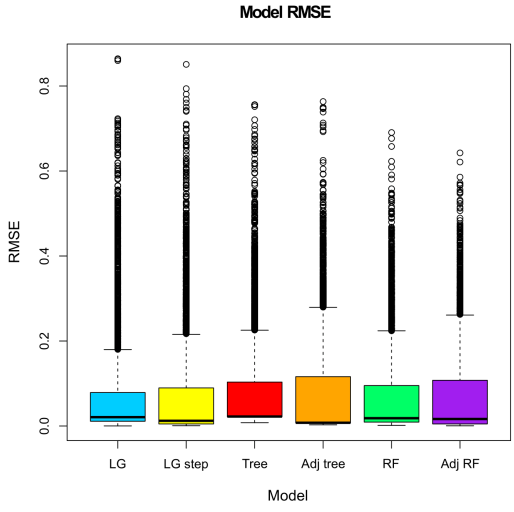
(a) LGM bias



(b) TM bias



(c) LGM RMSE



(d) TM RMSE

Figure 3.8: Bank marketing data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 6 of the appendix.

true versus predicted probabilities for the six algorithms for the LGM and TM respectively. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.7.

Figure 3.9 reveals that for the LGM again a clear distinction can be made between the

Table 3.7: Churn data: hyperparameters found by the grid search for the tree and RF models.

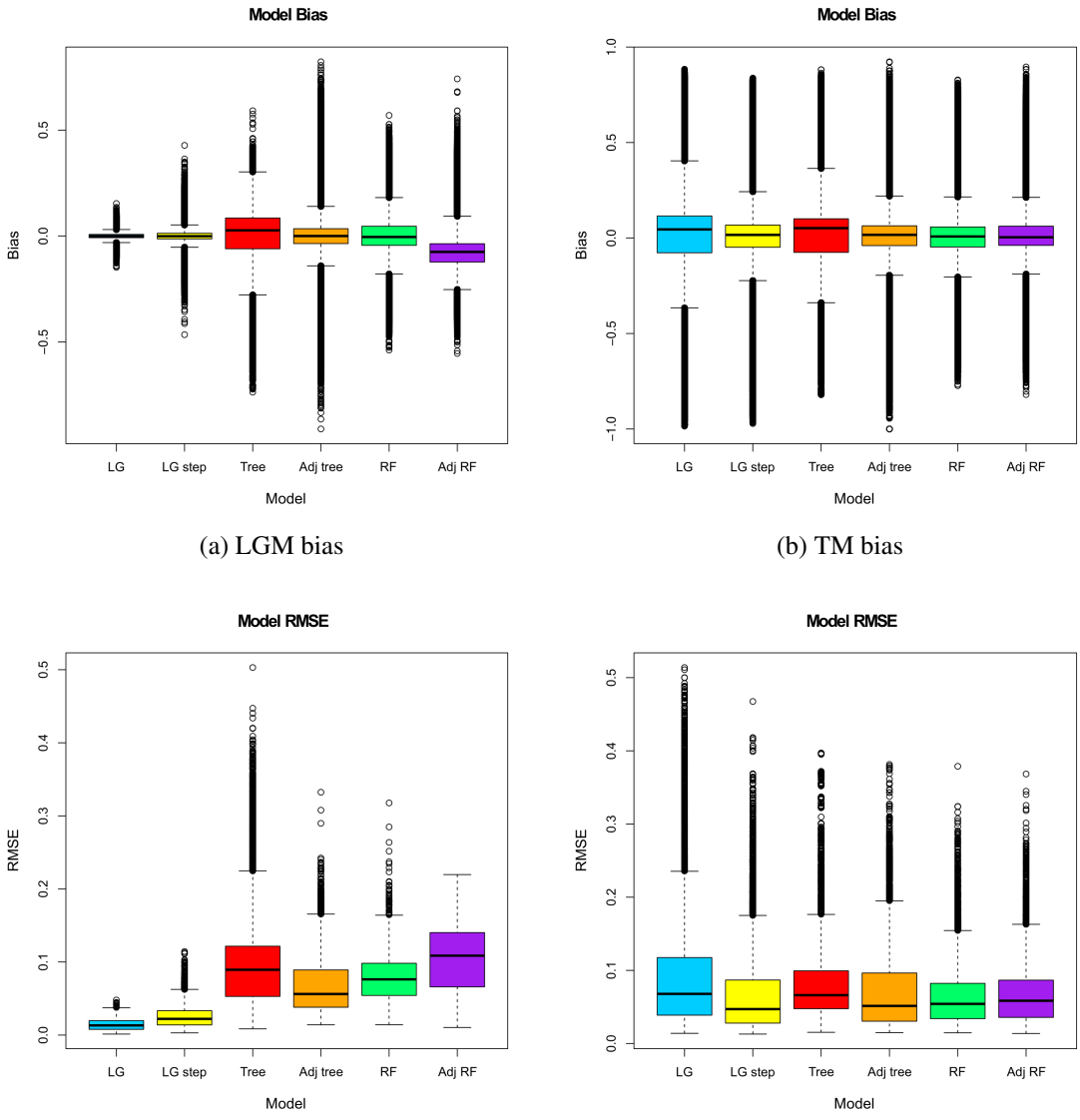
	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(20, 5, -1)	(1000, 50, 6)
TM	(20, 5, -1)	(500, 1, 6)

six different machine learning models. Especially the RMSE plot highlights the superiority of the logistic regression models in predicting probabilities correctly. The simple logistic regression has the lowest absolute median RMSE of 0.0132 followed by the logistic regression with step function with a median of 0.0222. The adjusted tree has the third-lowest RMSE (0.0562) and the adjusted random forest performs the worst with a median of 0.1087.

The contrast for the TM in terms of performance is not as high as for the first probability. However, the RMSEs of both logistic regression models have significantly increased. In terms of bias, the logistic regression with step function, adjusted tree, random forest, and adjusted forest perform very similarly. This is also reflected in the RMSE plot. However, the logistic regression with step function still yields the lowest mean RMSE of 0.0667. This is followed by the adjusted decision tree with 0.0753 and then the random forest with 0.0807 median RMSE.

3.2.8 Internet churn data

For the internet churn data, the probabilities generated from the LGM had a mean, median, and standard deviation of 0.5539, 0.7058, and 0.3721, respectively with a range of 0.0004 to 0.9990. The probabilities of the TM had a mean, median, and standard deviation of 0.5539, 0.8354, and 0.4504. They ranged from 0 to 1. Figure 15 and Figure 16 in the appendix show the true versus predicted probabilities for the six machine learning algorithms for the LGM and TM respectively. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.9: Churn data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 7 of the appendix.

shown in Table 3.8.

As for the previous data set, both logistic regression models are visibly superior to the other machine learning models for the LGM. The bias plot also shows that the majority of the bias of the logistic regression model is much closer to zero than for the other four

Table 3.8: Internet churn data: hyperparameters found by the grid search for the tree and RF models.

	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(10, 5, -1)	(1000, 50, 6)
TM	(20, 10, -1)	(1000, 50, 6)

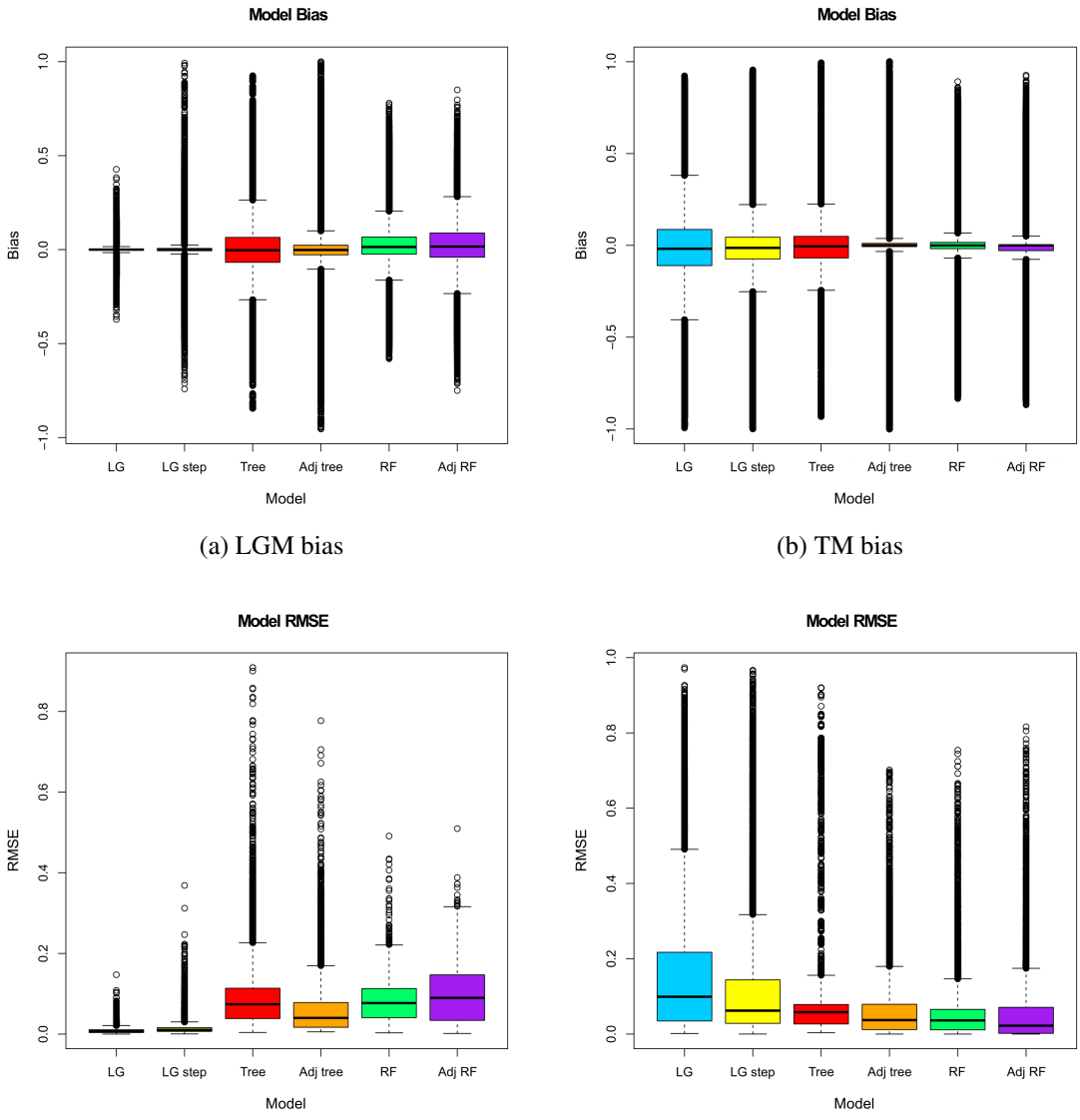
models. In terms of median RMSE, the simple logistic regression has the lowest value of 0.0074. Next is the step logistic regression with 0.0105 followed by the adjusted decision tree with 0.0400.

For the TM, the random forests are the best-performing machine learning models, slightly outperforming the adjusted decision tree. Analyzing Figure 16 in the appendix reveals that the first two and last two probability groups have the largest number of entries. The random forests give the least biased predictions for these extreme probability groups with lower variance than the other models. The median RMSE of the adjusted random forest is 0.0222, that of the default random forest is 0.0363 and that of the adjusted decision tree follows closely with 0.0371. The simple logistic regression model has the highest median RMSE of 0.0991.

3.2.9 Loan data

Finally, the mean, median, and standard deviation of the LGM for the loan data set were 0.1601, 0.1386, and 0.0939 respectively. The probabilities ranged from 0.0129 to 0.9646. For the TM, the mean, median, and standard deviation were 0.1601, 0.1043, and 0.1626. The probabilities ranged from 0 to 1. Figure 17 and Figure 18 in the appendix show the true versus predicted probabilities for the six machine learning algorithms for the LGM and TM respectively. The hyperparameters selected by the grid search for the adjusted decision tree and random forest for both probability models are shown in Table 3.9.

As can be seen in Figure 3.11, both of the logistic regression models perform best



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.10: Internet churn data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 8 of the appendix.

for the LGM according to bias and RMSE. The median RMSE for the simple logistic regression is 0.0138 and that of the logistic regression step model is 0.0229. The third best is the adjusted decision tree with a median RMSE of 0.0421.

For the TM, the performances are more similar for both bias and RMSE. The random

Table 3.9: Loan data: hyperparameters found by the grid search for the tree and RF models.

