On the Application of Supervised Machine Learning to Trustworthiness Assessment

24.04.2013

Sascha Hauke, Sebastian Biedermann, Max Mühlhäuser and Dominik Heider

TECHNISCHE
UNIVERSITÄT
DARMSTADT

24.04.2013

Technical Report No. TUD-CS-2013-0050
Technische Universität Darmstadt

Telecooperation Report No. TR-014,
The Technical Reports Series of the TK Research Division, TU Darmstadt
ISSN 1864-0516
http://www.tk.informatik.tu-darmstadt.de/de/publications/

On the Application of Supervised Machine Learning to Trustworthiness Assessment

Sascha Hauke*†, Sebastian Biedermann†, Max Mühlhäuser*†
*Telecooperation Lab and †CASED
Department of Computer Science
Technische Universität Darmstadt, Germany
{Sascha.Hauke, Sebastian.Biedermann}@cased.de,
Max@informatik.tu-darmstadt.de

Dominik Heider
Center for Medical Biotechnology
Deptartment of Bioinformatics
University of Duisburg-Essen, Germany
Dominik.Heider@uni-due.de

***Abstract*—State-of-the art trust and reputation systems seek to apply machine learning methods to overcome generalizability issues of experience-based Bayesian trust assessment. These approaches are, however, often model-centric instead of focussing on data and the complex adaptive system that is driven by reputation-based service selection. This entails the risk of unrealistic model assumptions. We outline the requirements for robust probabilistic trust assessment using supervised learning and apply a selection of estimators to a real-world data set, in order to show the effectiveness of supervised methods. Furthermore, we provide a representational mapping of estimator output to a belief logic representation for the modular integration of supervised methods with other trust assessment methodologies.**

***Keywords*-supervised prediction; trust models; machine learning**

I. Introduction

Computational trust models provide a grounding for trust assessment within the extended framework of probability theory. A commonly accepted (though somewhat reductionist, cf. [1]) point of view holds *trust* to be a "*subjective probability with which an agent* [the trustor] *assesses that another agent* [the trustee] *[. . .] will perform a particular action*" [2]. In this paper, we will follow this definition of trust, as well as the notion that trust is a dyadic, directed and conditionally transitive relation. Furthermore, *trust assessment* will refer to the estimation of the trustworthiness of the trustee by the truster, using an appropriate statistical estimator.

Experience-based Bayesian prediction methods are the mainstay of computational trust models. However, reinforcement learning, prevalent in their model design, still offers room for improvement. The reliance on a single type of predictor (either direct or reputation-mediated experience), for instance, leads to poor generalizability. While better generalizability can be reached by direct modification of the trust model and the introduction of new assumptions and model parameters, the resulting increase in model complexity is undesirable.

A number of approaches, particularly stereotyping trust models [3], [4], seek to address the generalizability issue by leveraging supervised learning for trustworthiness prediction. These approaches provide monolithic trust models centered around supervised feature-based prediction. Their focus, however, is on model-building and the presented models require a high discriminatory power of the provided feature set. Additionally, the distributional assumptions that enable supervised learning methods to build a prediction model depend heavily on the process that generates the data. Here, the influences of a reputation system on the selection and data generation process are often not taken into account, leading to unrealistic distributional assumptions when creating simulated datasets for model validation.

Consequently, since the quality of the prediction is therefore predicated on the quality of the data that is presented to the prediction model, trust assessment has to be considered not just from a model-based, but also from a data-driven perspective. To this end, we have compiled a real-world dataset[1] of hotel features and ratings, which exhibits distributional properties induced on the data generation process by reputation-based selection. To this dataset, we apply several off-the-shelf machine learning algorithms, in order to investigate to what extent the features presented on a hotel booking website encode a hotel's trustworthiness.

In the latter part of this paper, we will discuss the peculiarities of the dataset, the results of applying supervised learning methods, and describe how to integrate them with existing trust models, e.g., reputation-based methods, by providing a mapping to a belief logic representation.

In the following, we present the assumptions and preconditions for performing non-parametric and model-free supervised prediction in trustworthiness assessment (section II). The hotel dataset is explored and different regression machines are tested on this real-world data in section III. In sections IV and V, we present and discuss the results and propose a mapping of the estimates to the opinion space representation of commonly used belief logics. Finally, we briefly reference related work (sec. VI) and provide a concluding section that also outlines future work (sec. VII).

[1]This dataset, containing more than 3000 hotels, with 33 features for each hotel, is made available, so that our results can be reproduced (and improved upon).

II. METHODS

This paper will not attempt to present a complete trust model based around a specific supervised prediction method. Rather, we will present the requirements that a supervised prediction approach for trust assessment has to meet, discuss its application to the data-set and provide a mapping (in section V) that enables the integration of the prediction results with existing trust models.

Furthermore, we will use non-parametric, model-free learning methods in order not to be constrained by model assumptions and ease the burden of excessive parameterising for the user.

We will consider prediction methods that operate in *batch* mode. The data we are evaluating in section III are stable with regard to concept drift – that is, the value of the regressand does not change rapidly. In the given scenario (Hotel Ratings), dataset updates, in the form of newly added hotels and ratings, are comparatively infrequent. Therefore, we do not consider *online* training. Model update is achieved by retraining the regression machines with the entire, updated dataset. It is therefore fundamentally equivalent to estimator training, and will not be specifically discussed in detail.

A. Pre- and Postconditions

As a *training precondition*, trust computation based on supervised learning requires a training dataset consisting of $n \in \mathbb{N}, n \gg 0$ records in the form $(\mathbf{x}, y) = (x_1, x_2, \ldots, x_m, y)$. y is the dependent variable, in the case of trustworthiness assessment ideally the true trustworthiness score of a particular trustee, and the vector $\mathbf{x}$ consists of a number m of observable attributes (or *features*) $x_1, x_2, \ldots, x_m$ that are used as input variables. A model-free supervised learning mechanism creates its own prediction model from the data.

As an *assessment precondition*, trust computation requires, once a trained regression machine is available, a feature vector $(x_1, x_2, \ldots, x_m)$ for computing an estimated trustworthiness score $\hat{y}$.

Within the scope of a formal trust model defining trust as a probability, the *postcondition* of the trust computation is, at the least, a probability score. The further specifics of this postcondition is determined by the representational model used, for instance for decision making. Thus, when using the *CertainTrust* [5] representational model, we require a proper probability score, as well as a goodness-of-fit (*gof*) characteristic for determining the certainty parameter.

When estimating probabilities that are to be used in rigorous reasoning, the *consistency* [6] of the estimate is an important prerequisite (see section II-B). A definition of the consistency of estimators will be given in the following. Consistency of the estimator is not only an important postcondition for probability machines, but it also enables us to use a experience-based Bayesian trustworthiness estimate as an estimate for the unobservable trustworthiness of a trustee, i.e., y.

In particular, we will investigate two distinct cases. First, we consider a regression model in which a trustworthiness score of a particular trustee is available in the training dataset as a probability score $0 \leq y \leq 1$. Since this is unobservable, we will substitute an estimate in the form of a reputation score. In order to meet the consistency requirement for reasoning, this estimate itself should be consistent.

Second, we will consider a case where only a class label in $\{0; 1\}$ is available in the training data to classify a particular trustee. However, our goal is still to determine an actual probability score $p \in [0; 1]$ for each trustee. For this, we will use so-called *probability machines* [7]; that is, supervised estimators that are known to provide consistent probability estimates from binary regressands.

B. Consistent Trustworthiness Estimation

In the broadest sense, we consider the decision whether or not to trust as a binary classification problem – a truster classifies a trustee as either trustworthy or untrustworthy. In this sense, trustworthiness classification is a discriminatory problem suitably assigned to statistical learning methods. However, in order to satisfy the definition of trust as a subjective probability [2], assigning a class label is insufficient. Rather, the goal in trust assessment is estimating the *probability of class membership*, establishing just *how* likely a particular trustee is to be trustworthy.

Thus, the aim of trustworthiness prediction is to reliably estimate the probability of the trustee acting in a trustworthy manner in the next interaction with the truster, based upon representative input data. Thus, if $y \in \{0; 1\}$ is the outcome of such a future interaction, the goal is to compute a *conditional* probability $P(y = 1|\mathbf{x})$ given the features $\mathbf{x}$. For binary outputs, it follows that $P(y = 1|\mathbf{x}) = E(y|\mathbf{x})$. Both trustworthiness assessment by experience-based Bayesian prediction methods and probability machines leverage this equality in the estimation process.

1) Experience-based Bayesian Trustworthiness Prediction Model: State-of-the-art trust models [8] rely on Bayesian prediction models that take experience from past interactions as inputs to compute a probability score. This probability score can be interpreted as the probability that the trustee will act as expected in a future interaction. Technically, we face a classification task with binary class labels for the input (and output) data, i.e., class labels trustworthy and untrustworthy. The posterior probability distribution we want to estimate is a Bernoulli distribution. In particular, the desired probability score is the point estimate of its expectation value. This can easily be obtained by computing the expectation value of the Bernoulli distribution's conjugate prior, a Beta distribution.

Bayesian trust estimators (e.g., [5]) use experience from prior interactions as input. Their output (in the case of binary

input variables) is the probability that the *next* interaction with a specific trustee will be a positive one. A fundamentally important quality of naive Bayesian estimation is its *consistency* [6]. Informally speaking, an estimator is consistent, if the error of the prediction converges to zero in the limit *with high probability*.

Formally, the consistency of an estimator can be defined thusly [6]:

Definition 1: Let sample $X = (X_1, \ldots, X_n)$ be a member of a sequence corresponding to $n = n_0, n_0 + 1, \ldots$.

1) A sequence of random variables X_n defined over sample spaces $(\mathcal{X}_n, \mathcal{B}_n)$ *tends in probability* to a constant c ($X_n \xrightarrow{P} c$) if for every $a > 0$ it holds that $P[|X_n - c| \geq a] \to 0$ as $n \to \infty$.
2) A sequence of estimators δ_n of some parameter $g(\theta)$ is *consistent* if for every $\theta \in \Omega$ it holds that $\delta_n \xrightarrow{P_\theta} g(\theta)$.

The basic prediction model of the estimators used in [9], [5] is a point estimate of the expectation value of the prior Beta distribution. That is, if r and s are the sum of positive and negative prior interactions between truster and trustee, the probability estimate[2] is $\frac{r+1}{r+s+2}$. Here, the use of the expectation value as an appropriate estimator is due to the equality $P(y = 1|\mathbf{x}) = E(y|\mathbf{x})$. The consistency of this estimator follows from the consistency of the mean as an estimator.

Consequently, experience-based naive Bayesian prediction yields accurate trust scores, under the assumptions that prior experience is a reliable predictor for future behaviour and that the available prior experience is sufficient – with regard to both quality and abundance – for obtaining a representative point estimate.

The consistency of the estimation method is an important prerequisite for rigorous reasoning. The quality of convergence in the limit enables reliable probability assessment of past performance, which is the primary predictor for trustworthiness in computational trust models. Based on the consistency properties of the mean as an estimator of the expectation value, we will, in the following, assume that Bayesian trustworthiness estimators represent an adequate regressand for supervised machine learning approaches.

2) Regression Machines for Trustworthiness Prediction: A key argument behind the introduction of experience-based computational trust modelling was the scarcity of traditional cues related to trustworthiness in computer mediated interactions [8]. A cue for trustworthiness can be thought of as a feature or set of features that a trustee possess that are supposedly representative of its trustworthiness. While *traditional* cues learned from interactions in brick-and-mortar environments often cannot be applied to online interactions, modern online services expose a wealth of observable features. These can form the basis for learning new cues, which in turn can provide better generalizability for computational trust assessment.