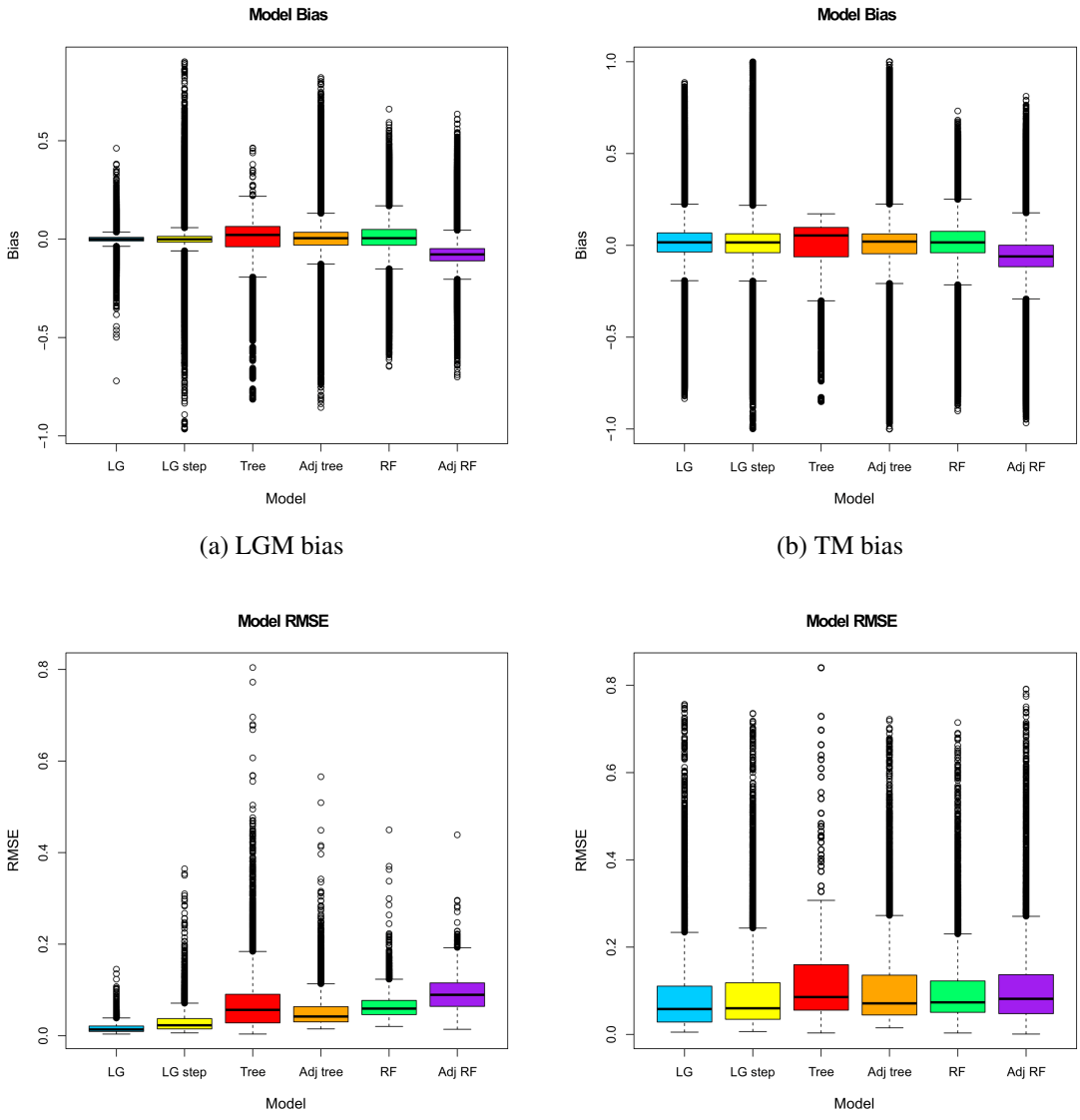
	Tree	RF
Model	(minsplit, maxdepth, cp)	(ntree, nodesize, mtry)
LGM	(20, 3, -1)	(1000, 50, 6)
TM	(10, 5, -1)	(1000, 50, 9)

forest has the lowest absolute median bias of 0.0137, followed closely by both logistic regression models (0.0165 and 0.0147 respectively). In terms of RMSE, however, the logistic regression models do the best. The median RMSE values for the TM are 0.0582 and 0.0599 for the simple and step logistic regression models respectively. The next best model is the adjusted decision tree with a median RMSE of 0.0711.

3.3 Discussion of results

To assess the predictive capabilities of six different machine learning algorithms, we ranked them for both median bias and median RMSE for every data set and for both probability models (LGM and TM). The ranking was done in two different ways. First, the algorithms were ranked for each data set separately according to the absolute median bias and the median RMSE, and then a final rank was obtained by taking the mean of the individual ranks. There was only one tie for the internet churn data between the adjusted decision tree and the adjusted random forest for the median bias. The tie was broken in favor of the algorithm that had a lower absolute mean bias. Table 3.10 shows the rankings of the algorithms in terms of absolute median bias for both probability models. Table 3.11 shows the rankings of the machine learning algorithms in terms of median RMSE for both probability models.

The second set of rankings was based on the median bias and median RMSE values for each algorithm and data set. The final rank was determined according to the average absolute median bias and average median RMSE values. Table 3.12 and Table 3.13 show



(a) LGM bias

(b) TM bias

(c) LGM RMSE

(d) TM RMSE

Figure 3.11: Loan data: boxplots for the bias and RMSE for the 1,000 repetitions of the LGM and TM. Descriptive statistics for the values shown in each boxplot may be found in Table 9 of the appendix.

these rankings. There is a slight change when using the average values versus the individual rankings, except for the RMSE ranking for the LGM. The changes are a result of errors being very close to each other for the algorithms for some of the data sets.

The top three machine learning algorithms for the LGM are the same for both bias

Table 3.10: Rankings of the biases for six different machine learning algorithms on data generated from the LGM and TM. The ranks are calculated for each of the nine data sets. The average rank across the data sets provides a global ranking.

		Data set									
BIAS LGM	Rank	1	2	3	4	5	6	7	8	9	Ave Rank
LG	1	2	2	1	3	2	4	1	1	2	2.00
LG Step	2	1	1	2	4	1	3	2	2	3	2.11
Adj Tree	3	3	4	4	1	3	1	3	3	4	2.89
RF	4	5	5	3	5	5	2	4	5	1	3.89
Tree	5	4	3	6	2	4	6	5	4	5	4.33
Adj RF	6	6	6	5	6	6	5	6	6	6	5.78
BIAS TM	Rank	1	2	3	4	5	6	7	8	9	Ave Rank
Adj Tree	1	3	2	2	1	1	4	4	1	4	2.44
LG Step	2	1	3	1	3	3	2	3	5	2	2.56
RF	3	5	5	4	5	6	3	2	3	1	3.78
Adj RF	4	6	6	5	2	5	1	1	2	6	3.78
LG	5	4	4	3	4	2	5	5	6	3	4.00
Tree	6	2	1	6	6	4	6	6	4	5	4.44

and RMSE in all four tables. The best overall performing algorithm is the simple logistic regression, followed by the logistic regression with step function and the adjusted decision tree. The average ranks and average bias/RMSE values show that there is a notable gap in performance between the logistic regression and tree-based models. The default decision tree ranks among the middle two machine learning algorithms in all of the tables for the LGM. However, it should not be used to predict probabilities since it only gives a few unique probabilities in each iteration and thus struggles to give correct values. This is partly explained by the fact that the tree stops growing once no split improves the relative error by more than 0.01. This leads to a small number of unique probability estimates because of the smaller number of terminal nodes. The default decision tree has better bias and RMSE when the bulk of the true probabilities isn't at extreme ends and the median predictions are closer to the true p around the middle groups of the probability range. It does better than the random forests because it has a much lower variance in predictions. In general, it is likely to give inaccurate estimates as has been noted by authors such as Breiman (1996), Provost and Domingos (2003), and Zadrozny and Elkan (2001) in

Table 3.11: Rankings of the RMSE for six different machine learning algorithms on data generated from the LGM and TM. The ranks are calculated for each of the nine data sets. The average rank across the data sets provides a global ranking.

		Data set									
RMSE LGM	Rank	1	2	3	4	5	6	7	8	9	Ave Rank
LG	1	2	1	1	1	1	1	1	1	1	1.11
LG Step	2	1	2	2	4	2	2	2	2	2	2.11
Adj Tree	3	3	3	3	5	3	3	3	3	3	3.22
Tree	4	4	4	5	6	4	5	5	4	4	4.56
RF	5	5	5	6	3	5	6	4	5	5	4.89
Adj RF	6	6	6	4	2	6	4	6	6	6	5.11
RMSE TM	Rank	1	2	3	4	5	6	7	8	9	Ave Rank
LG Step	1	1	1	4	4	1	2	1	5	2	2.33
Adj Tree	2	4	3	1	5	3	1	2	3	3	2.78
LG	3	2	2	5	3	2	5	6	6	1	3.56
Adj RF	4	6	5	2	1	6	3	4	1	5	3.67
RF	5	5	6	6	2	5	4	3	2	4	4.11
Tree	6	3	4	3	6	4	6	5	4	6	4.56

literature. Lastly and surprisingly, both random forest models are among the three worst-performing machine learning algorithms in all of the tables for the LGM. Intuitively, a more similar performance or improved performance compared to the decision trees would be expected since forests have been proven to give better results than single trees when predicting a class target. It becomes clear from the results that the ability of an algorithm to give good rankings of probabilities versus the ability to give good probability estimates themselves are very different problems.

For the TM, the rankings are not as straightforward. While the adjusted decision tree ranks first, followed by the logistic step model in terms of bias (Table 3.10), it ranks second for the RMSE after the logistic step model (Table 3.11). The RMSE emphasizes larger biases, thus the ranking indicates that the logistic step model has a lower variance in its predictions than the adjusted decision tree. The ranking completely changes when using the average absolute median biases (Table 3.12) and average median RMSEs (Table 3.13). Now, the simple logistic regression ranks second in terms of RMSE with a slightly lower average of 74.56 versus that of the adjusted decision tree with a mean of

Table 3.12: Rankings of the machine learning algorithms based on the mean absolute value of biases (times 1,000) for the LGM and TM.

		Data set									
BIAS LGM	Rank	1	2	3	4	5	6	7	8	9	Ave Bias
LG	1	0	0	1	8	0	2	0	0	0	1.28
LG Step	2	0	0	1	11	0	2	1	0	1	1.83
Adj Tree	3	3	3	15	1	9	2	1	2	3	4.30
Tree	4	9	1	35	7	9	21	21	2	21	14.09
RF	5	36	60	13	16	117	2	10	14	0	29.87
Adj RF	6	71	161	16	82	118	18	80	19	83	71.95
BIAS TM	Rank	1	2	3	4	5	6	7	8	9	Ave Bias
Adj Tree	1	5	2	1	11	3	7	18	0	18	7.11
LG Step	2	0	4	1	27	10	3	17	14	15	10.09
LG	3	8	4	12	31	7	11	45	18	16	16.93
Tree	4	1	0	32	50	23	22	55	10	55	27.59
RF	5	29	60	17	32	108	4	6	2	14	30.10
Adj RF	6	240	112	30	26	33	0	2	0	66	56.65

75.24. The default decision tree is last for the TM for both bias and RMSE when using the pure rankings but performs third to last in the rankings based on average values. As already observed for the LGM, the random forests perform worse than the logistic regression models and adjusted tree in terms of RMSE. They seem to have difficulties in giving unbiased estimates of probabilities that fall within the middle of a given probability range. They do exceptionally well when almost all of the probabilities are either very close to zero or very close to one or both. This indicates that they may perform better in the case where the target variable is an immutable state. In the random forest algorithm, the default setting for the number of observations per terminal node is one for classification problems. This could be part of the issue for random forests to give reliable probability estimates. However, in the simulations, the number of observations of the terminal nodes was altered in the adjusted random forest model in the hyperparameter selection process. Utilizing minimum node sizes of 1, 50, 100, and 500 did not lead to significantly differing probability estimates. Note that the *minsplit* parameter of the decision tree, however, has a larger effect on the probability estimates and small node sizes can lead to worse estimates. In the adjusted decision tree models, for most of the terminal nodes, the number of

Table 3.13: Ranking of the machine learning algorithms based on the mean RMSE (times 1,000) for the LGM and TM.

		Data set									
RMSE LGM	Rank	1	2	3	4	5	6	7	8	9	Ave RMSE
LG	1	44	21	10	102	10	10	13	7	14	25.80
LG Step	2	23	21	16	128	14	11	22	11	23	29.84
Adj Tree	3	68	101	50	165	35	17	56	40	42	63.90
Tree	4	80	105	64	214	41	29	89	74	57	83.63
RF	5	116	173	68	111	114	35	76	77	59	92.16
Adj RF	6	150	188	59	106	114	22	109	90	89	102.91
RMSE TM	Rank	1	2	3	4	5	6	7	8	9	Ave RMSE
LG Step	1	34	76	39	190	50	12	67	62	60	65.55
LG	2	52	76	40	164	53	21	107	99	58	74.56
Adj Tree	3	94	119	20	193	60	8	75	37	71	75.24
Tree	4	76	126	32	204	61	22	104	58	86	85.49
RF	5	112	186	68	141	106	18	81	36	73	91.40
Adj RF	6	250	174	31	140	129	17	89	22	82	103.76

observations was much larger than the $minsplit$ value because of the $maxdepth$ parameter.

While the logistic regression models were consistent probability estimators for the LGM, the decision trees and random forests did not seem to be consistent probability estimators for the TM. Even the adjusted decision tree with the same parameter settings as the tree used to generate probabilities in the TM did not yield performances that seem compatible with consistency (e.g. TM for the internet churn data). Using the average bias and RMSE medians further illustrates that for the LGM, the difference in performance between the two logistic regressions and the adjusted tree is much larger than the difference between these algorithms for the TM. For the LGM, the average median RMSE of the adjusted tree is twice as large as that of the logistic regression step model, which is the second-best performing algorithm. The difference between the simple logistic regression (second-best) and the adjusted tree (third-best) for the TM is low. However, in comparison to the logistic regression step model which is the first best algorithm, the RMSE of the adjusted tree is about 15% larger.

Some of the main shortcomings of the logistic regression that have been largely criticized are model misspecification and dealing with missing values, as discussed for ex-

ample by Zhao et al. (2016). In our simulations, the missing values were either kept as their own category for categorical values, the rows were removed if there were not too many of them (less than 1% of observations), or the missing values were replaced by the mean values of the variable and corresponding indicator variables added to the data set. The results revealed that dealing with the missing values in this way can still lead to both logistic regression models giving better probability estimates than the trees. Regarding model misspecification, the results revealed that the simple logistic regression, in terms of average RMSE values for the TM, is slightly better than the adjusted decision tree. This indicates that a simple regression model still has a better chance of outperforming a decision tree and random forest at probability estimation and as discussed in the previous paragraph, the difference in performance for the LGM is extreme. Naturally, there are specific distributions that will make a simple logistic regression fail in giving the correct probability estimates. An example of this is data distributed uniformly in a square where the Euclidean distances to the center of the square indicate the true probabilities (Mease et al., 2007). In this case, a simple interaction term between the two variables is necessary for the logistic regression to become a consistent estimator. Adding this interaction term makes the logistic regression outperform the tree-based models (based on a simple experiment).

A further takeaway from the experiments is the stark contrast between an algorithm's ability to predict the correct target class versus the ability to predict the correct probability. For example, when doing the hyperparameter search for the tree models for the internet churn data set, the AUCs for the TM went as high as 0.97. However, the probability graphs in the appendix reveal that except for the two extreme probability groups closest to zero and one, the algorithms are unable to give satisfactory probability estimates. Consequently, the AUC or other ranking-based assessment techniques are inapplicable to the assessment of probability estimates. Another idea that was examined was whether the imbalance of the data set affected the model performance. The results did not yield any indications of more imbalanced data sets being easier for a specific type of algorithm. Lastly, it was analyzed whether the number of observations in the data set and variables

had an impact on the algorithm's predictive abilities. For the three data sets with the smallest number of observations (HMEQ, German credit, and the bank marketing data set), either the adjusted decision tree or the adjusted random forest had the lowest median RMSE values. These three data sets also had at least twenty variables, whereas the others had nine to fourteen. This observation is not valid for the fourth data set for which the adjusted random forest ranks first (internet churn). A more detailed investigation would be necessary to see if this observation could be made into a general statement.