Data mining approaches for exploiting high-dimensional feature spaces for probability estimation tasks are numerous. Parametric models, such as logistic regression, are traditionally applied there. However, they suffer from considerable drawbacks that limit their use in trust assessment in computer mediated interactions. In particular, parametric models have to be specifically fitted to the problem they are to address. In order to avoid model misspecification, predictors and supposed interrelations have to be input correctly. This limits their use considerably considering the scalability and flexibility required in data-rich environments where features can exhibit different scale types, dimensionality and correlation structures [7].

Model-free, non-parametric regression machines support the robust estimation of conditional probabilities from feature sets of different scale types and potentially high dimensionality. They make no distributional assumptions for the vector of features, make no restrictions on the length of the feature list, and do not rely on a specified model as a starting point [7]. In order to allow for *robust* probability estimation and thereby enable rigorous and meaningful inferences with regard to the trustworthiness of a trustee, consistency of the regression model has to be established. When using a Bayes estimate of the trustworthiness score as regressand, consistency is inherent in the consistent Bayes estimator.

However, when using a class label, instead of an already consistent estimate of the trustworthiness score, the supervised estimator itself has to be consistent. Malley et al. [7] term consistent non-parametric and model-free probability estimators that estimate the conditional probability function for a binary outcome as *probability machines*. We will apply several different probability machines to the task of trustworthiness assessment, namely, Random Forests [10], k-Nearest Neighbour [11] approaches and Decision Trees [12], [13] .

Regression Model: Following [7], we will treat the probability estimation problem constituted by trust assessment as a *non-parametric regression* problem. Thus, the regression machine will serve to estimate the non-parametric regression function $f(\mathbf{x}) = E(y|\mathbf{x}) = P(y = 1|\mathbf{x})$, where $\mathbf{x}$ is a vector of features (regressors).

Methods of web data extraction, for instance, can be employed for gathering relevant information. However, the *true* regressand, that is the intrinsic trustworthiness of the trustee, is an unobservable variable in real-world applications. In its place, a point estimate from an experience-based naive Bayes estimation method can be used. Ideally, this is a robust reputation-based trust model, such as [9], [5]. Due to the mostly academical nature of these works and the consequent absence of their real-world application, widely-used basic reputation systems will have to be substituted instead. For testing of estimators as probability machines, we will use a

[2]We present a basic version here; [9], [5] allow for a further parameterisation of the prediction model.

binary dichotomisation of the reputation score.

Random forests [10] are non-parametric ensemble classifiers consisting of a multitude of decision trees. They are generally considered to be fast and and accurate classifiers that offer considerably better performance than single trees [11], for instance, CART[12] or M5 [13].

Random forests have several strengths that make them theoretically well-suited to trustworthiness assessment. In particular, they can handle high dimensional feature spaces of different scale types, with little user input. Thus, they can be presented with arbitrary sets of feature vectors that result from web data extraction, without requiring user-driven feature selection or model specification. Additionally, they typically provide robust estimates, even under conditions of missing data. Conveniently, Random Forests perform rudimentary error estimation using an OOB method[3] during the learning process.

In classification tasks, the output of a random forest is the mode of the classification outputs of its constituent classification trees. Instead of outputting a class label, the random forest can also return an estimate of the conditional probability $P(y|\mathbf{x})$. As we are concerned with *probability estimation* of binary classes, the probability estimate can be obtained by computing the proportion $\frac{|y=1|}{|y=0|+|y=1|}$, averaged over all constituent trees, when running the random forest in classification mode. In regression mode, the random forest consists of regression trees instead of classification trees. Thus, the probability estimates are averaged over the regression results of the individual trees, instead. For the prediction of hotel ratings (section III), we will use a random forest estimator in classification and regression mode, termed *classRF* and *regRF*.

The consistency of random forests has been shown by Biau et al. [14]. For a detailed description of random forest bootstrapping and classification procedures, see [10], [7].

K-Nearest Neighbour (*k-NN*) estimators are a special case of kernel density balloon estimators. The (simplified) classification process is intuitive: An unlabelled sample is classified by comparing its feature vector to labeled samples from a training set and choosing the k closest according to an appropriate distance metric. The class of the unlabelled sample is estimated by determining the mode of the k labels of the labeled neighbours. In a regression model with a continuous regressand, the mode can, for instance, be replaced by an inverse distance weighted average function.

Breiman [15] introduced a variation of nearest neighbour classifiers that combines several k-Nearest Neighbour into an ensemble classifier, using *bagging* (bootstrap aggregating). This is analogous to formation of random forests from decision trees. Thus, the output of the bagged *k-NN* (*b-NN*) is the mode of its constituent *k-NN* estimators for a classification task. A probability estimate can be obtained in the same manner as for the *classRF* random forest [7].

In recent publications dealing with the application of machine learning to trustworthiness assessment tasks [3], [16], decision trees have been used for classification tasks. There are several decision tree algorithms that can perform regression and are suitable for trustworthiness assessment. Specifically, we will test CART [12] and M5 [13] decision tree algorithms on the dataset.

Decision trees offer white box behaviour and interpretability of the generated models. They are also reasonably robust, performant and can deal with different scale types as input data.

We omit another popular estimator, support vector machines (*SVM*), because it cannot guarantee universal consistency [7].

In section III we present a real-world dataset and test the methods on it – with regard to their capability to predict reputation scores from the given features. We do not present synthetic data. This is done intentionally. The power of the machine learning methods described above is well-established. Generating synthetic data to show the discriminatory qualities of these methods would thus be only an – inadequate – replication of work. For an application of probability machines to benchmarking datasets, the interested reader is referred to [7].

III. Data

Hotel booking and ranking sites represent a real-world application of reputation systems that combine both electronic availability of the reputation data, as well as physical service provisioning in a mature and regulated market. The records furnished by hotel booking sites actually guide real customers to make a trust decision and, through their rating feature, provide a feedback mechanism. They provide the user not only with reputation scores for hotels, but also with collections of features, that are standardised, complete and verifiable to some extent. The physical nature of the service provisioning and the correspondingly required monetary collateral (e.g., costs of realty, furnishings, personnel, etc.) justify assumptions of slow concept drift and market persistence of individual hotels.

In order to test regression machines for trustworthiness assessment, we acquired a dataset of $3,006$ hotel records for hotels in 9 major European cities from a German hotel booking site. Each record consists of an ID, an aggregated rating score, the number of individual binary ratings that were aggregated into the rating score, as well as 33 features of various scale types (table I).

When rating a hotel, raters were asked '*Would you recommend this hotel?*' and could answer either *yes* or *no*. Individual ratings, therefore, are binary. Rating aggregation into an aggregate recommendation score is achieved via simple averaging. Ratings are only available as aggregate

[3]Therefore, they do not necessarily require dedicated cross validation to control overfitting.

Scale Type	Feature
Nominal	ID, City
Binary	*Payment Options*: Master, Visa, AmEx
	Hotel Ammenities: Laundry Service, WiFi, Restaurant, Bar, Bistro and Cafe, Steam Bath, Elevator, Special Access, Gym, Sauna, Solarium
	Room Ammenities: Telephone, TV, Radio, AC, Safe, Minibar, Desk, Hair Dryer, Bath Tub
Ordinal	Hotel Stars
Ratio	Aggregate Recommendation, Number of Recommendations
	Distances to next: Airport, Highway Access, Railway Station, Commuter Station
	Number of Rooms: Total, Single, Double
	Price

Table I
SCALE TYPES AND FEATURES FOR THE HOTEL DATASET

recommendation scores. In particular, no time series of individual ratings was available. Furthermore, raters were only able to rate hotels that they had booked through the booking site.

Overall, raters contributed $199,168$ ratings, of which $151,868$ ($\approx 76\%$) were positive and $47,300$ ($\approx 23\%$) were negative ratings. Of the $3,006$ hotels in the dataset, 356 ($\approx 11.8\%$) have not been rated. Of those $2,650$ hotels that have been rated, the mean number of ratings per hotel is 75.16 – the median, however, is considerably lower at 25 (for a summary, see table II). Figures 1(a), 1(b) show histogram information of aggregate recommendations, clearly displaying the peakedness of the empirical distribution and the effect of the excess positive individual and aggregate ratings (see also table II).

Figure 1(c) shows a long-tailed distribution of the number of recommendations per hotel, i.e., a small number of hotels have a high number of recommendations, while the vast majority of hotels have a comparatively small number of recommendations. Figure 1(d) plots the distribution of the recommendation score against the number of recommendations. The distribution evident in these figures hints at *preferential attachment* processes that are induced by the decision making and feedback mechanisms of the reputation system.

In section IV we apply the off-the-shelf regression machines described in section II to the hotel dataset. We follow the non-parametric regression function $f(x) = E(y|\mathbf{x}) = P(y = 1|\mathbf{x})$, where $\mathbf{x}$ is a vector of features (regressors). The aggregate recommendation score is used as regressand, while the 33 features listed in table I (omitting *ID* and *Number of Recommendations*) will serve as regressors. We assume that the aggregate recommendation score is an adequate surrogate for the unobservable true trustworthiness of each trustee (i.e., hotel), which is justified by the arithmetic mean being a consistent and stable estimator.

IV. RESULTS

In the following, we apply the estimators that were introduced in section II. First (section IV-A), we test the random forest, *k-NN*, CART and M5 decision tree algorithms in a regression scenario with the aggregate recommendation scores as unmodified regressands y. In addition, logistic regression was performed to provide a baseline.

Second (section IV-B), we use the probability estimation capabilities of the regression machines in a classification scenario (i.e., in a dichotomous regression scenario with values 0 or 1, with the estimators operating as probability machines). For this, we generated dichotomous outcomes from the aggregate recommendation scores. For each hotel, a new dichotomous response variable y was computed by using a binomial random number generator with the hotel's recommendation score as the corresponding probability. Random forests, *k-NN*, *b-NN*, CART and M5 decision tree estimators were trained using the new binary response variable and the 33 features of the hotel dataset as regressors. The estimators were *not* presented with the recommendation scores or the number of ratings per hotel.

In both cases, the area under the curve (*AUC*) was computed against the dichotomised response, based on the receiver operator characteristics (*ROC*).

10-fold cross validation (*CV*) was performed to check for overfitting. None of the estimators exhibited tendencies towards overfitting the data and the goodness-of-fit *gof* did not vary noticeably between random forest *OOB* estimates, standard holdout and *CV*. We evaluated *gof* according to several standard error measures (see table III) based on the difference between the estimates $\hat{P}(y = 1)$ and the recommendation score, which we assume to represent the true trustworthiness $P(y = 1)$.