Conclusion

Decision trees and random forests are being increasingly suggested for binary classification problems due to certain advantages over parametric models, such as a logistic regression. The main advantages include the avoidance of model misspecification, the automatic handling of missing variables, and the ability to deal with small sample sizes, high-dimensional feature spaces, and complex data structures (Scornet et al., 2015). Decision trees have proven to perform extremely well and robustly in predicting the correct class label in classification problems and random forests often allow to further improve this performance. Although the tree-based algorithms perform very well for the actual classification, this is not the case for the probability estimates. While in some cases, the target variable describes an immutable state, there are situations where the event of interest may or may not happen. There is a notion of risk in these scenarios and the true probabilities of the target variable range between zero and one. In such cases, consistency would be a welcome property of the chosen probability estimator.

While some consistency results exist for decision trees and random forests as both class target predictors and probability estimators, they are based on highly simplified versions of the tree and forest building mechanism and thus cannot be translated to the methods employed in practice, such as the *rpart* and *randomForest* packages in R. As described in Chapter 1, utilizing consistent probability estimators is crucial in certain cases as uncontrolled bias, for instance, could have dire consequences. Examples include the direct weighting of survey samples using propensity scores, cost-sensitive decisions, and uplift modeling with the two-model approach. The consistency results for regression by

Scornet et al. (2015) and Klusowski (2021) come closest to the algorithm used in practice. However, these algorithms are continuous target predictors of regression problems.

In order to evaluate the performance of the probability estimates of tree-based models, a Monte Carlo simulation was performed for nine data sets including real data and simulated data. Two probability models were used to generate the true probability of events that drove the generation of the binary response variable. The LGM was a logistic regression-based model, whereas the TM was a tree-based model which yielded a data structure for which logistic regression would be misspecified, and hence inconsistent. Using the generated probabilities from the two models, six different machine learning algorithms were trained on the data sets. These were two logistic regression models, two decision trees, and two random forests. The probability estimates of the six machine learning algorithms were then compared for both probability models using the bias and RMSE as performance measures.

The results of the simulations showed that the difference in performance for the LGM between the logistic regressions and tree-based algorithms was substantial. The results for the TM did not show a difference as clear as for the LGM, since all algorithms had a harder time predicting the correct probabilities. Most importantly, the simulations revealed that the logistic regression models that are consistent estimators for the LGM generally perform better than the tree-based models for the TM. The tree-based methods, although giving good estimates for the LGM in some cases, do not seem to be consistent probability estimators for either probability model and are more variant in their predictions. An important takeaway from the TM is that on average, a logistic regression performs better than the tree-based models in terms of RMSE for probability estimation although the true p is based on a tree structure, i.e. even if it is misspecified.

In practice, the true probability distribution is unknown. Thus it is advisable to use a machine learning model that performs well in different types of scenarios. Based on the findings of our simulations, decision trees cannot be recommended for probability estimation and are inferior to the logistic regression model for probability estimation in realistic data scenarios.

The results of the simulations encourage further studies of the topic. Additional data sets could be considered to compare the probability estimates using other types of generative models and performance measures. Moreover, one could analyze how changing the terminal node size in a classification tree impacts the probability estimates in an empirical setting. One could also look into adding interaction terms of higher orders for the logistic regression step model. This could possibly make the difference in performance even higher for the TM. Another possible study could be using the regression setting in the R *randomForest* package and examining how much this could improve the probability estimates of single trees or random forests, expanding on the work of Malley et al. (2012). It would be interesting to assess these ideas in more extensive empirical studies and to examine whether there is a large difference between data sets describing immutable states versus the occurrence of an event.

Bibliography

Austin, P. C. (2011). An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, 46(3):399–424.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139.

Biau, G. (2012). Analysis of a random forests model. *The Journal of Machine Learning Research*, 13:1063–1095.

Biau, G., Devroye, L., and Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9:2015–2033.

Blattberg, R. C., Kim, B.-D., and Neslin, S. A. (2008). Why database marketing? In *Database marketing*, pages 13–46. Springer.

Breiman, L. (1996). Out-of-bag estimation. Technical report, University of California at Berkeley.

Breiman, L. (2000). Some infinity theory for predictor ensembles. Technical report, University of California at Berkeley.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Breiman, L. (2004). Consistency for a simple model of random forests. statistical department. Technical report, University of California at Berkeley.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Chapman & Hall/CRC, Boca Raton, Florida.

Buskirk, T. D. and Kolenikov, S. (2015). Finding respondents in the forest: A comparison of logistic regression and random forest models for response propensity weighting and stratification. *Survey Methods: Insights from the Field, Weighting: Practical Issues and 'How to' Approach.* Retrieved from https://surveyinsights.org/?p=5108.

Casella, G. and Berger, R. L. (2002). *Statistical inference*, volume 2. Duxbury. Thomson Learning, Pacific Grove, California.

Chawla, N. V. and Cieslak, D. A. (2006). Evaluating probability estimates from decision trees. *American Association for Artificial Intelligence*.

Chickering, D. M. and Heckerman, D. (2000). A decision theoretic approach to targeted advertising. In *Uncertainty in Artificial Intelligence Proceedings*.

Denil, M., Matheson, D., and Freitas, N. (2013). Consistency of online random forests. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1256–1264. PMLR, PMLR.

Devroye, L., Györfi, L., and Lugosi, G. (1996). *A probabilistic theory of pattern recognition*, volume 31. Springer, New York, New York.

Dua, D. and Graff, C. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Flach, P. A. (2016). Roc analysis. In *Encyclopedia of machine learning and data mining*, pages 1–8. Springer.

Gelein, B., Haziza, D., and Causeur, D. (2018). Propensity weighting for survey nonresponse through machine learning. In *13es Journées de méthodologie statistique de l'Insee (JMS)*, Paris, France. hal-02076739.

Gubela, R. M., Lessmann, S., and Jaroszewicz, S. (2020). Response transformation and profit decomposition for revenue uplift modeling. *European Journal of Operational Research*, 283(2):647–661.

Gutierrez, P. and Gérardy, J.-Y. (2017). Causal inference and uplift modelling: A review of the literature. In *Proceedings of The 3rd International Conference on Predictive Applications and APIs*, volume 67, pages 1–13. PMLR.

Klusowski, J. M. (2021). Universal consistency of decision trees in high dimensions. *arXiv preprint arXiv:2104.13881*.

Lin, Y. and Jeon, Y. (2002). Random forests and adaptive nearest neighbors. Technical report, University of Wisconsin.

Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G., and Ziegler, A. (2012). Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods of information in medicine*, 51(1):74–81.

Margineantu, D. D. and Dietterich, T. G. (2003). *Improved class probability estimates from decision tree models*, pages 173–188. Springer, New York, New York.

Mease, D., Wyner, A. J., and Buja, A. (2007). Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, 8:409–439.

Moro, S., Cortez, P., and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

Provost, F. and Domingos, P. (2000). Well-trained PETs: Improving probability estimation trees. *CeDER Working Paper IS-00-04, Stern School of Business, New York University*.

Provost, F. and Domingos, P. (2003). Tree induction for probability-based ranking. *Machine learning*, 52(3):199–215.

Radcliffe, N. J. and Surry, P. D. (2011). Real-world uplift modelling with significance-based uplift trees. *Stochastic Solutions*, pages 1–33.

Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

Rzepakowski, P. and Jaroszewicz, S. (2012). Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327.

SAS Institute Inc (2015). *SAS/STAT® 14.1 User's Guide*. SAS Institute Inc, Cary, NC.

Scornet, E., Biau, G., and Vert, J.-P. (2015). Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741.

Setoguchi, S., Schneeweiss, S., Brookhart, M. A., Glynn, R. J., and Cook, E. F. (2008). Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and drug safety*, 17(6):546–555.

Sołtys, M., Jaroszewicz, S., and i, P. (2015). Ensemble methods for uplift modeling. *Data mining and knowledge discovery*, 29(6):1531–1559.

Therneau, T., Atkinson, B., Ripley, B., and Ripley, B. (2015). Package 'rpart'. `cran.ma.ic.ac.uk/web/packages/rpart/rpart.pdf`. Accessed: 2022-04-01.

Westreich, D., Lessler, J., and Funk, M. J. (2010). Propensity score estimation: machine learning and classification methods as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8):826–833.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 204–213.

Zaniewicz, Ł. and Jaroszewicz, S. (2013). Support vector machines for uplift modeling. In *2013 IEEE 13th International Conference on Data Mining Workshops*, pages 131–138. IEEE.

Zhao, P., Su, X., Ge, T., and Fan, J. (2016). Propensity score and proximity matching using random forest. *Contemporary clinical trials*, 47:85–92.

Appendix

Bias and RMSE tables

The following tables show summary statistics of the bias and RMSE values of the six machine learning algorithms for every data set and both probability models (LGM and TM). The true values were multiplied by 1,000.

Table 1: Census data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-732	-732	-431	-743	-661	-518
MAX	717	717	472	647	567	480
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	33	55
MEDIAN	≈ 0	≈ 0	-9	-3	36	71
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-759	-759	-419	-747	-603	-497
MAX	825	825	472	818	526	596
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	26	134
MEDIAN	-8	≈ 0	1	5	29	240
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	18	13	18	36	57	47
MAX	514	511	258	203	273	263
MEAN	47	27	85	75	121	155
MEDIAN	44	23	80	68	116	150
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	19	20	13	32	55	77
MAX	344	184	278	233	297	439
MEAN	57	37	79	96	119	245
MEDIAN	52	34	76	94	112	250

Table 2: Iowa recidivism data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-479	-479	-479	-804	-695	-561
MAX	447	447	425	936	758	671
MEAN	≈ 0	≈ 0	≈ 0	1	-46	-130
MEDIAN	≈ 0	≈ 0	-1	-3	-60	-161
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-851	-853	-676	-1000	-812	-958
MAX	858	859	552	1000	896	914
MEAN	≈ 0	≈ 0	≈ 0	1	-40	-91
MEDIAN	4	4	≈ 0	-2	-60	-112
RMSE R1	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	≈ 0	19	28	17	≈ 0
MAX	159	159	449	396	404	402
MEAN	23	24	116	109	170	187
MEDIAN	21	21	105	101	173	188
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	9	10	3	2	0	0
MAX	776	777	667	730	703	754
MEAN	106	106	156	146	179	176
MEDIAN	76	76	126	119	186	174

Table 3: HMEQ data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-538	-994	-925	-996	-780	-748
MAX	597	994	935	1000	756	740
MEAN	≈ 0	≈ 0	≈ 0	1	3	11
MEDIAN	1	1	-35	-15	13	16
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-937	-982	-1000	-1000	-869	-887
MAX	983	1000	1000	1000	944	990
MEAN	≈ 0	≈ 0	1	1	4	30
MEDIAN	-12	-1	-32	1	17	30
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	≈ 0	7	9	2	1
MAX	224	400	899	765	536	598
MEAN	20	31	98	92	82	74
MEDIAN	10	16	64	50	68	59
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	≈ 0	18	0	6	14
MAX	939	940	936	895	820	931
MEAN	88	87	87	92	96	80
MEDIAN	40	39	32	20	68	31

Table 4: German credit data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-746	-917	-924	-924	-603	-693
MAX	742	998	951	981	598	494
MEAN	1	2	1	≈ 0	2	-109
MEDIAN	-8	-11	7	1	16	-82
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-1000	-1000	-1000	-1000	-802	-787
MAX	996	1000	1000	1000	722	759
MEAN	≈ 0	1	-1	-1	1	4
MEDIAN	31	27	50	11	32	26
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	2	5	18	51	24	5
MAX	293	504	538	488	355	489
MEAN	99	129	190	177	123	143
MEDIAN	102	128	214	165	111	106
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	16	18	25	30	22	17
MAX	859	811	618	659	585	599
MEAN	193	212	215	215	171	169
MEDIAN	164	190	204	193	141	140

Table 5: Marketing promotion campaign data: Biases and RMSEs (times 1,000) for the LGM and TM

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-203	-411	-324	-301	-426	-425
MAX	200	559	115	785	324	682
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	-115	-115
MEDIAN	≈ 0	≈ 0	9	9	-117	-118
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-621	-629	-616	-714	-750	-747
MAX	387	646	153	1000	440	874
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	-114	1
MEDIAN	7	10	23	3	-108	-33
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	4	6	3	16	26	33
MAX	83	216	315	231	267	256
MEAN	11	16	47	40	119	118
MEDIAN	10	14	41	35	114	114
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	5	7	3	18	1	1
MAX	592	590	607	563	719	615
MEAN	76	77	88	87	124	136
MEDIAN	53	50	61	60	106	129

Table 6: Bank marketing data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-911	-999	-941	-1000	-704	-757
MAX	660	979	962	962	668	678
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	5	-40
MEDIAN	-2	-2	21	2	-2	-18
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-939	-1000	-1000	-1000	-885	-889
MAX	934	1000	1000	1000	849	921
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	3	6
MEDIAN	11	3	22	7	4	≈ 0
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	2	5	4	7	1
MAX	253	429	801	578	380	562
MEAN	30	37	64	62	58	57
MEDIAN	10	11	29	17	35	22
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	1	8	3	1	≈ 0
MAX	864	851	756	763	691	643
MEAN	79	77	88	81	72	71
MEDIAN	21	12	22	8	18	17

Table 7: Churn data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-148	-466	-737	-911	-538	-554
MAX	153	428	592	823	569	742
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	2	-72
MEDIAN	≈ 0	-1	21	-1	-10	-80
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-985	-972	-821	-1000	-773	-820
MAX	884	838	882	923	828	896
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	1	8
MEDIAN	45	17	55	18	6	2
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	1	3	8	14	14	10
MAX	48	114	503	332	318	220
MEAN	14	25	99	67	78	104
MEDIAN	13	22	89	56	76	109
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	2	≈ 0	5	4	3	1
MAX	980	889	751	721	716	695
MEAN	162	107	137	113	102	106
MEDIAN	107	67	104	75	81	89

Table 8: Internet churn data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

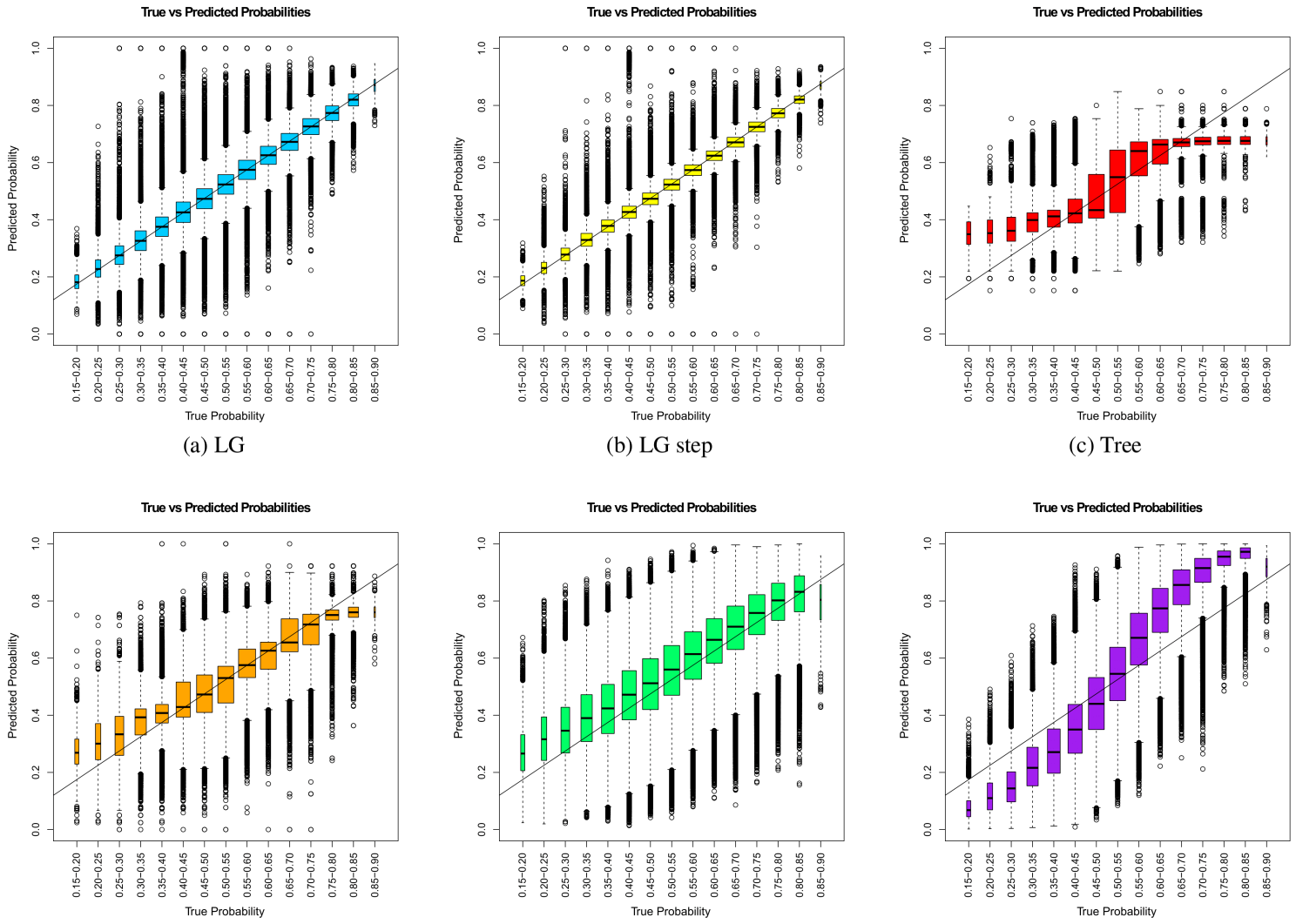
BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-372	-742	-845	-955	-581	-750
MAX	427	992	926	1000	779	850
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	17	17
MEDIAN	≈ 0	≈ 0	-2	-2	14	19
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-995	-1000	-932	-1000	-835	-869
MAX	923	956	994	1000	892	927
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	-7
MEDIAN	-18	-14	-10	0	-2	0
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	≈ 0	≈ 0	4	6	3	2
MAX	147	369	909	777	491	510
MEAN	8	14	91	57	79	92
MEDIAN	7	11	74	40	77	90
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	1	≈ 0	4	0	0	0
MAX	974	967	920	702	754	816
MEAN	163	128	89	64	55	54
MEDIAN	99	62	58	37	36	22

Table 9: Loan data: bias and RMSE (times 1,000) based on 1,000 replicates of the LGM and TM.

BIAS LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-721	-965	-814	-856	-647	-701
MAX	462	901	464	821	661	636
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	12	-76
MEDIAN	≈ 0	-1	21	3	≈ 0	-83
BIAS TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	-835	-1000	-851	-1000	-902	-967
MAX	887	1000	172	1000	732	812
MEAN	≈ 0	≈ 0	≈ 0	≈ 0	10	-69
MEDIAN	16	15	55	18	14	-66
RMSE LGM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	4	6	4	15	20	14
MAX	145	365	804	566	450	439
MEAN	17	31	69	54	65	91
MEDIAN	14	23	57	42	59	89
RMSE TM	LG	LG step	Tree	Adj tree	RF	Adj RF
MIN	5	7	3	15	3	1
MAX	756	736	841	722	715	792
MEAN	94	99	115	110	105	113
MEDIAN	58	60	86	71	73	82

Boxplots of the true probability vs the predicted probability

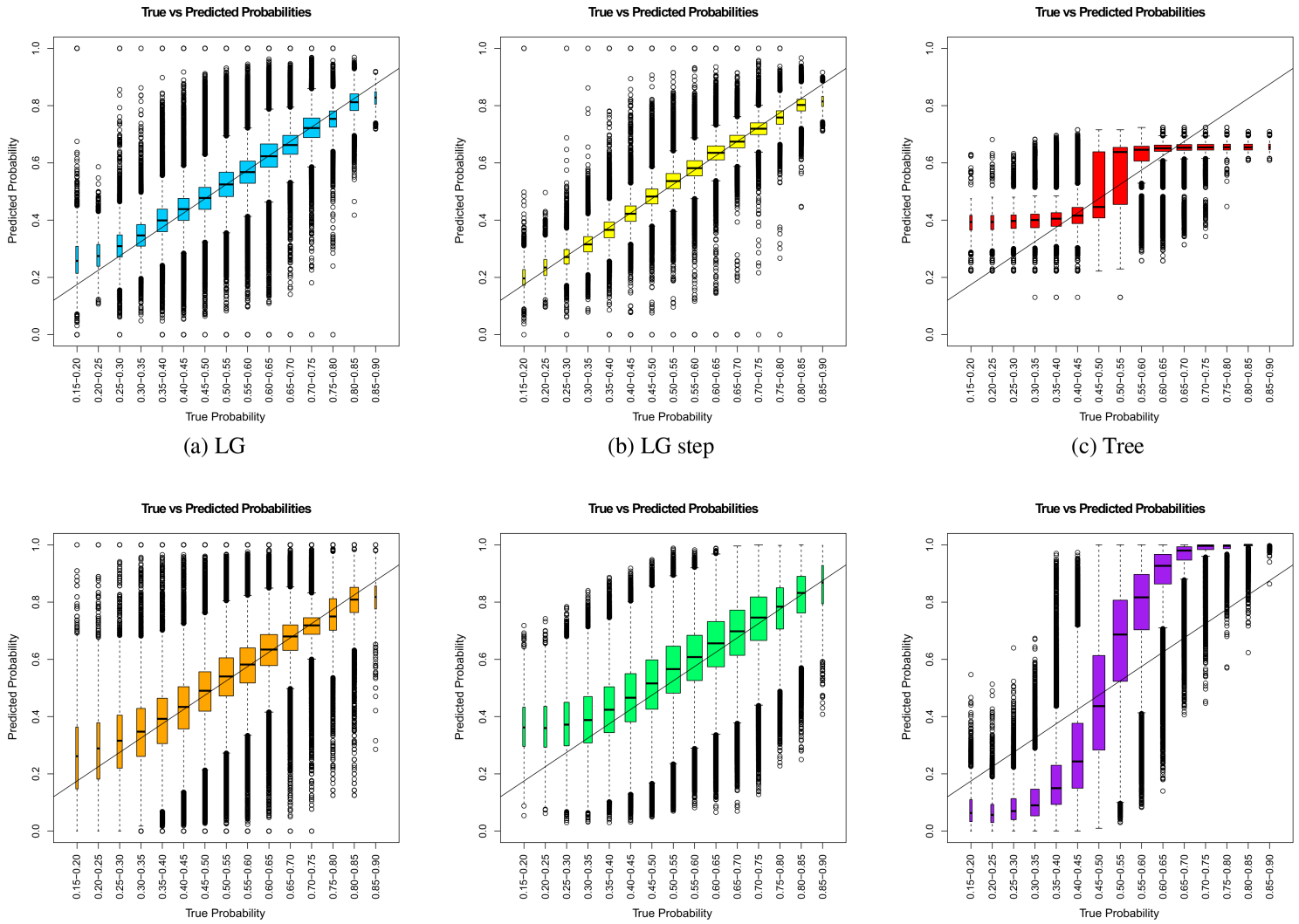
The following graphs are boxplots of the true probabilities versus the predicted probabilities of each machine learning algorithm and probability model for the nine data sets. Individual values of all data poitns from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

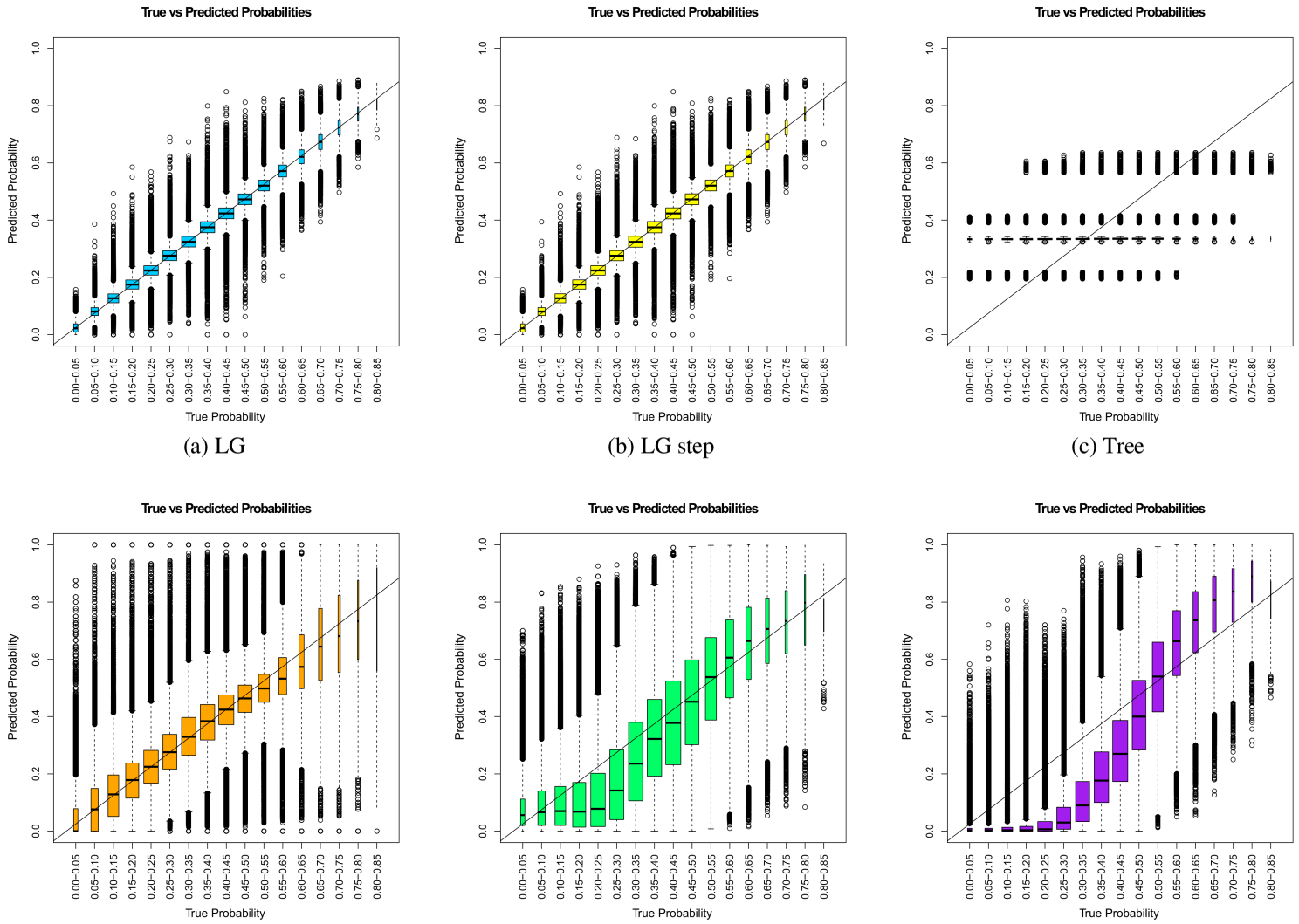
Figure 1: Census data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