Random forest estimators were applied in regression mode (*regRF*, to both recommendation score and class label regressands) and classification mode (*classRF*, to class label regressand). For each of these, two distinct configurations were chosen: one that guarantees consistency (according to [7]), in which individual trees were not fully grown, and one that grows the individual trees to their full extent, according to the default settings [10] for *regRF* and *classRF*. In the latter case, universal consistency of the random forest estimator cannot be guaranteed.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD	Skew	Kurt
Number	2	9	25	75.16	78	1531	132.40	4.03	23.08
Score	0.0	0.65	0.75	0.73	0.83	1.0	0.138	-0.89	1.38

Table II
DISTRIBUTION OF NUMBER OF RECOMMENDATIONS AND RECOMMENDATION SCORE

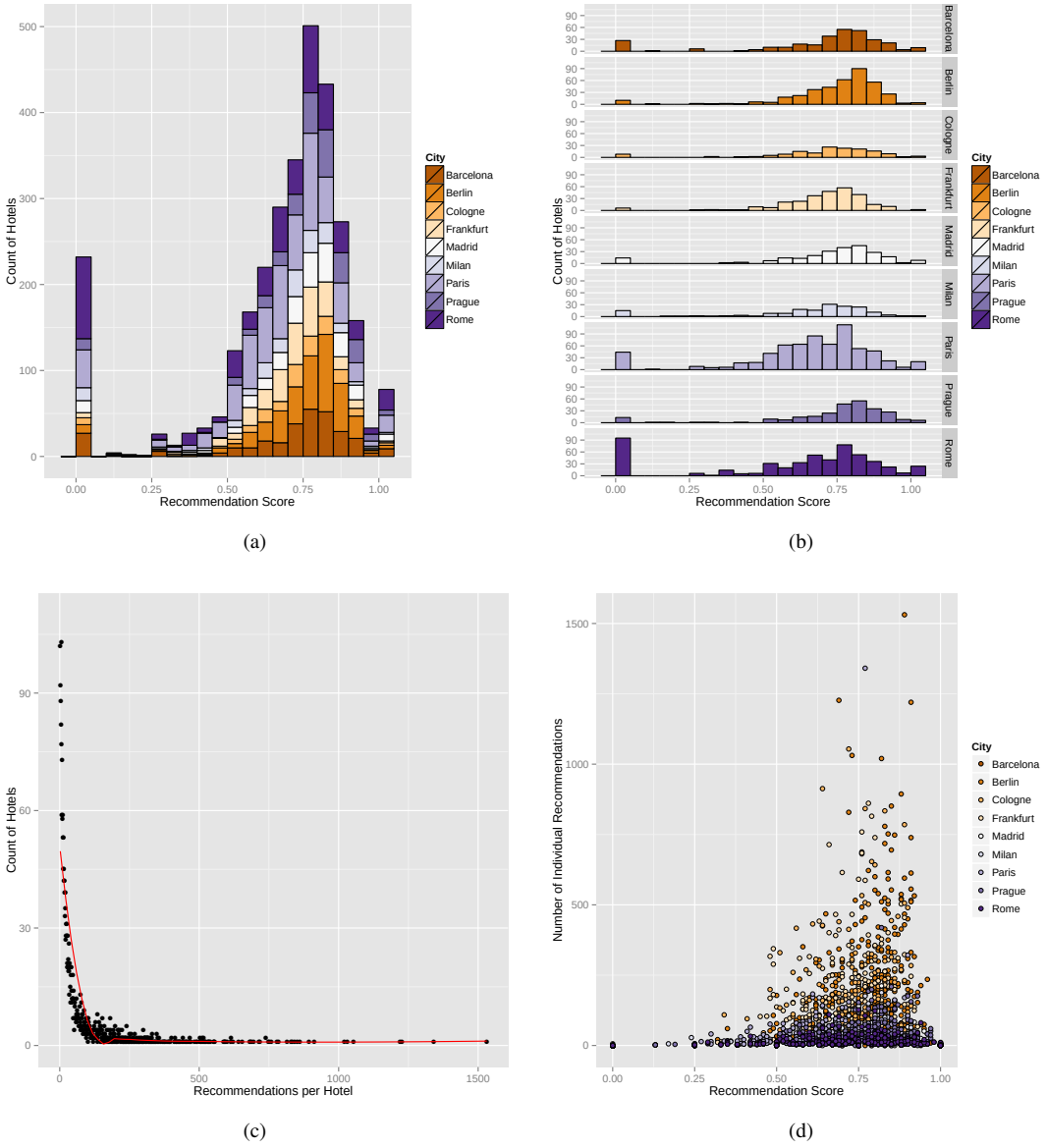


Figure 1. Aggregate Recommendations in the Hotel dataset

A. *Regression to Recommendation Score*

The results of applying the regression machines can be seen in table III, in terms of various goodness of fit measures (for a documentation of the measures, see [17]). The normalised root mean square error ($nrmse$, see definition 2) indicates that the Random Forest estimators perform marginally better than the decision trees. As per the mapping presented in the discussion section (section V), we consider a prediction informative, if the *percentage* $nrmse$ ($nrmse\%$) is smaller than 100. While all tree-based estimators (*regRF*, *M5*, *CART*) achieve an $nrmse\% < 100$, nearest neighbour and logistic regression return no informative results.

When considering the *AUC*, as per table IV, the random forests outperform the other estimators. However, the margin between the different methods is small, and the overall performance of all methods is only slightly better than random guessing (as indicated by an *AUC* of 0.5).

B. *Regression to Class Label*

When operating the estimators as probability machines, results of the probability estimation (tables V and VI) are qualitatively broadly similar to those of the regression

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	0	0.1	0.02	0.13	**91.8**	0	0.92	0.29	0.16	0.11	0.43	0.32
regRF (default)	0	0.09	0.02	0.12	**89.6**	-0.6	0.9	0.44	0.2	0.14	0.56	0.41
M5	0	0.09	0.02	0.13	**91.8**	0	0.92	0.44	0.16	0.11	0.53	0.38
CART	0	0.1	0.02	0.13	**94**	0	0.94	0.39	0.12	0.08	0.47	0.35
k-NN	0	0.11	0.02	0.14	**103.4**	-0.3	1.03	0.52	-0.07	-0.02	0.45	0.34
logit	0.29	0.33	0.17	0.42	**301.2**	39.3	3.01	2.37	-8.07	-2.12	0.39	0.26

Table III
AVERAGE GOODNESS OF FIT OF REGRESSION TO RECOMMENDATION SCORE (FOR A DOCUMENTATION OF THE MEASURES, SEE [17])

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	0	0.1	0.02	0.13	**94.2**	-0.1	0.94	0.36	0.11	0.08	0.45	0.33
regRF (default)	-0.02	0.12	0.02	0.15	**110.6**	-2.9	1.11	0.75	-0.22	-0.11	0.52	0.38
classRF (consistent)	0.26	0.26	0.09	0.29	**212.8**	35.6	2.13	0.13	-3.53	-1.43	0.39	0.29
classRF (default)	-0.01	0.11	0.02	0.15	**107.6**	-1.7	1.08	0.7	-0.16	-0.07	0.52	0.38
M5	0	0.1	0.02	0.14	**102.2**	-0.1	1.02	0.59	-0.04	0.03	0.49	0.37
CART	-0.46	0.46	0.23	0.48	**347.6**	-62.8	3.48	0.17	-11.08	-3.3	0.3	0.19
k-NN	-0.01	0.1	0.02	0.13	**96.8**	-1.2	0.97	0.28	0.06	0.04	0.36	0.27
b-NN	-0.01	0.1	0.02	0.13	**96.8**	-1.1	0.97	0.3	0.06	0.04	0.37	0.28
logit	0.3	0.41	0.35	0.59	**427.8**	41.5	4.28	3.8	-17.3	-2.84	0.27	0.23

Table V
AVERAGE GOODNESS OF FIT FOR REGRESSION TO A CLASS LABEL (FOR A DOCUMENTATION OF THE MEASURES, SEE [17])

	avg AUC	**MIN**	**MAX**	**± SD**
regRF (cons)	0.590***	0.563	0.604	± 0.012
regRF (def)	0.585***	0.565	0.599	± 0.014
M5	0.582***	0.565	0.6	± 0.012
CART	0.56***	0.543	0.575	± 0.01
k-NN	0.547***	0.543	0.55	± 0.01
logit	0.582***	0.563	0.603	± 0.014

Table IV
AVERAGE CLASSIFICATION PERFORMANCE WITH RECOMMENDATION SCORE AS REGRESSAND (***: p VALUE (95 % CONFIDENCE INTERVAL) OF ONE-SIDED WILCOXON TEST, AUC PREDICTION VS. GUESSING, I.E. $\mu = 0.5, p < 0.001$)

	avg AUC	**MIN**	**MAX**	**± SD**
regRF (cons)	0.568***	0.552	0.585	± 0.013
regRF (def)	0.547***	0.523	0.579	± 0.019
classRF (cons)	0.529***	0.503	0.545	± 0.012
classRF (def)	0.55***	0.527	0.579	± 0.02
M5	0.554***	0.523	0.584	± 0.019
CART	0.529***	0.505	0.544	± 0.012
k-NN	0.548***	0.529	0.564	± 0.014
b-NN	0.541***	0.505	0.564	± 0.024
logit	0.557***	0.535	0.583	± 0.016

Table VI
AVERAGE CLASSIFICATION PERFORMANCE WITH CLASS LABEL AS REGRESSAND (***: p VALUE (95 % CONFIDENCE INTERVAL) OF ONE-SIDED WILCOXON TEST, AUC PREDICTION VS. GUESSING, I.E. $\mu = 0.5, p < 0.001$)

machines in section IV-A. Goodness of fit of the probability estimates and classification performance (as *AUC*) are even weaker, however. Only the consistent *regRF* and the two nearest neighbour approaches achieve a $nrmse\% < 100$.

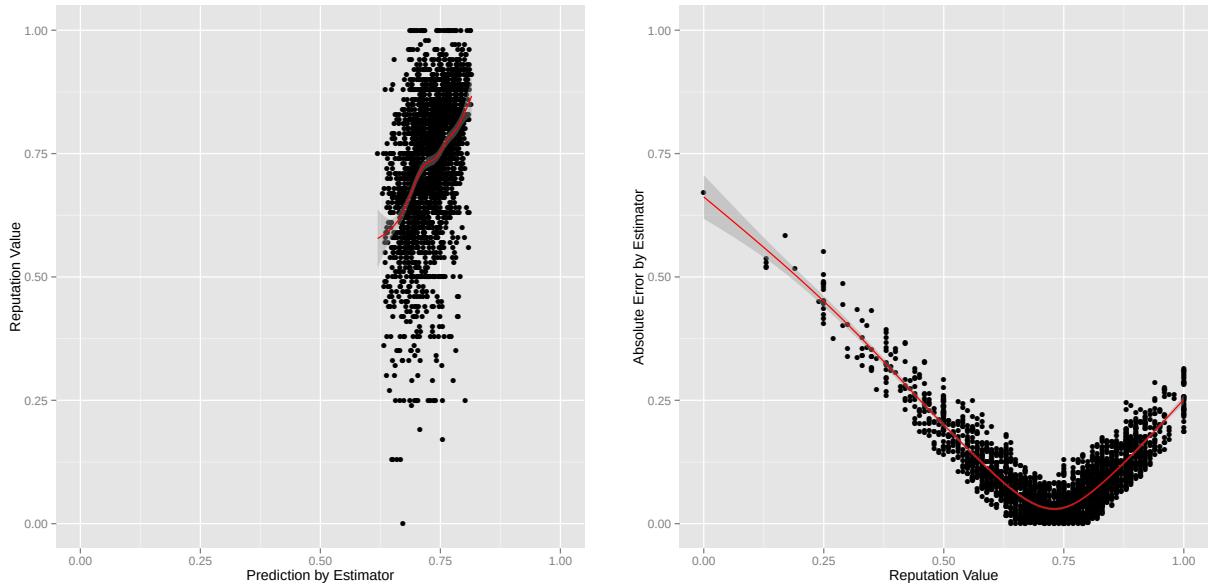
Figure 2 shows the predictive performance and absolute error of the best performing (in terms of *AUC*) estimator, a consistent *regRF* trained on recommendation score regressands. The distribution of the prediction versus the actual recommendation score and the distribution of the error indicate the limited ability of the estimator to create a good prediction model. Predictions are centred around the mean recommendation score, thereby decreasing the goodness of the prediction the further the actual recommendation score deviates from this mean. Majority class undersampling was performed to check if this was solely induced by the distribution of the recommendation score. This did not lead to improved performance.