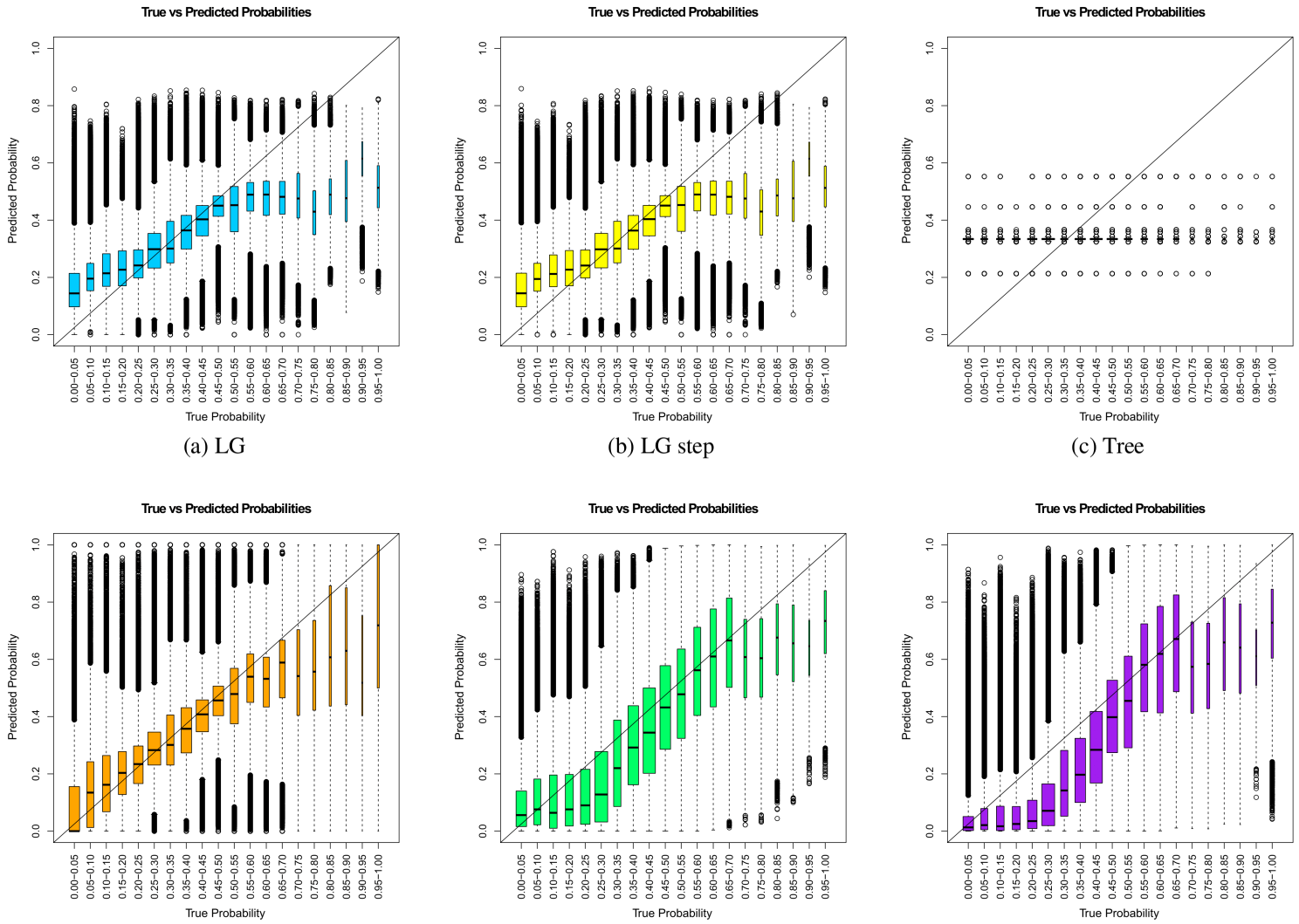
Figure 2: Census data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

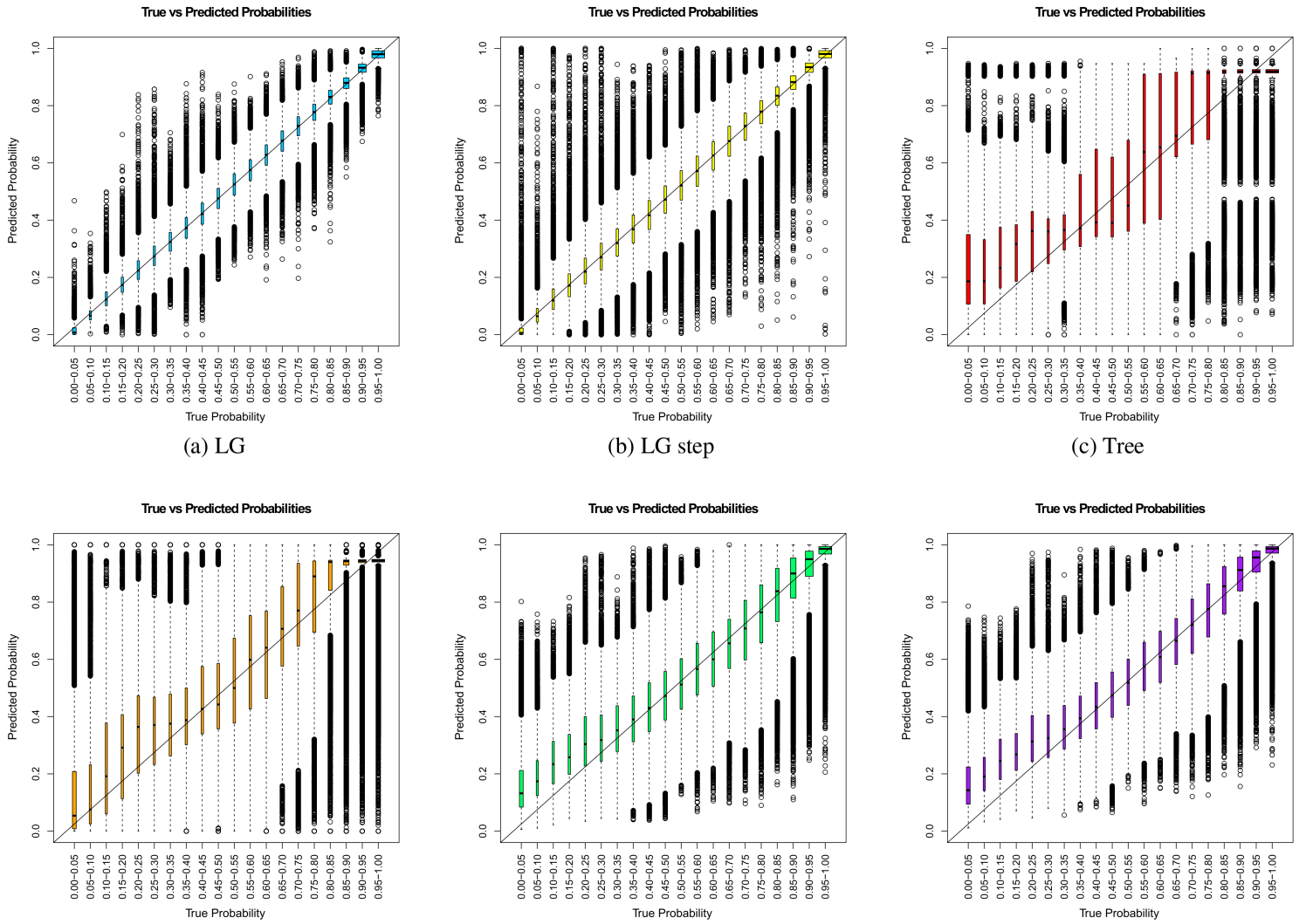
Figure 3: Iowa recidivism data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

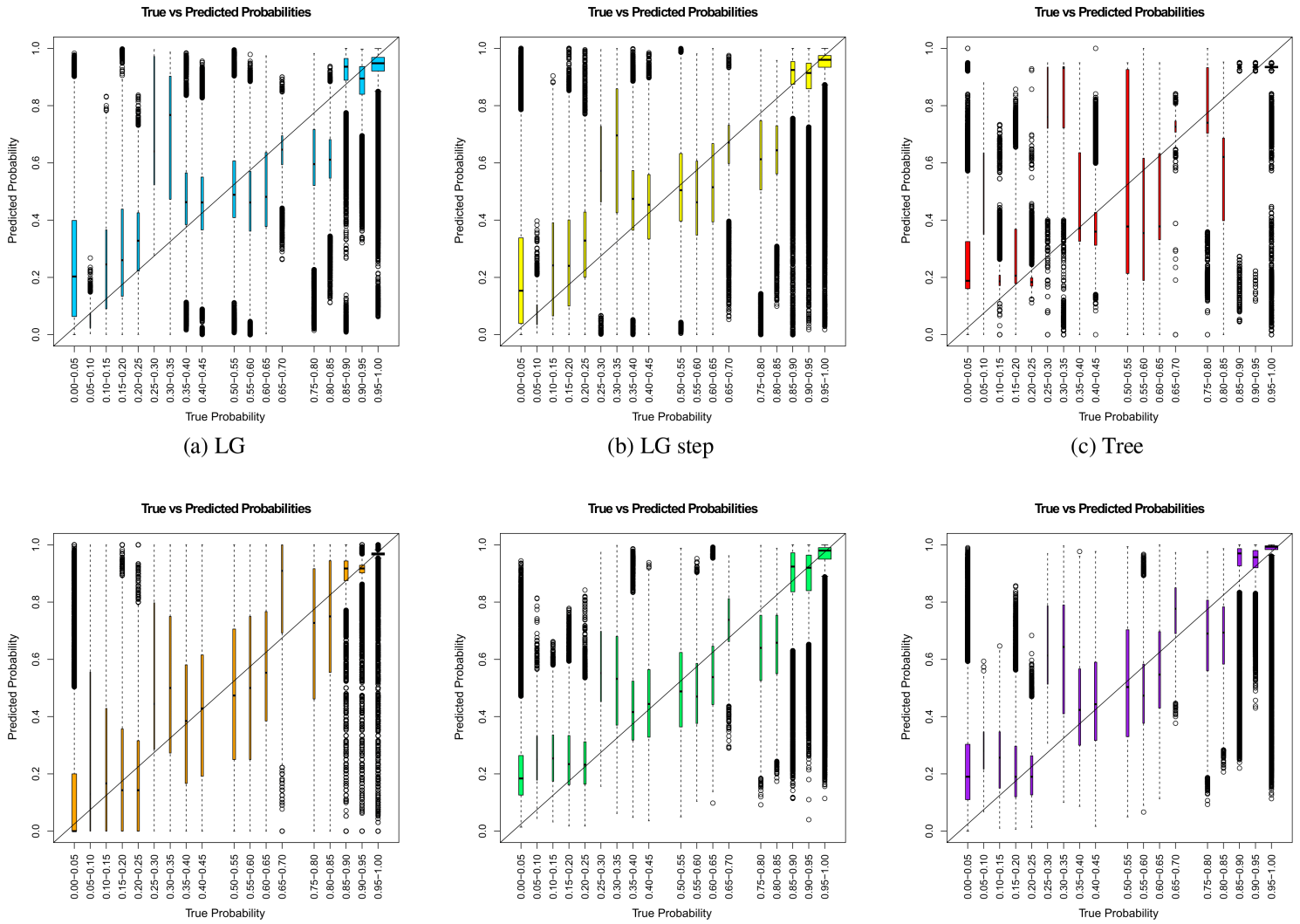
Figure 4: Iowa recidivism data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

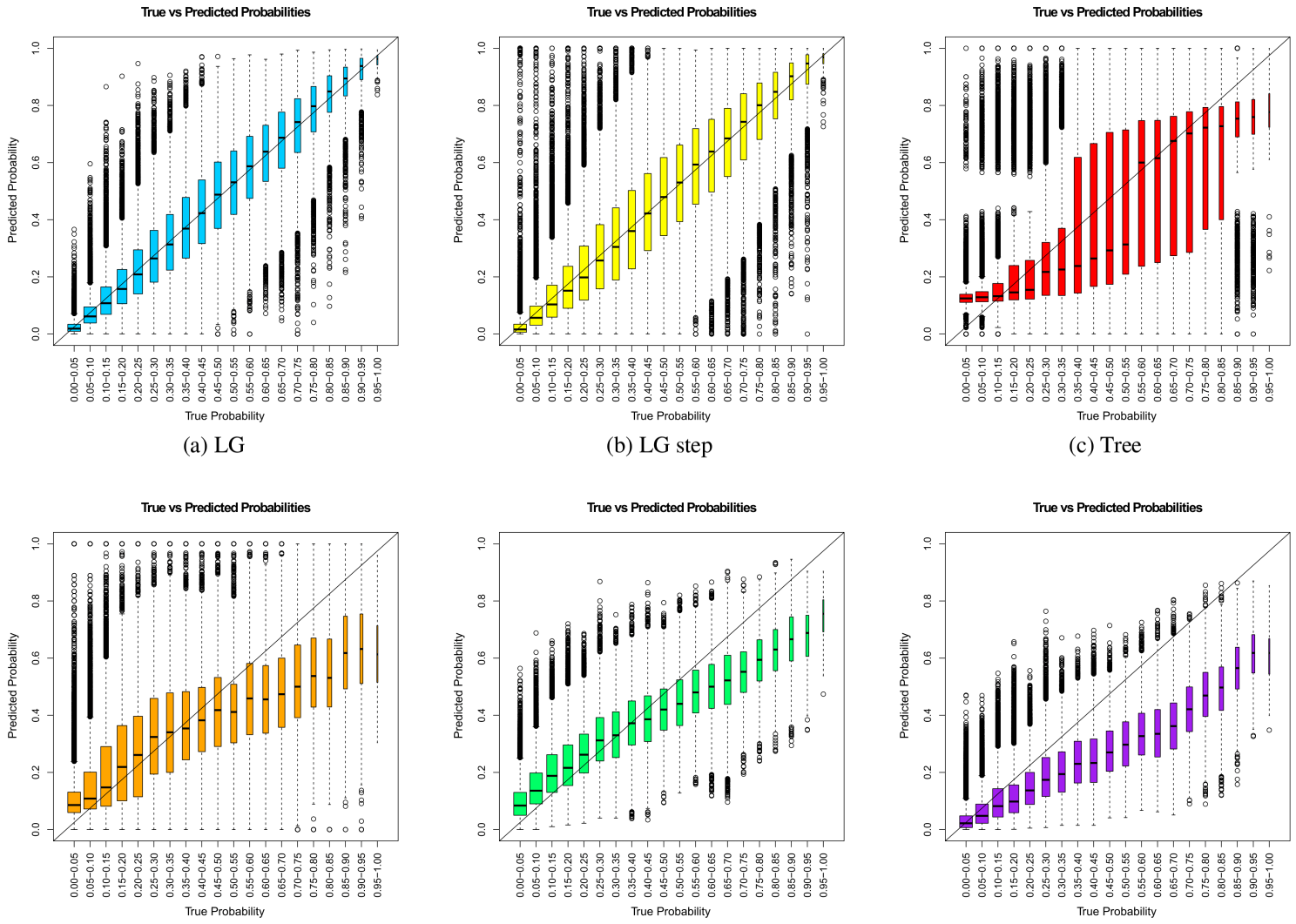
Figure 5: HMEQ data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