V. DISCUSSION

The dataset presented in section III illustrates peculiarities that are caused by the presence of reputation systems in service selection. The data exhibits a strong prevalence of positive ratings over negative ones (figures 1(a) and 1(b)). Assuming that ratings are, for the most part, authentic, we attribute this to two main reasons.

First, the type of service provided is physical in nature, rather than virtual, and has a long and established tradition, and is well-regulated by social norms, as well as economic and legal bodies. Thus, providing a service as advertised is strongly encouraged by the environment of service provisioning. At the same time, there are established expectations what a customer can expect from the service provider/hotelier, leading to positive expectation confirmation. Simply put, providing a physical service as advertised is simply the social and legal norm, while at the same time the customer knows what to expect from a 3-star hotel at a given price point.

Second, and more interestingly from a data-centric perspective, is a tendency towards preferential attachment that is visible from the data. Considering figures 1(c) and 1(d), we can observe *a*) a long-tailed distribution signifying that only a small number of hotels have many ratings, reminiscent of a power law distribution; and *b*) high numbers of ratings are considerably more frequent among hotels with higher recommendations scores. Because hotels with good ratings are preferentially selected – as a risk minimisation strategy – and because hotels with a good rating can be considered to be more likely to provide satisfactory service, reputation systems contribute to the skewed distribution observable



(a) Prediction

(b) Absolute Error

Figure 2. Predictive Performance and Error for Regression Random Forest (regRF, consistent, ntree=10%)

from the data. By design, reputation systems dissolve the independence of service selection and feedback – a fact that is both indicative of and contributes heavily to their success as a soft security control instrument. Well-behaving providers are rewarded by building a good reputation and attracting more customers, while badly performing providers are effectively eliminated from the selection process.

Thus, the dataset reflects the success of a functioning reputation system in a real-world application scenario, in which transaction costs are non-negligible. At the same time, however, the effects of preferential attachment that are driven by the reputation system also pose challenges. Exploitative service selection is encouraged over explorative selection, which leads to established markets and market entry issues for new hotels. Not only that, but because *presumably* bad providers are very quickly eliminated from the market, by not being selected, *and* because these presumably bad providers only have a low number of bad ratings, feature-based trustworthiness prediction methods are limited in their effectiveness. The number of negative samples is simply not sufficient to build accurate predictive models.

Consequently, the features presented on the selected hotel booking website encode a hotels trustworthiness to a very limited degree. This limits the usefulness of stereotyping approaches in service selection scenarios, because the data foundation that is used for machine learning is necessarily skewed by the selection process. However, the performance of the regression machines is significantly (tables IV, VI) better than pure guessing and therefore can (and should) be harnessed.

A. Belief Logic Mapping

The goodness-of-fit of the supervised estimators evaluated in section IV does *not* warrant building a standalone trust management system around them. The features of the hotel dataset do not provide sufficient discriminatory power to build accurate models from the skewed data and do not yield reliable trust scores. However, the probability machines still provide if not an accurate fit of the trustworthiness, then at least an *indication* of how trustworthy a particular hotel is. As such, they can still be of value within a trustworthiness estimation *ensemble*. They can be used in a supplementary role, for instance as input to the base rate or initial expectation of an experience-based Bayesian model.

The meaningful combination of different trustworthiness estimators and the logical inference over their output require a framework for reasoning. *Subjective Logic* [8] is a popular choice for reasoning under uncertainty that is inherent in the estimation process. A more recent but similar framework is *CertainLogic* [18], which is derived from and fully isomorphic to *Subjective Logic*.

We model the integration of trust estimating regression machines with other estimators, e.g., reputation-based trust models, using *CertainLogic*. This choice is governed primarily by the fact that the opinion representation of *CertainLogic* corresponds more intuitively to the outputs and error estimates of the regression machines. Choosing *CertainLogic* over *Subjective Logic* should not be understood as a reflection on the capabilities of each; rather, we believe that using the *CertainLogic* opinion representation will ease understanding.

CertainLogic is derived from *Subjective Logic* and is therefore rooted in belief theory [19]. As such, it allows not only for the modelling, combination and inference over probabilities, but over so-called *opinions*. Opinions allow expressing any possible *uncertainty* regarding the probabilities. Ries et al. [20] propose to represent opinions as ordered triples $\omega = (t, c, f)$, where:

- $t \in [0; 1]$ is a *probability estimate* that $y = 1$.
- $c \in [0; 1]$ is a *certainty estimate* that the probability estimate t is correct.
- $f \in [0; 1]$ is a *base rate*, modelling an a-priori assumption.

and $t, c, f \in [0; 1]$.

In experience-based naive Bayesian trustworthiness prediction, such as [9], [5], the probability estimate t corresponds to the proportion of good ratings ($y = 1$) to all ratings a truster has with regard to a specific trustee. The certainty estimate c is typically a function of the number of such ratings. Establishing the certainty estimate in this manner is made possible by leveraging model assumptions of the naive Bayesian prediction, in particular the convergence of the mean to the true expectation value.

When using regression machines, mapping the probability estimate t is trivially achieved by using the prediction value, as in naive Bayesian models. Certainty estimation, however, has to be done in a different manner, due to different characteristics and purposes of the prediction paradigm.

We propose using a conservative goodness-of-fit measure, for instance the *normalised root-mean-square error* ($nrmse$):

Definition 2: Let $O = (o_1, o_2, \ldots, o_n)$ be a vector of observed values and $\hat{S} = (\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n)$ a vector of corresponding estimates. Let o_{max} be the largest, o_{min} the smallest element of O.

$nrmse = \left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{s}_i - o_i)^2}\right) / (o_{max} - o_{min})$

The mapping from estimator output to the *CertainLogic* opinion space is thus given as:

- $t = P(y = 1|\mathbf{x})$
- $c = 1 - (min(nrmse, 1))$
- $f \in [0; 1]$, a (user defined) default base rate to be used under complete uncertainty, e.g., 0.5.

The provided mapping enables the integration of regression machines in trustworthiness assessment ensembles. Using different fusion operators (cf., [21], [22]), different estimation paradigms can be flexibly combined, thereby enabling ensembles that can leverage the respective strengths of the different estimation paradigms.

VI. Related Work

As this paper concerns itself with the application of a subclass of supervised learning mechanisms to trustworthiness assessment for reputation-based trust management, this section will quite briefly present some related work in the field of computational trust.

Research on computational trust is an active and maturing field of research (cf., for instance, [8], [23]), a fact that is well attested by the proliferation of related concepts in literature and application. Bayesian reputation-based trust models, such as work by Jøsang [9] or Ries [5], have provided statistically well-founded trust models. As reinforcement learning approaches (that do not generalise), these models in particular have to cope with *bootstrapping* and *cold-start* issues, leading to an increased impact of the *exploration-vs.-exploitation dilemma*.

Recent research on trust bootstrapping via stereotyping by Burnett et al. [3], [24], Liu et al. [4] and Fang et al. [25] is directed at providing a better generalisation ability by creating stereotypical profiles for generalisation in agent societies. Through their use of machine learning and data mining, this research can be considered closely related to the work presented in this paper. In particular, decision trees are used for *classification* from feature vectors. However, the focus of our work is on consistent, model-free prediction for probability estimation.

In the field of recommendation systems, a rich landscape of literature exists that deals with the application of machine learning (for an introduction, see, e.g. [26]). Recommendations in tourisms, for instance, for hotels, are a popular application scenario (e.g., [27], [28]).

VII. Concluding Remarks

In the previous sections we have outlined the requirements for the application of supervised machine learning methods, so-called regression machines, to trustworthiness assessment. We have shown the impact, on a real-world dataset, of exploitative selection on the data generation process and how this affects predictive performance. Finally, we have provided a mapping from estimator output to a belief logic representation that enables the use even of weak predictive results within the framework of trust assessment ensembles.

Using reputation systems in service selection, particularly when non-negligible resources are at stake, reinforces a trend towards exploitation. This has effects on the data that is generated and available for future trust assessment. The resulting complex adaptive system of trust assessment, selection and data generation merits closer attention in the future. For this, a data-centric, rather than a model-centric, approach to investigating the dynamics of trust and reputation systems is necessary. Developing flexible, component-based trust management approaches, standardised evaluation methodologies and a systematic collection and analysis of trust related datasets, in the form of a publicly available reference library for testing, are, in our opinion, important next steps.

Acknowledgment

The work presented in this paper was performed in the context of the Software-Cluster project *EMERGENT* and the Software Campus project *MoVe4Dynamic*; it was funded by

the German Federal Ministry of Education and Research (BMBF) under grants no. "$01|C10S01$" and "$01|S12054$". The authors assume responsibility for the content.

REFERENCES

[1] C. Castelfranchi and R. Falcone, "Trust is much more than Subjective Probability: Mental Components and Sources of Trust," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.

[2] D. Gambetta, "Can We Trust Trust?" in *Trust: Making and Breaking Cooperative Relations*, D. Gambetta, Ed. Oxford: Basil Blackwell, 1988, pp. 213–237.

[3] C. Burnett, T. Norman, and K. Sycara, "Bootstrapping Trust Evaluations through Stereotypes," in *Proceedings. of 9th International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 241–248.

[4] X. Liu, A. Datta, K. Rzadca, and E. Lim, "Stereotrust: A Group Based Personalized Trust Model," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 7–16.

[5] S. Ries, "Extending Bayesian Trust Models Regarding Context-Dependence and User Friendly Representation," in *Proceedings of the 2009 ACM Symposium on Applied Computing*. New York, New York, USA: ACM, 2009, pp. 1294–1301.

[6] E. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed., ser. Springer Texts in Statistics. Springer, 1998, vol. XXVI.

[7] J. Malley, J. Kruppa, A. Dasgupta, K. Malley, and A. Ziegler, "Probability Machines: Consistent Probability Estimation Using Nonparametric Learning Machines," *Methods Inf Med*, vol. 51(1), pp. 74–81, 2012.

[8] A. Jøsang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems*, vol. 43 (2), pp. 618–644, 2007.

[9] A. Jøsang and R. Ismail, "The Beta Reputation System," in *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[10] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[11] G. Biau and L. Devroye, "On the Layered Nearest Neighbour Estimate, the Bagged Nearest Neighbor Estimate and the Random Forest Method in Regression and Classification," *Journal of Multivariate Analysis*, vol. 101, pp. 2499–2518, 2010.

[12] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees," Wadsworth & Brooks/Cole Advanced Books and Software, Tech. Rep., 1984.

[13] J. Quinlan, "Learning with continuous classes," in *Proceedings AI*, 1992, pp. 343–348.

[14] G. Biau, L. Devroye, and G. Lugosi, "Consistency of Random Forests and Other Averaging Classifiers," *Journal of Machine Learning Research*, vol. 9, pp. 2015–2033, 2008.

[15] L. Breiman, "Bagging Predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[16] X. Liu, G. Trédan, and A. Datta, "A Generic Trust Framework for Large-Scale Open Systems Using Machine Learning," *CoRR*, vol. abs/1103.0086, 2011.

[17] M. Zambrano-Bigiarini, *R-Package hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series*, 10 2012. [Online]. Available: http://cran.r-project.org/web/packages/hydroGOF/hydroGOF.pdf

[18] S. Ries, S. M. Habib, M. Mühlhäuser, and V. Varadharajan, "CertainLogic: A Logic for Modeling Trust and Uncertainty (Short Paper)," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST 2011)*. Springer, Jun 2011.

[19] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[20] S. Ries, "Trust in Ubiquitous Computing," Doctoral Thesis, TU Darmstadt, 2009.

[21] S. Habib, S. Ries, S. Hauke, and M. Mühlhäuser, "Fusion of Opinions under Uncertainty and Conflict–Trust Assessment for Cloud Marketplaces," in *Proceedings of IEEE TrustCom-12*, 2012.

[22] A. Jøsang, "Probabilistic Logic Under Uncertainty," in *Proceedings of Computing: The Australian Theory Symposium (CATS'07)*, 2007.

[23] Y. Wang and J. Vassileva, "Toward Trust and Reputation Based Web Service Selection: A Survey," *International Transactions on Systems Science and Applications*, vol. 3, no. 2, pp. 118–132, 2007.

[24] C. Burnett, T. Norman, and K. Sycara, "Sources of Stereotypical Trust in Multi-Agent Systems," in *Proceedings of the 14th International Workshop on Trust in Agent Societies*, 2011, p. 25.

[25] H. Fang, J. Zhang, M. Sensoy, and N. Thalmann, "A Generalized Stereotypical Trust Model," in *IEEE TrustCom-11*. IEEE, 2012, pp. 698–705.

[26] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems – An Introduction*. Cambridge University Press, 2010.

[27] D. Jannach, M. Zanker, and M. Fuchs, "Constraint-Based Recommendation in Tourism: A Multiperspective Case Study," *International Journal of Information Technology and Tourism*, vol. 11(2), pp. 139–155, 2009.

[28] D. Jannach, F. Gedikli, Z. Karakaya, and O. Juwig, "Recommending Hotels Based on Multi-Dimensional Customer Ratings," *Information and Communication Technologies in Tourism 2012*, pp. 320–331, 2012.

APPENDIX

Classification performance of different estimators for dichotomised lass labels of the hotel data set. For each hotel, a new dichotomous response variable $y \in 0; 1$ was computed by using a binomial random number generator with the hotels recommendation score as the corresponding probability.

A. Regression to Reputation Score, Undersampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Additionally, during the training process, the majority class $y = 1$ was undersampled, so that $|y = 1| == |y = 0|$. Results were generated from cross validated bootstrap samples.

	avg AUC	**MIN**	**MAX**	**± SD**
regRF (consistent)	0.575	0.543	0.599	± 0.017
regRF (default)	0.578	0.549	0.6	± 0.015
M5	0.571	0.544	0.588	± 0.017
CART	0.56	0.538	0.577	± 0.014
logit	0.574	0.546	0.599	± 0.016

Table VII
AVERAGE CLASSIFICATION PERFORMANCE FOR UNDERSAMPLED DATA AND RECOMMENDATION SCORE AS REGRESSAND

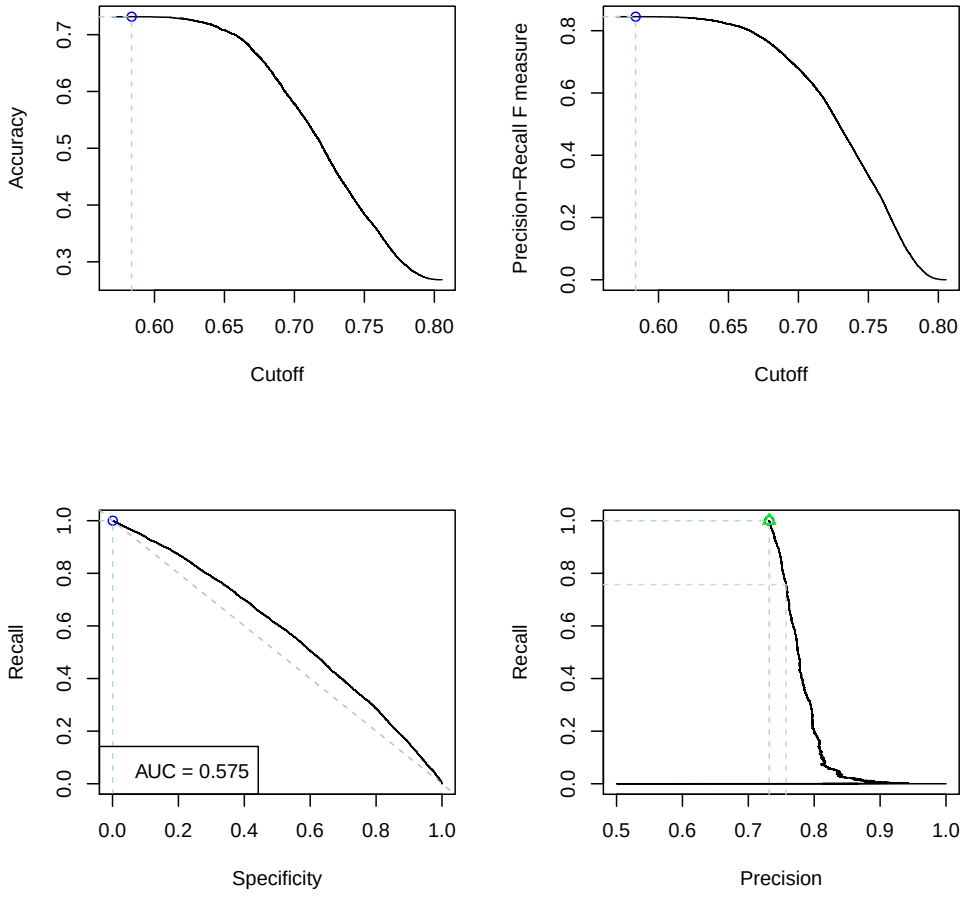


Figure 3. Regression Random Forest, ntree = 10% of number of samples (hotels), consistent