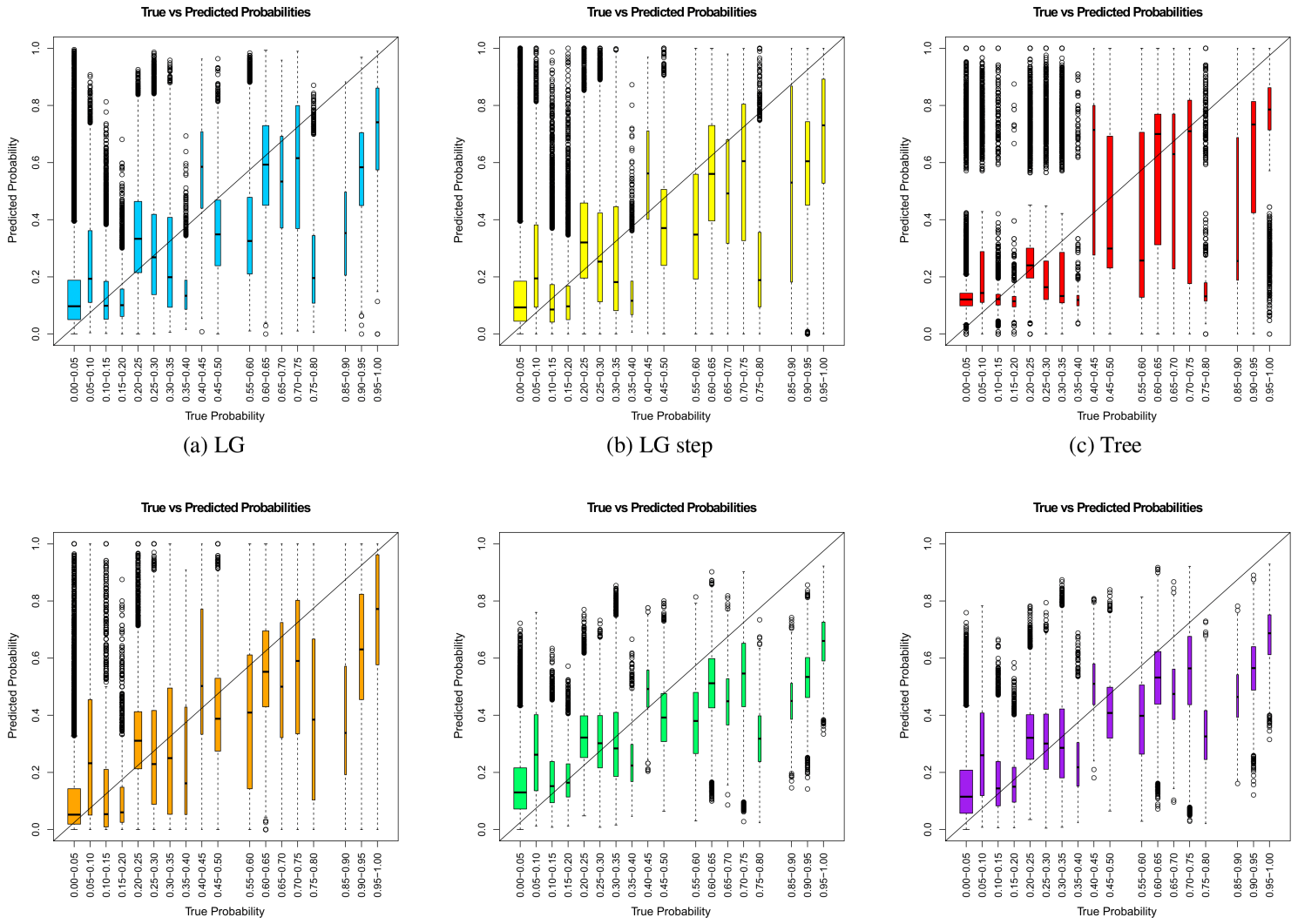
Figure 6: HMEQ data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

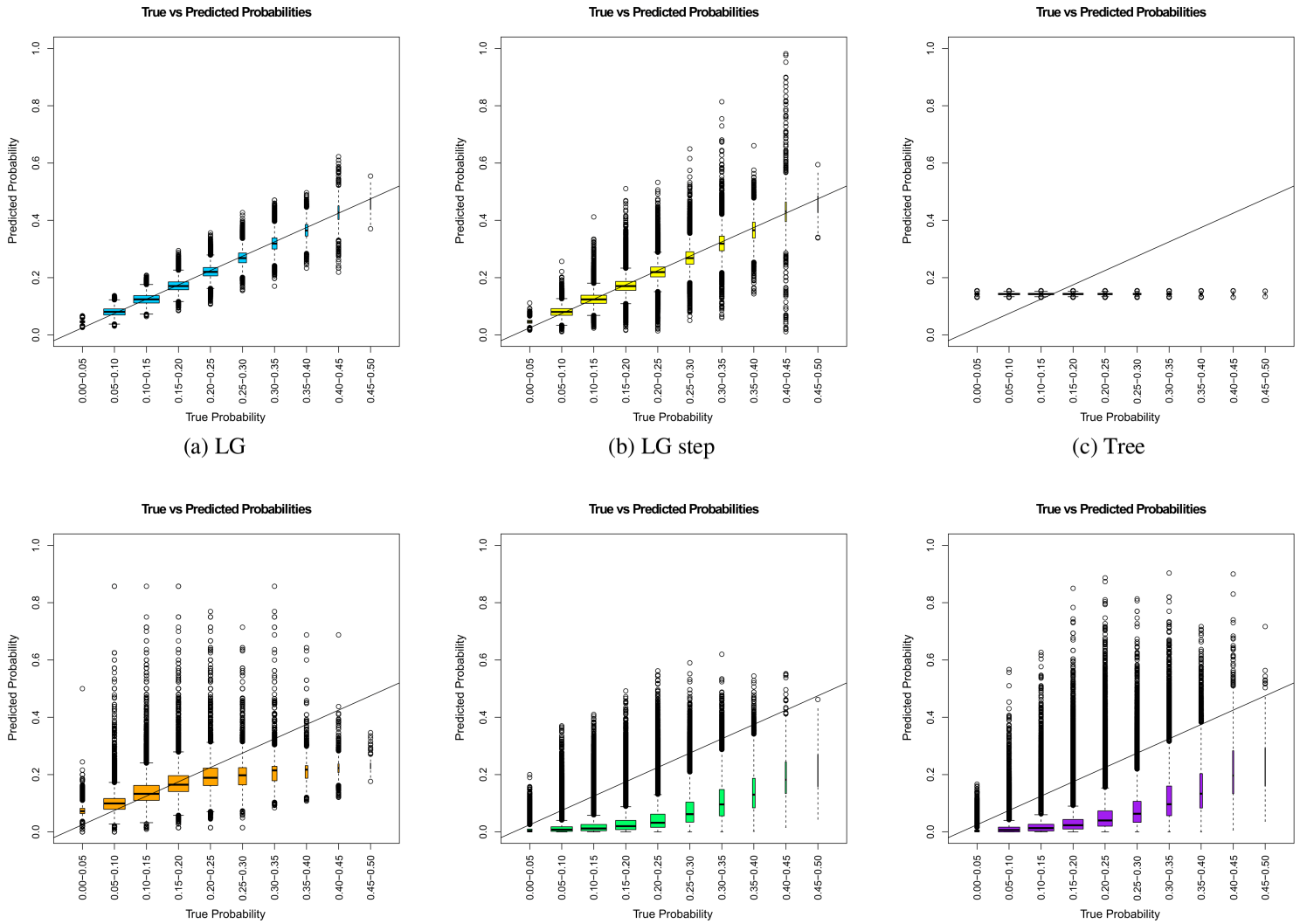
Figure 7: German credit data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

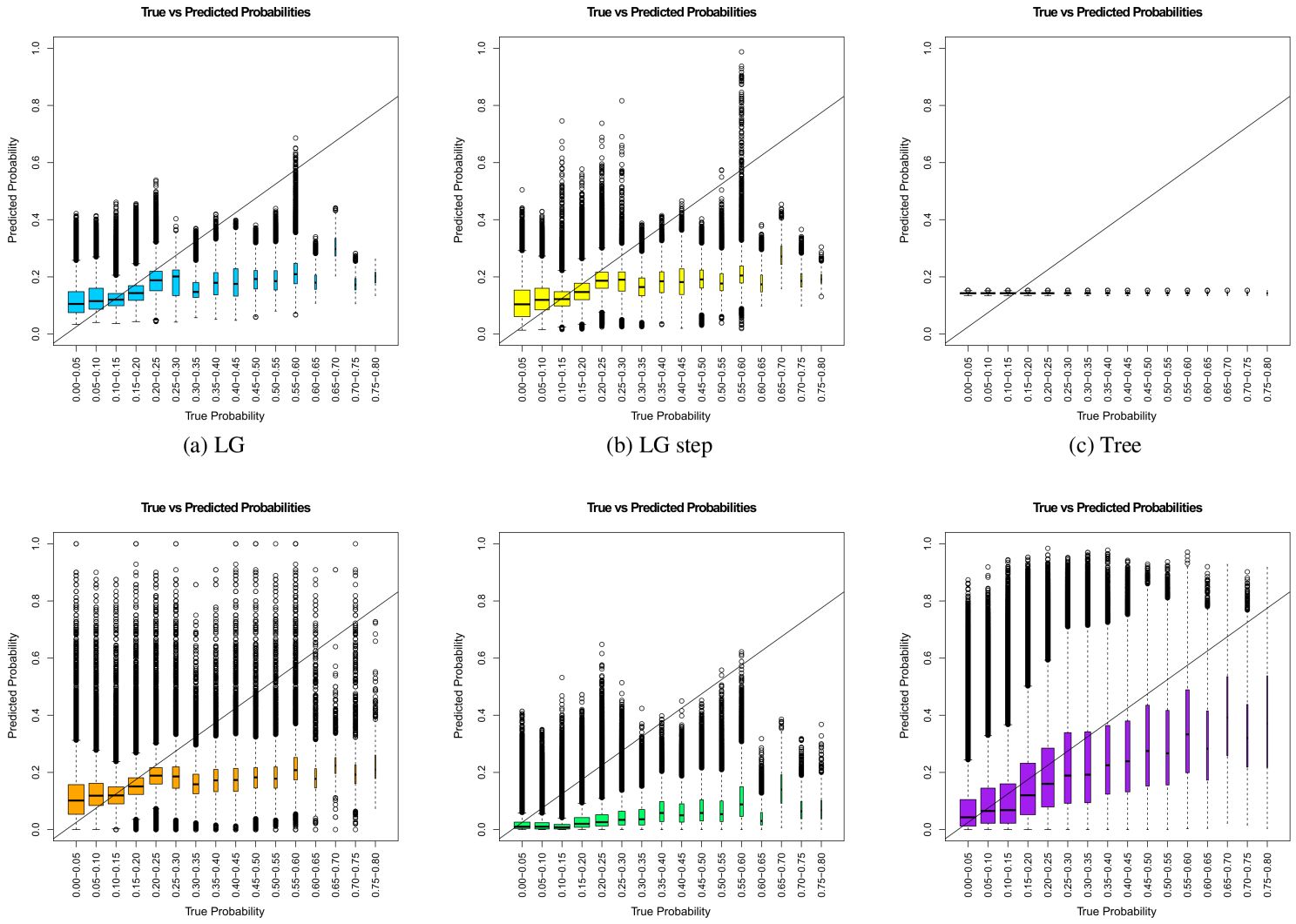
Figure 8: German credit data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

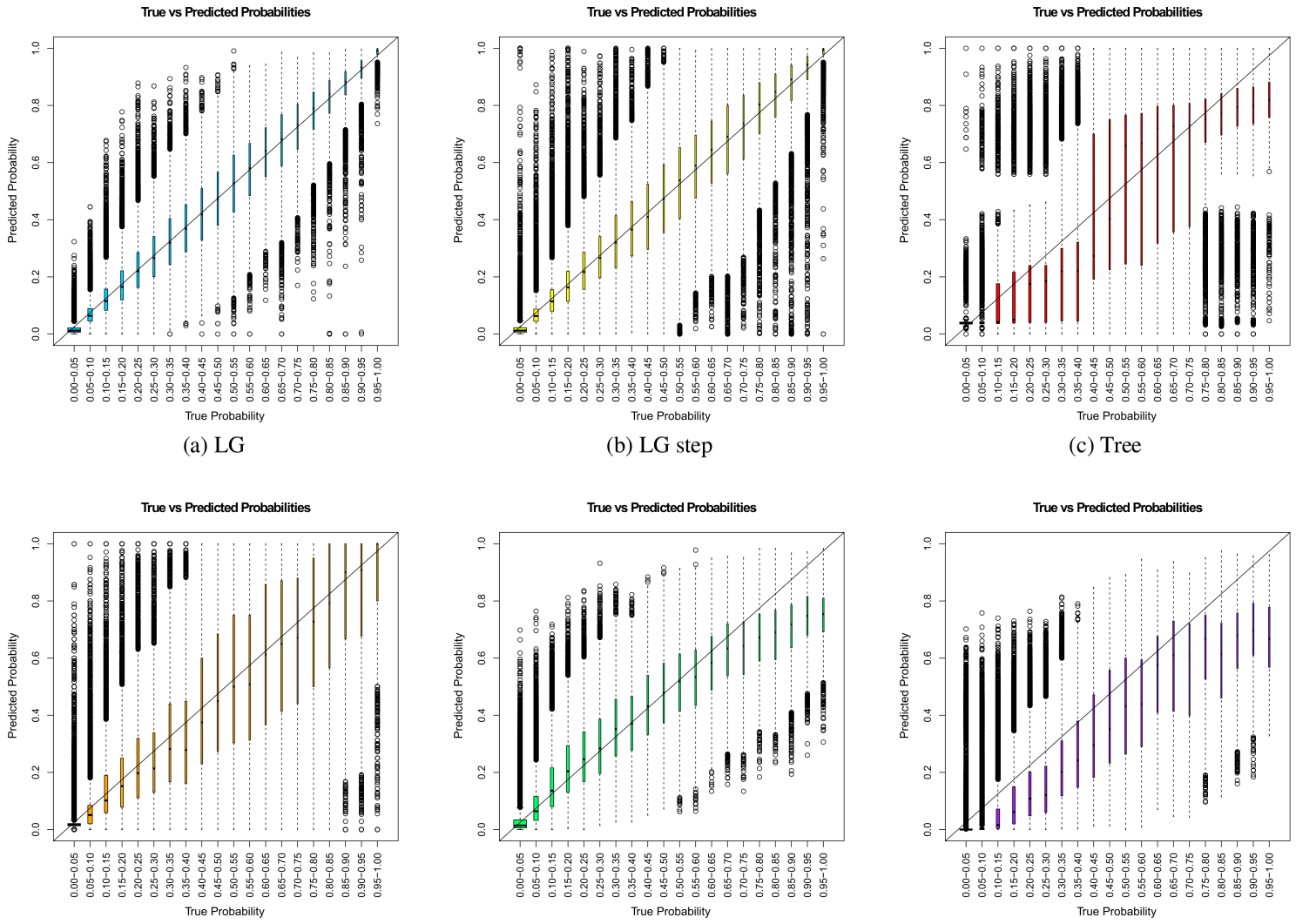
Figure 9: Marketing promotion campaign data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