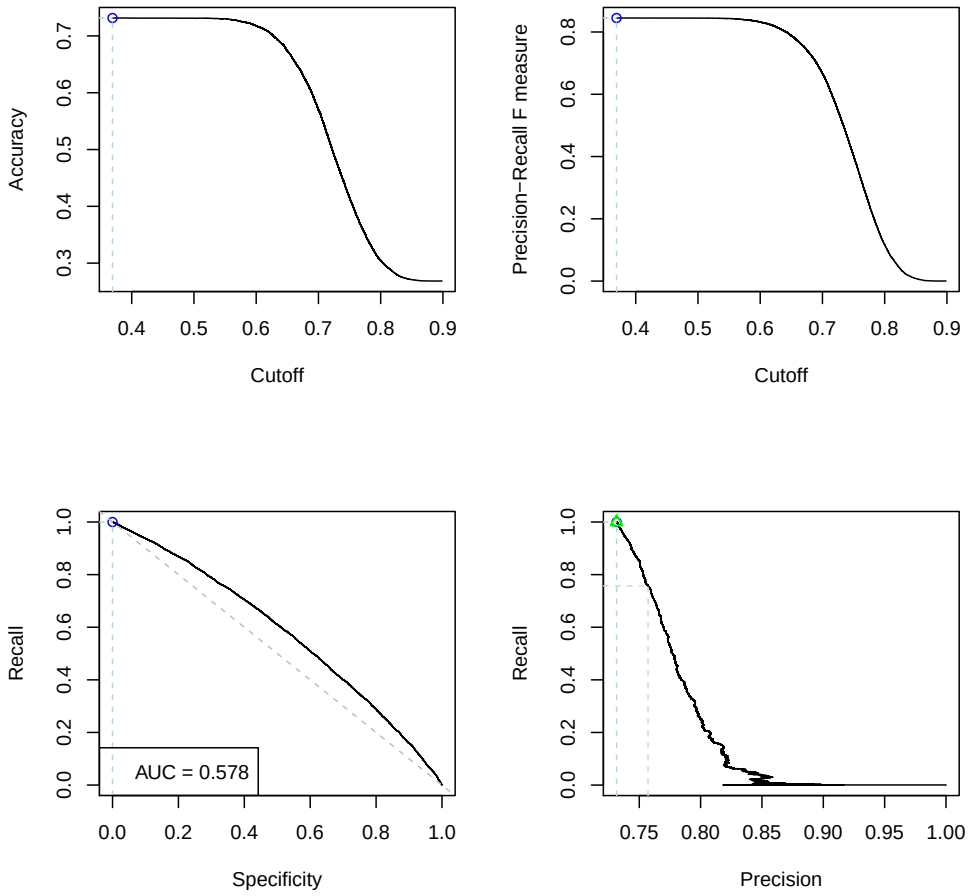


Figure 4. Regression Random Forest, ntree = 5, package default

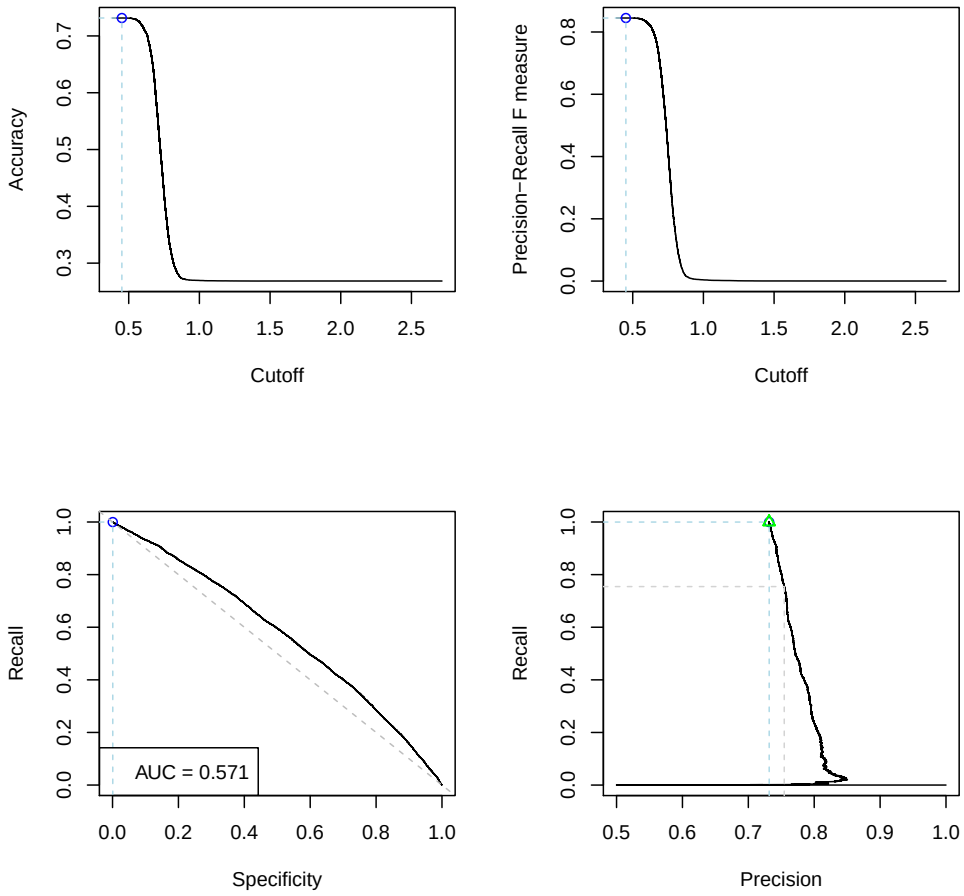


Figure 5. M5 Model Decision Tree

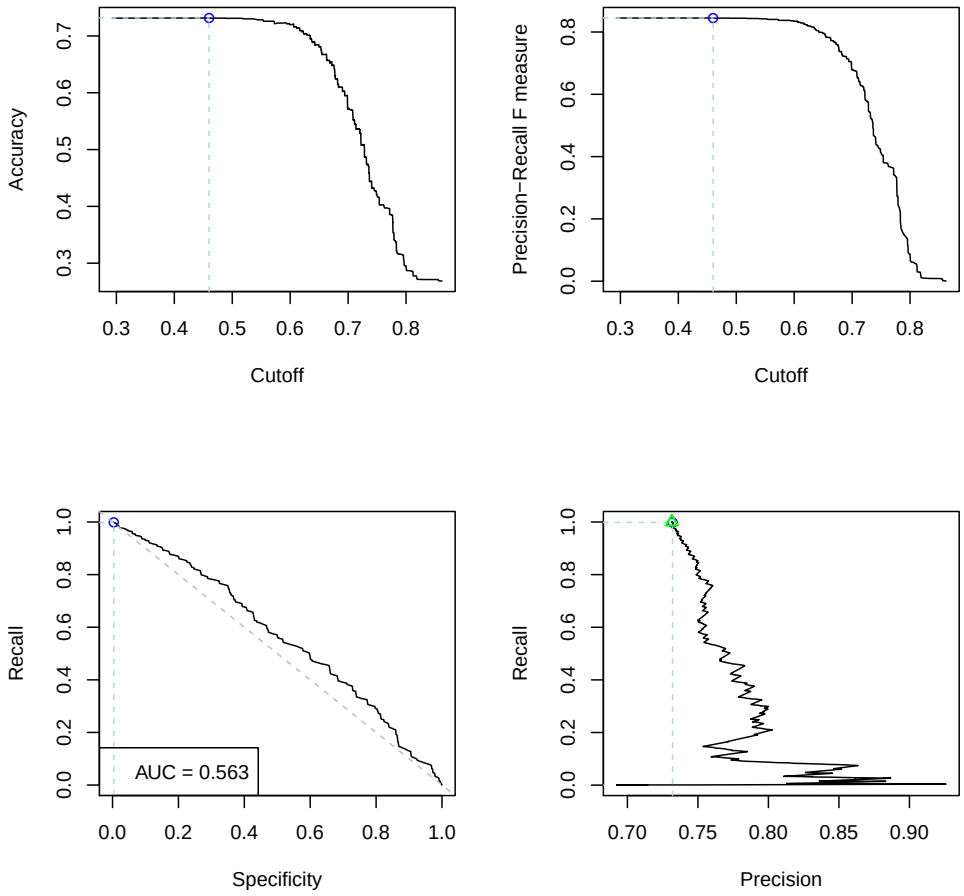


Figure 6. Classification and Regression Tree (CART)

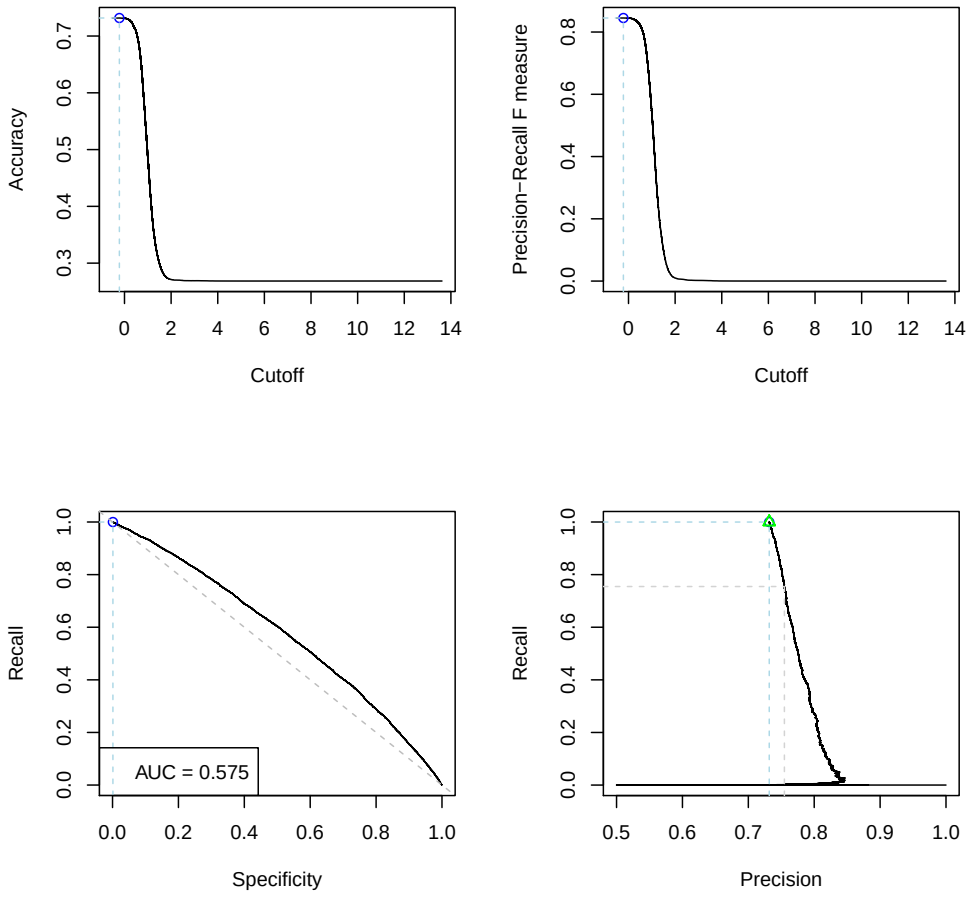


Figure 7. Logistic Regression

B. Regression to Class Label, Undersampling

Classification performance of estimators trained on hotel label, i.e., regressor is dichotomous variable in $\{0; 1\}$. Additionally, during the training process, the majority class $y = 1$ was undersampled, so that $|y = 1| == |y = 0|$. Results were generated from cross validated bootstrap samples.

	avg AUC	**MIN**	**MAX**	$\pm$ **SD**
regRF (consistent)	0.552	0.509	0.583	$\pm$ 0.026
regRF (default)	0.542	0.507	0.564	$\pm$ 0.021
classRF (consistent)	0.556	0.521	0.59	$\pm$ 0.023
classRF (default)	0.541	0.502	0.563	$\pm$ 0.022
M5	0.546	0.507	0.574	$\pm$ 0.02
CART	0.621	0.503	0.809	$\pm$ 0.108
logit	0.546	0.503	0.573	$\pm$ 0.024

Table VIII
AVERAGE CLASSIFICATION PERFORMANCE FOR UNDERSAMPLED DATA AND CLASS LABEL AS REGRESSAND

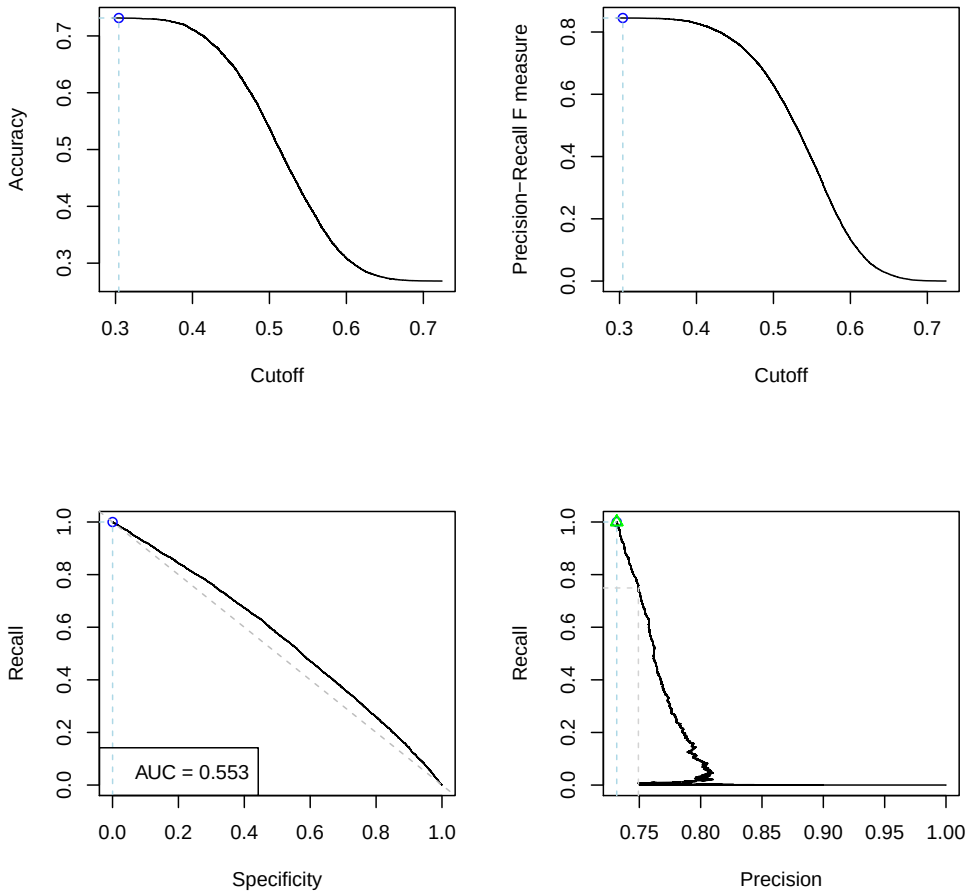


Figure 8. Regression Random Forest, ntree = 10% of number of samples (hotels), consistent