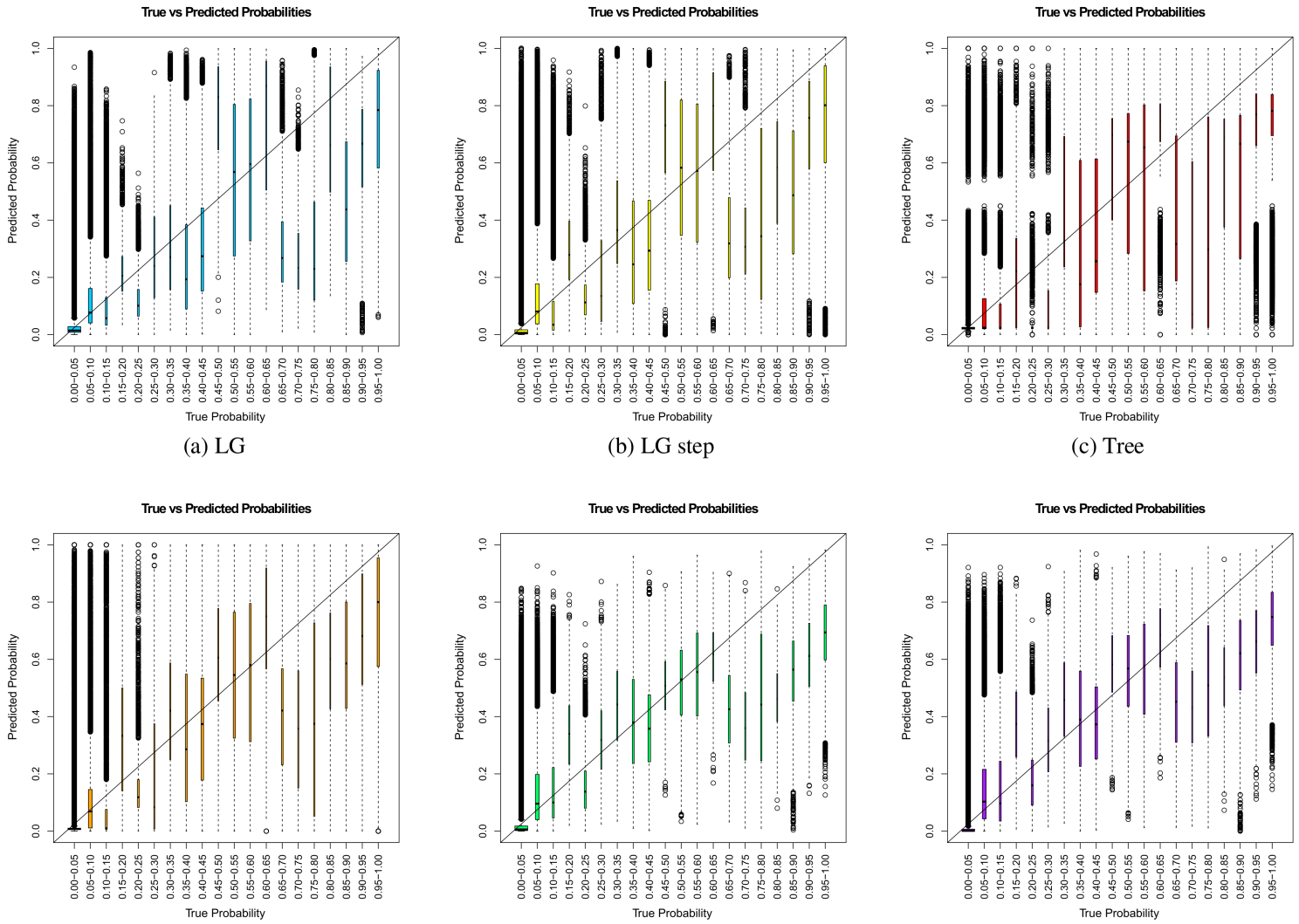
Figure 10: Marketing promotion campaign data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

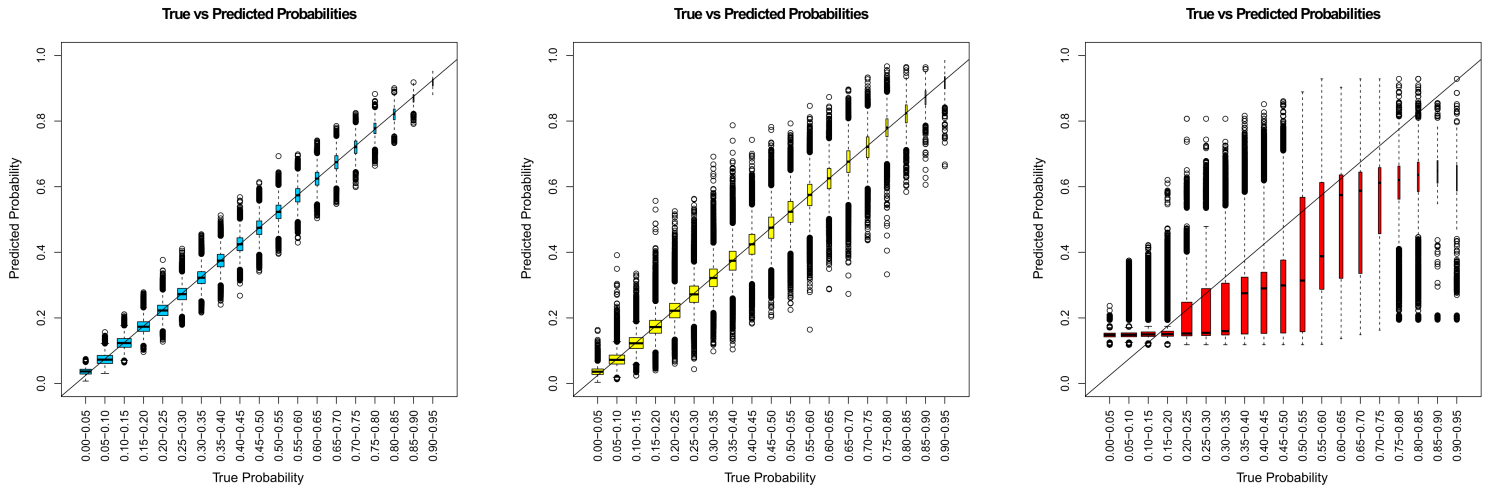
Figure 11: Bank marketing data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



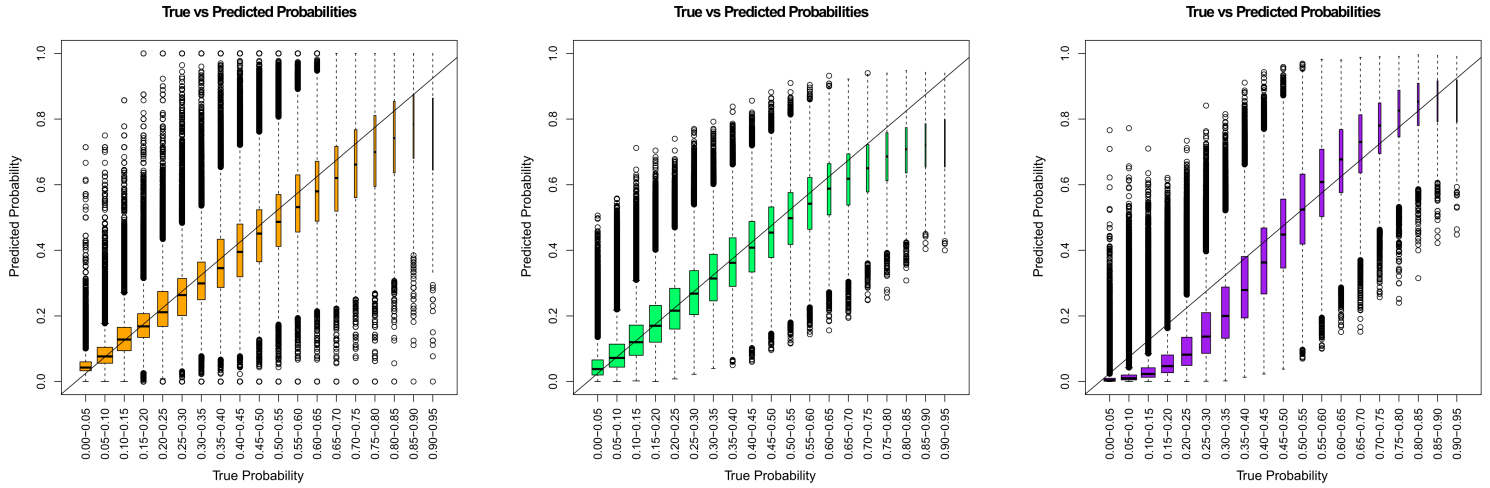
(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

Figure 12: Bank marketing data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.

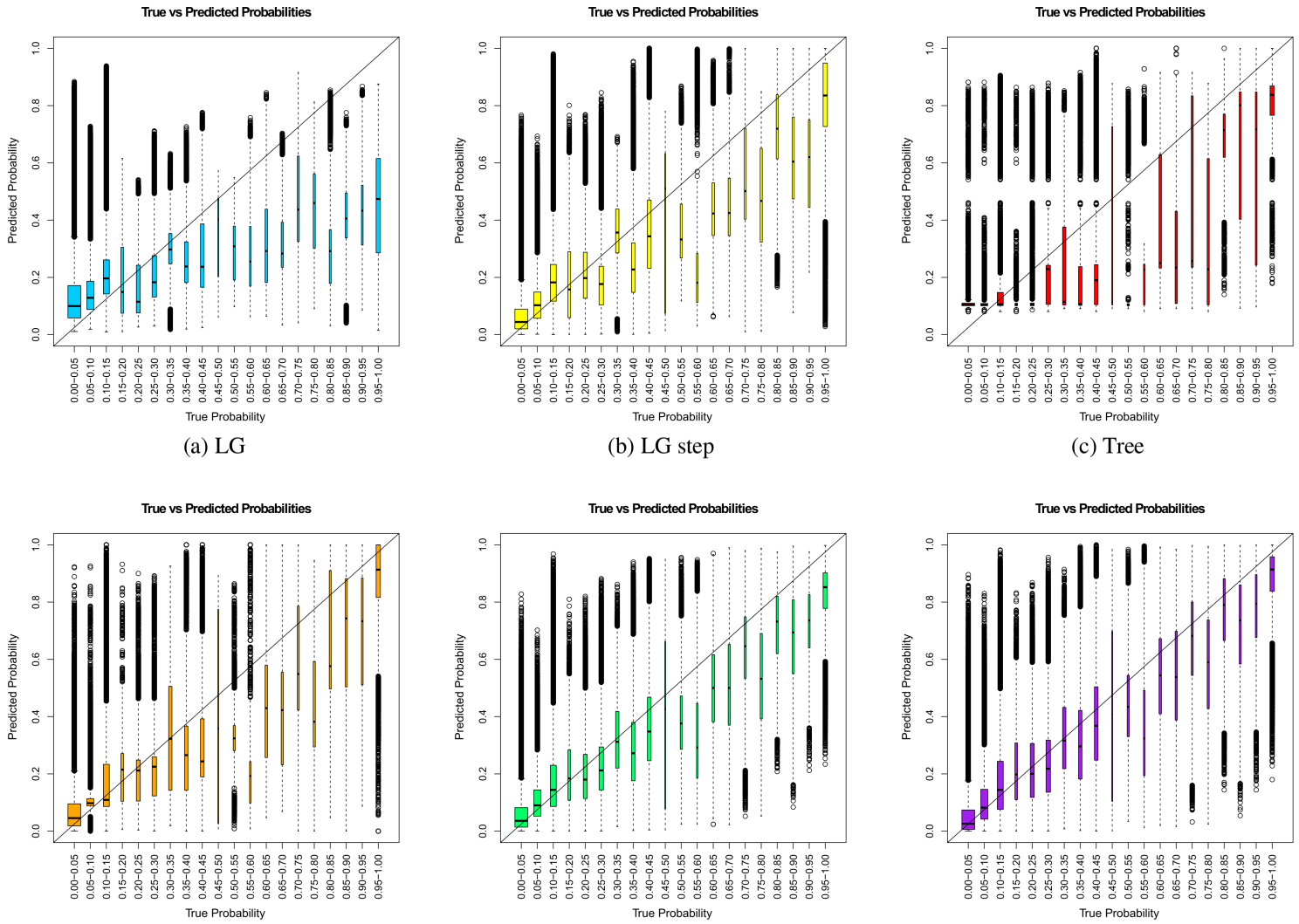


(a) LG (b) LG step (c) Tree



(d) Adj tree (e) RF (f) Adj RF

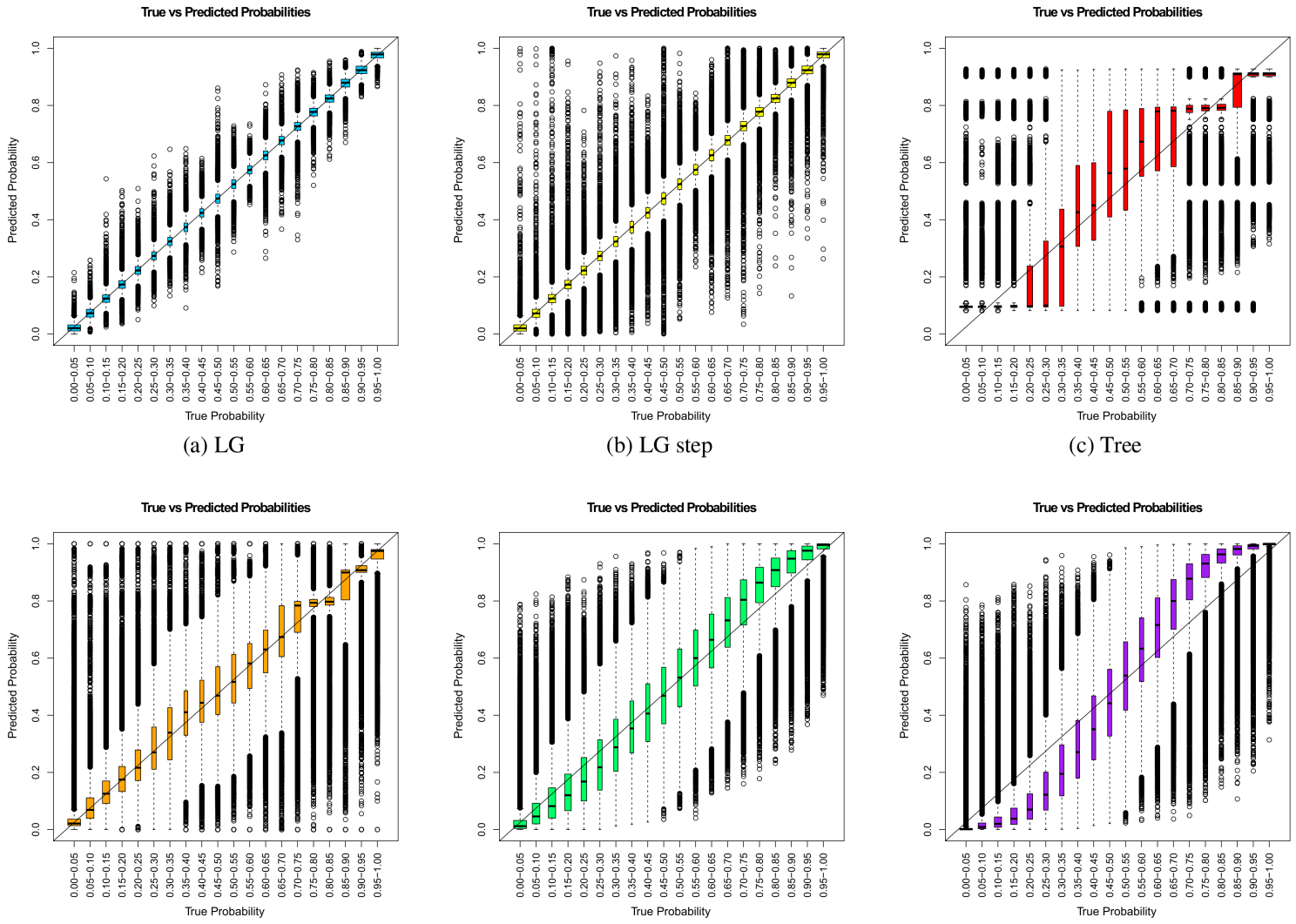
Figure 13: Churn data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

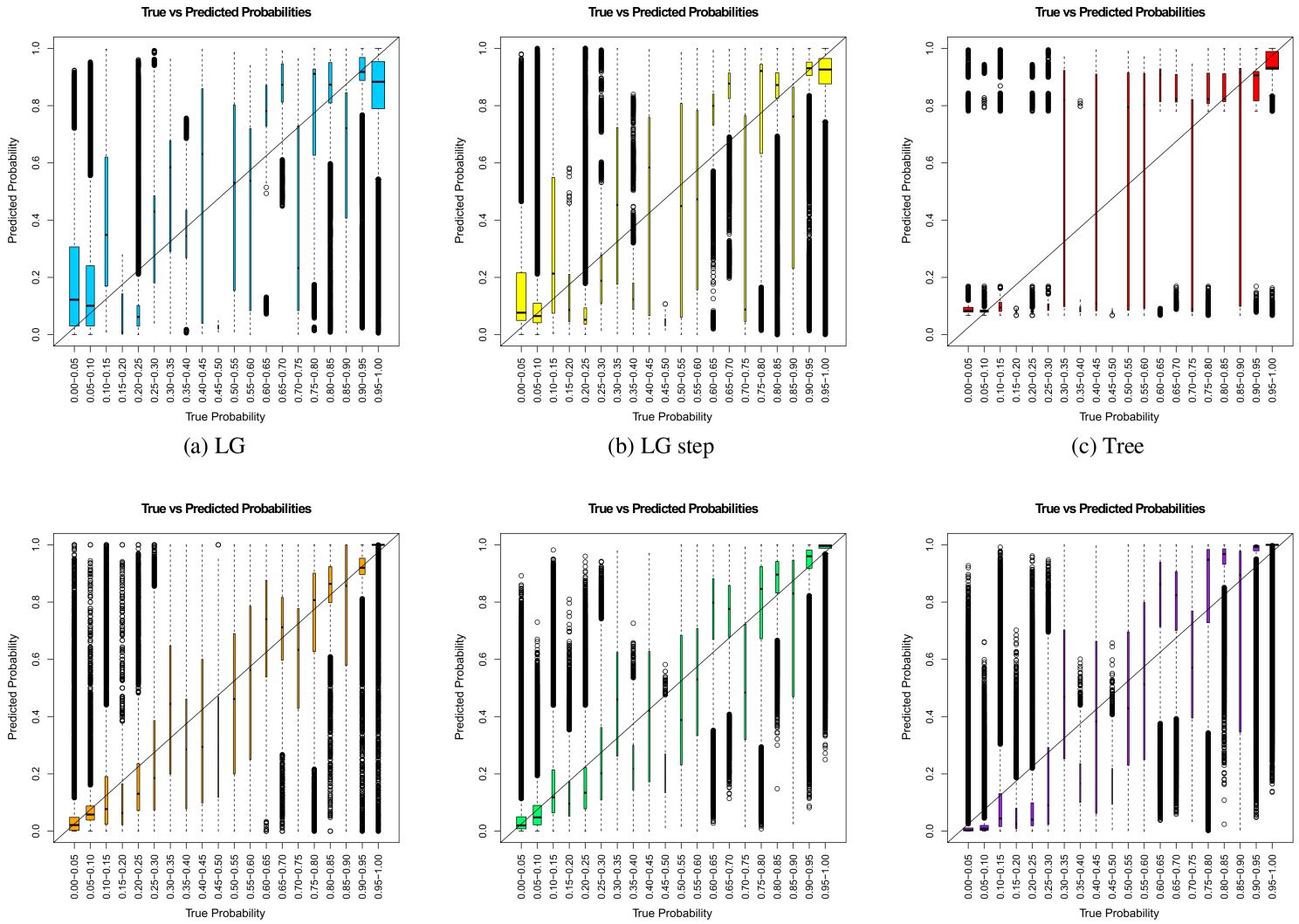
Figure 14: Churn data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

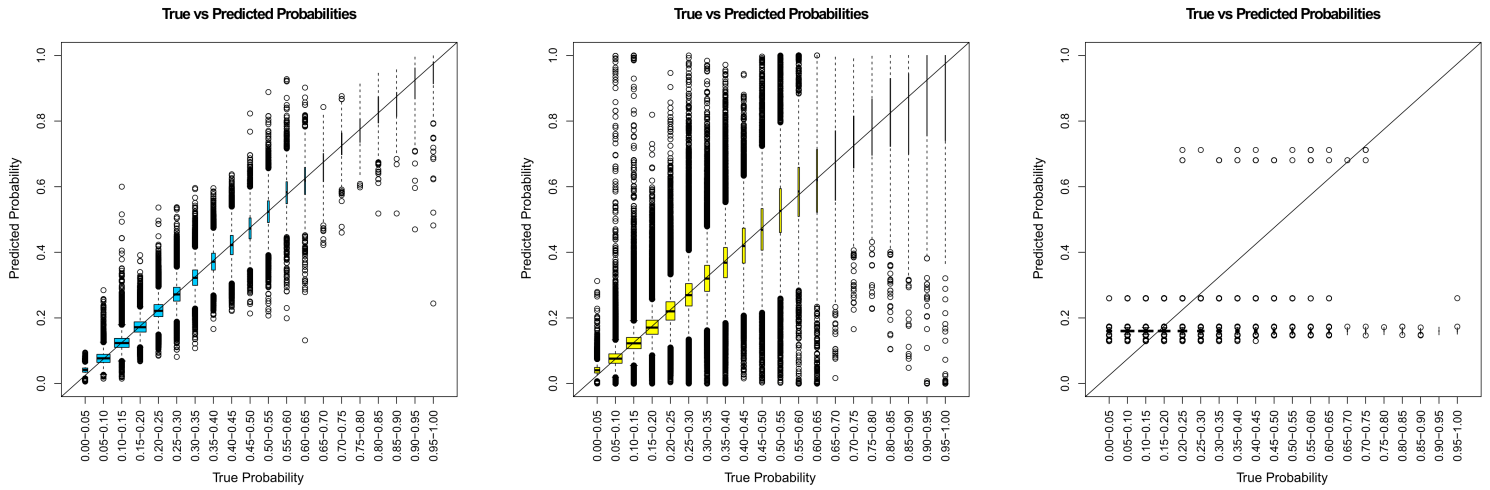
Figure 15: Internet churn data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

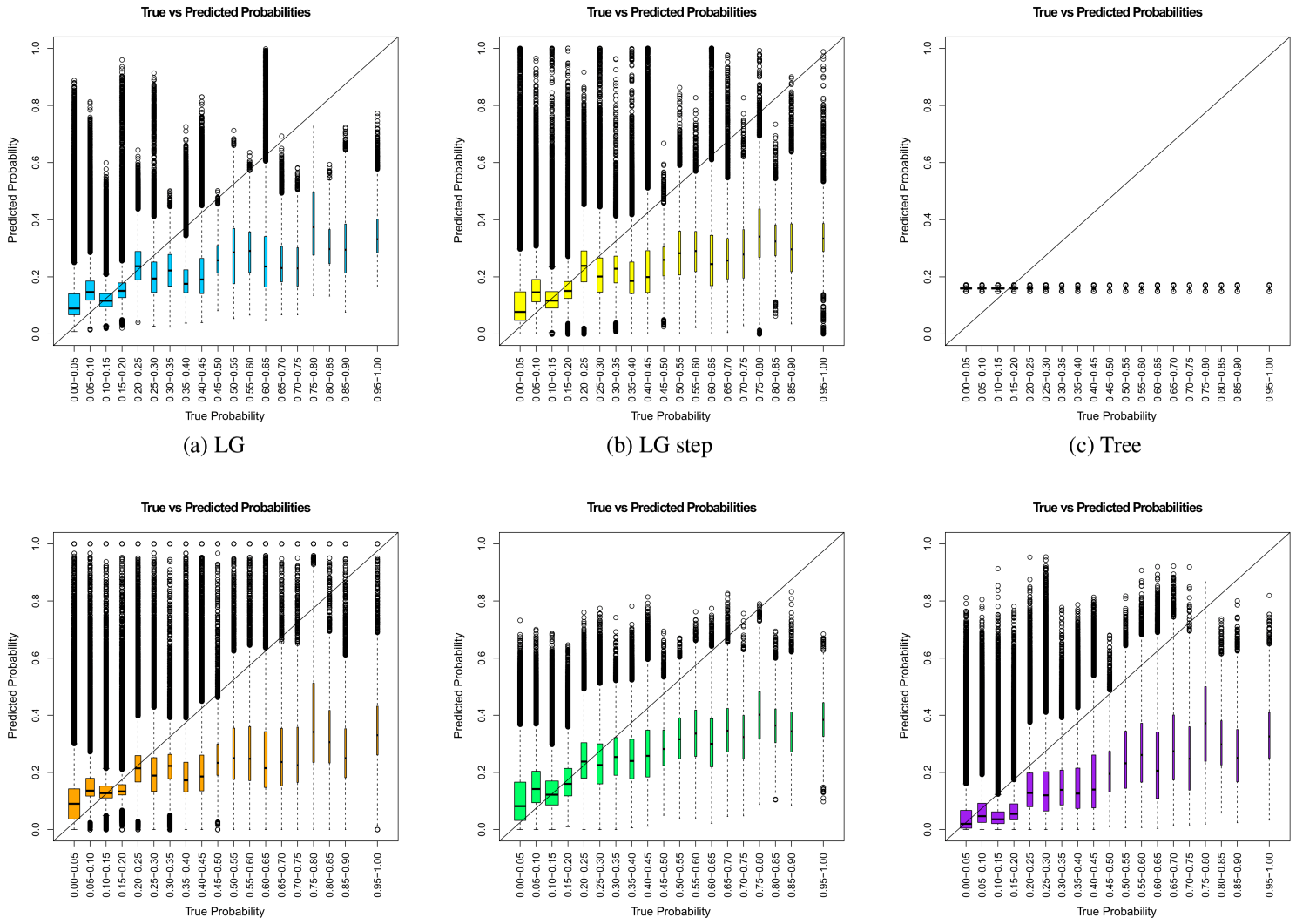
Figure 16: Internet churn data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

Figure 17: Loan data: true probabilities vs predicted probabilities for six machine learning approaches based on the LGM. Individual values of all data points from 1,000 replicates are displayed.



(a) LG (b) LG step (c) Tree

(d) Adj tree (e) RF (f) Adj RF

Figure 18: Loan data: true probabilities vs predicted probabilities for six machine learning approaches based on the TM. Individual values of all data points from 1,000 replicates are displayed.