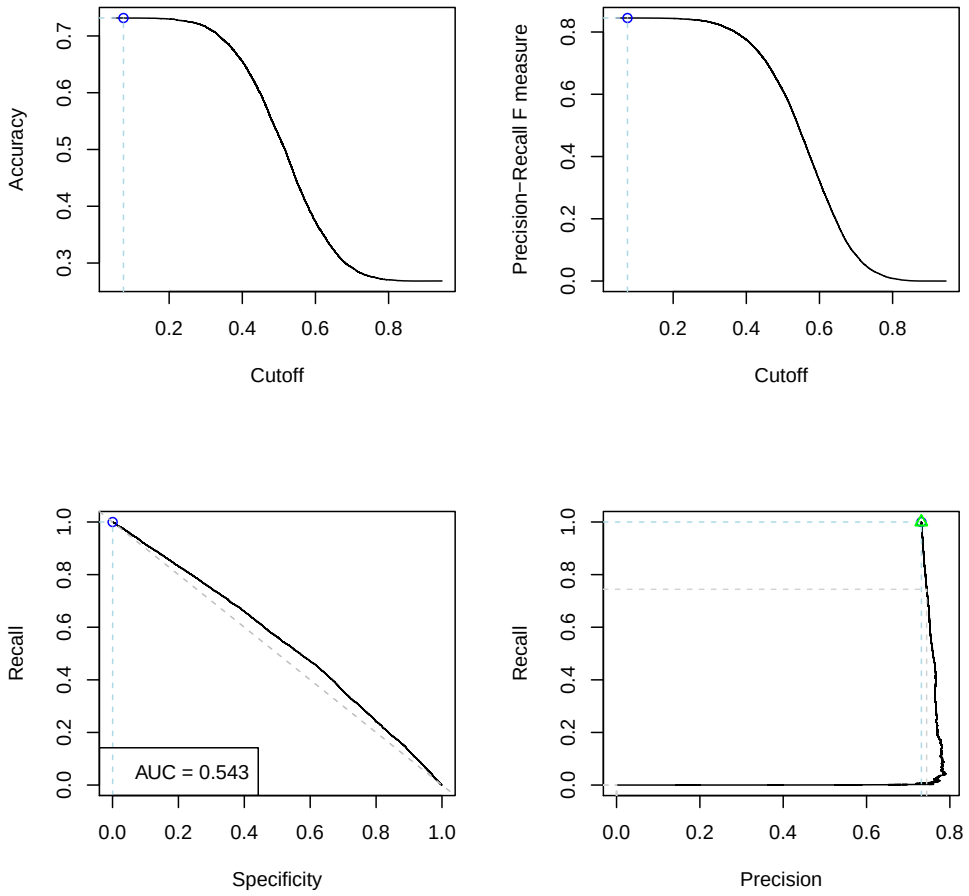


Figure 9. Regression Random Forest, ntree = 5, package default

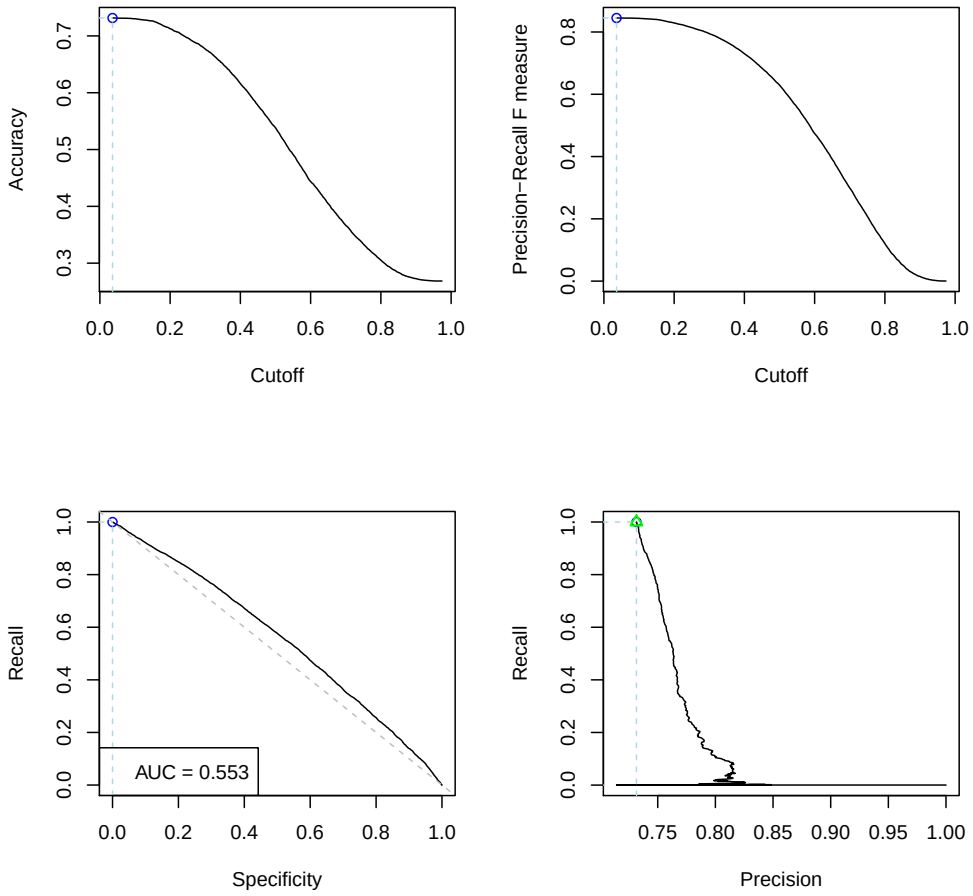


Figure 10. Classification Random Forest, ntree = 10% of number of samples (hotels), consistent

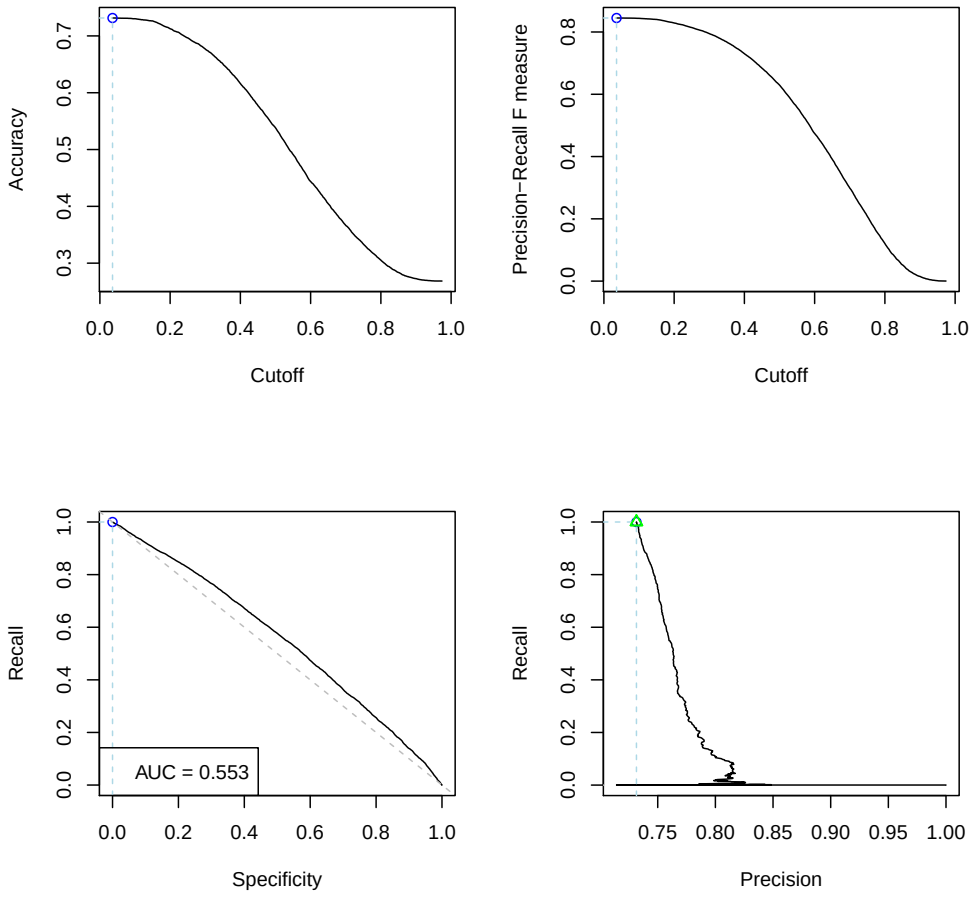


Figure 11. Classification Random Forest, ntree = 1, package default

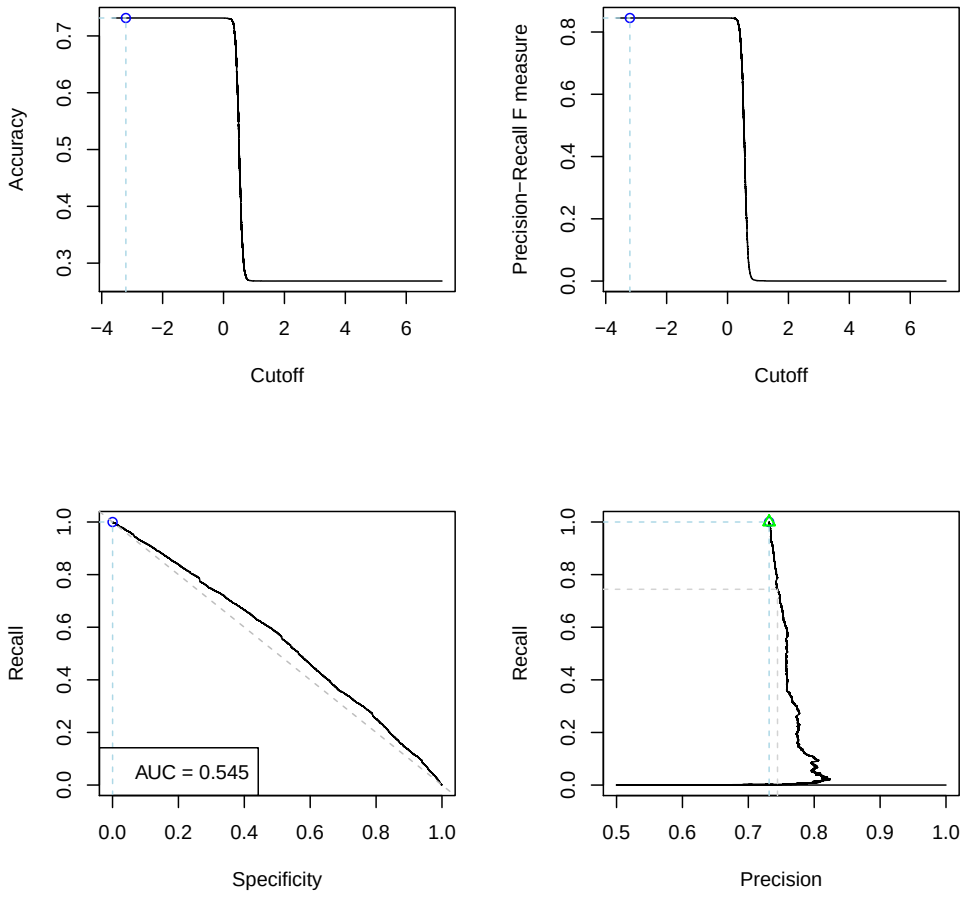


Figure 12. M5 Model Decision Tree

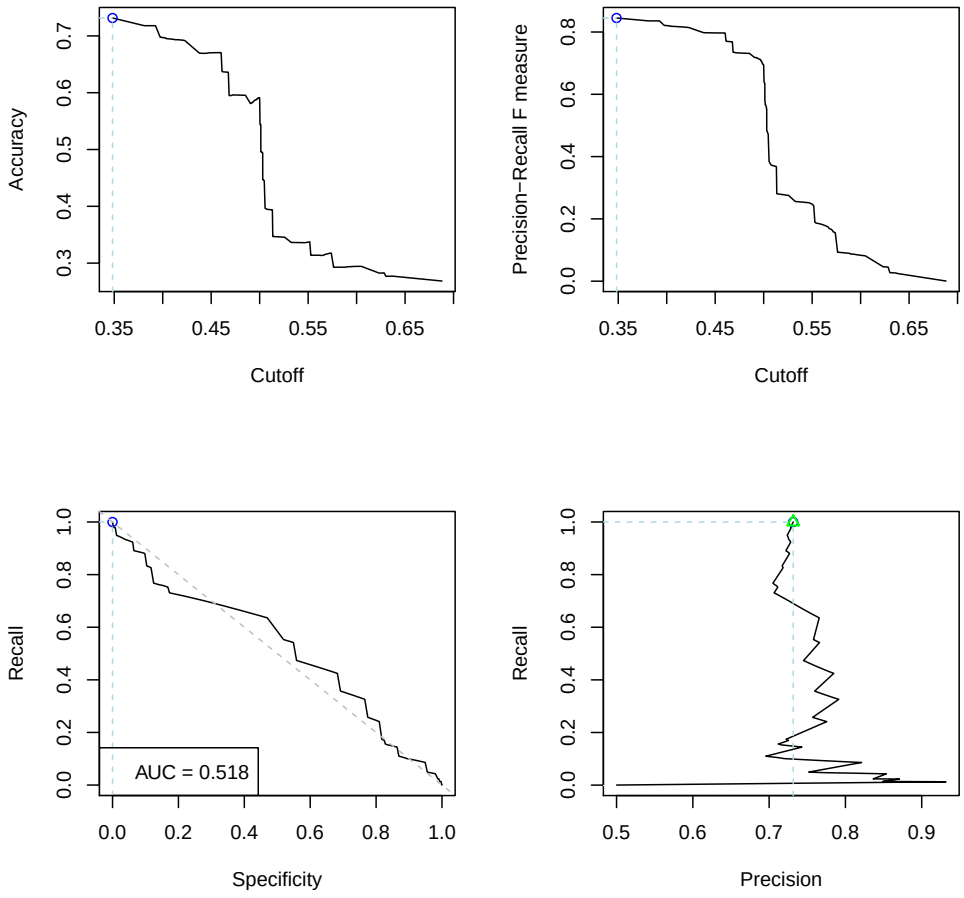


Figure 13. Classification and Regression Tree (CART)

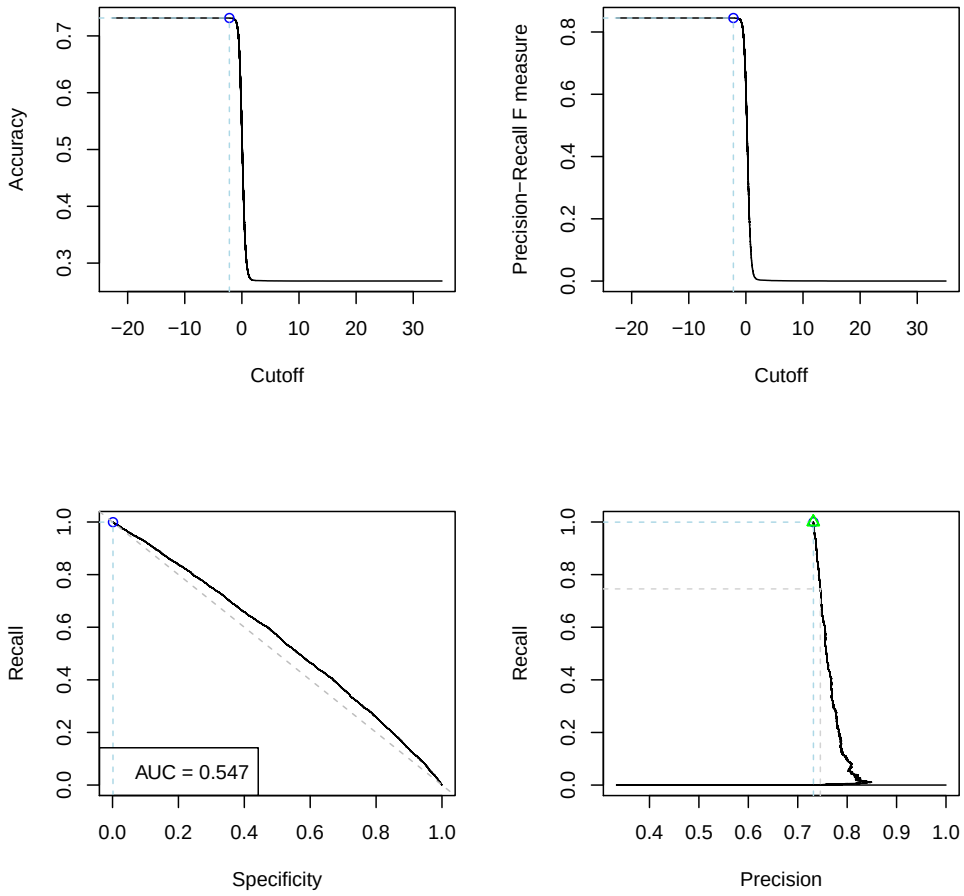


Figure 14. Logistic Regression

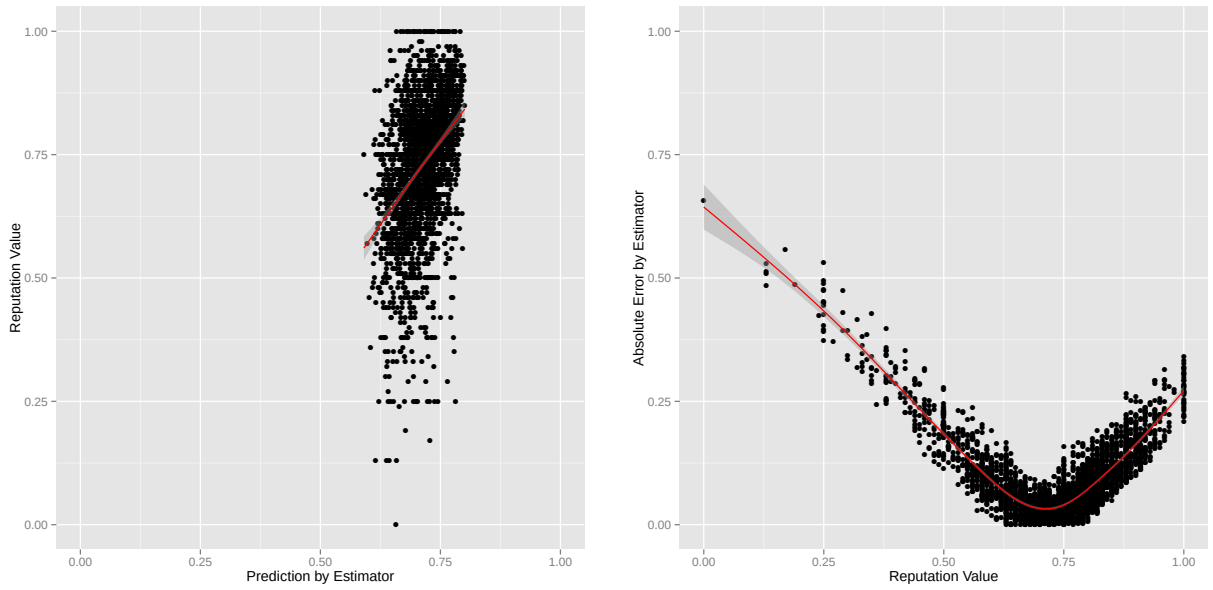
Predictive performance of regression and probability machines. Performance measured based on the error function between predicted value $\hat{Y}$ and reputation score Y.

C. Regression to Reputation Score, Undersampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Additionally, during the training process, the majority class $y = 1$ was under sampled, so that $|y = 1| == |y = 0|$. Results were generated from cross validated bootstrap samples.

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	-0.02	0.1	0.02	0.13	93.3	-2.7	0.93	0.31	0.13	0.07	0.45	0.31
regRF (default)	-0.02	0.1	0.02	0.13	92.6	-3.3	0.93	0.45	0.14	0.08	0.55	0.39
M5	-0.02	0.1	0.02	0.13	96.3	-2.5	0.96	0.51	0.07	0.07	0.52	0.38
CART	-0.02	0.1	0.02	0.13	95.9	-2.6	0.96	0.45	0.08	0.05	0.49	0.35
logit	0.2	0.28	0.14	0.38	275.2	26.8	2.75	2.53	-6.57	-1.66	0.42	0.3

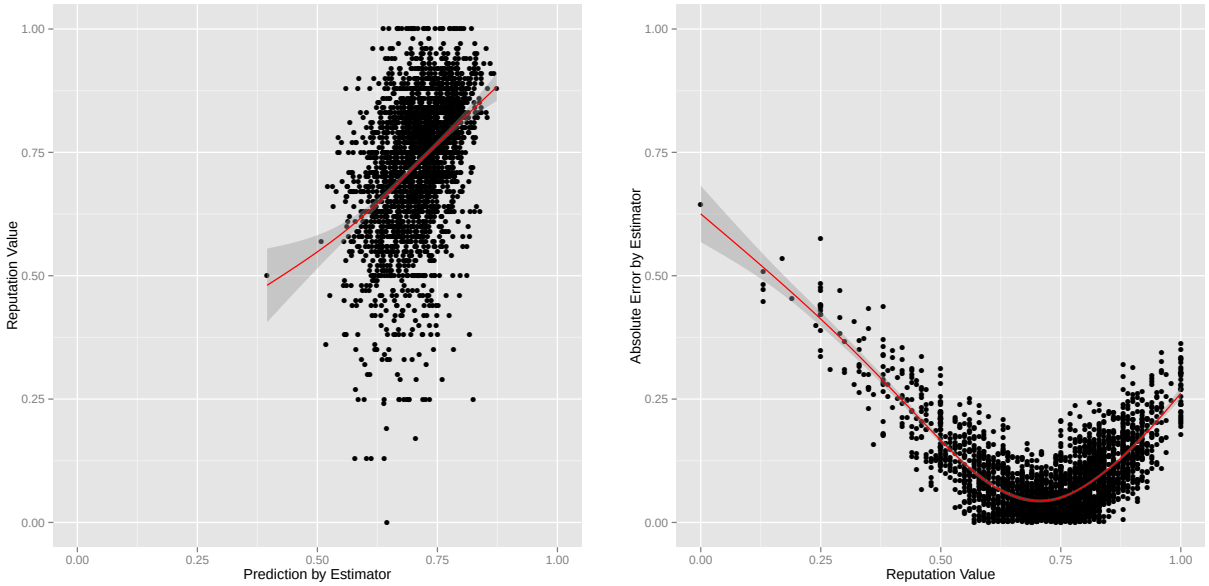
Table IX
AVERAGE GOODNESS OF FIT FOR UNBALANCED DATA AND RECOMMENDATION SCORE AS REGRESSAND



(a) Prediction

(b) Absolute Error

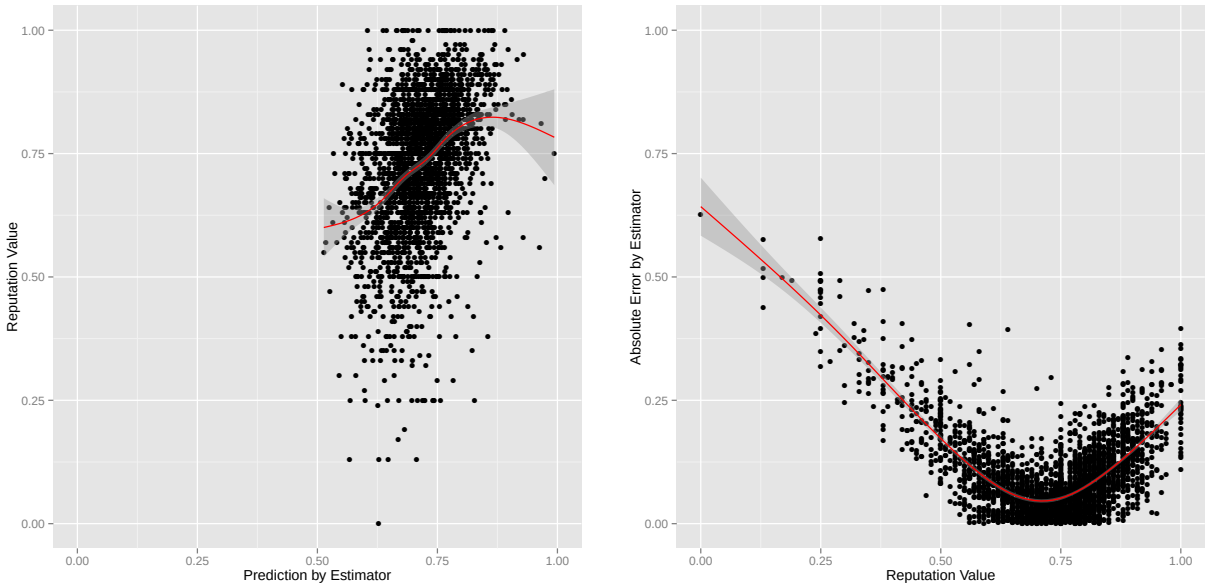
Figure 15. Predictive Performance and Error for Regression Random Forest (regRF, consistent, ntree=10%)



(a) Prediction

(b) Absolute Error

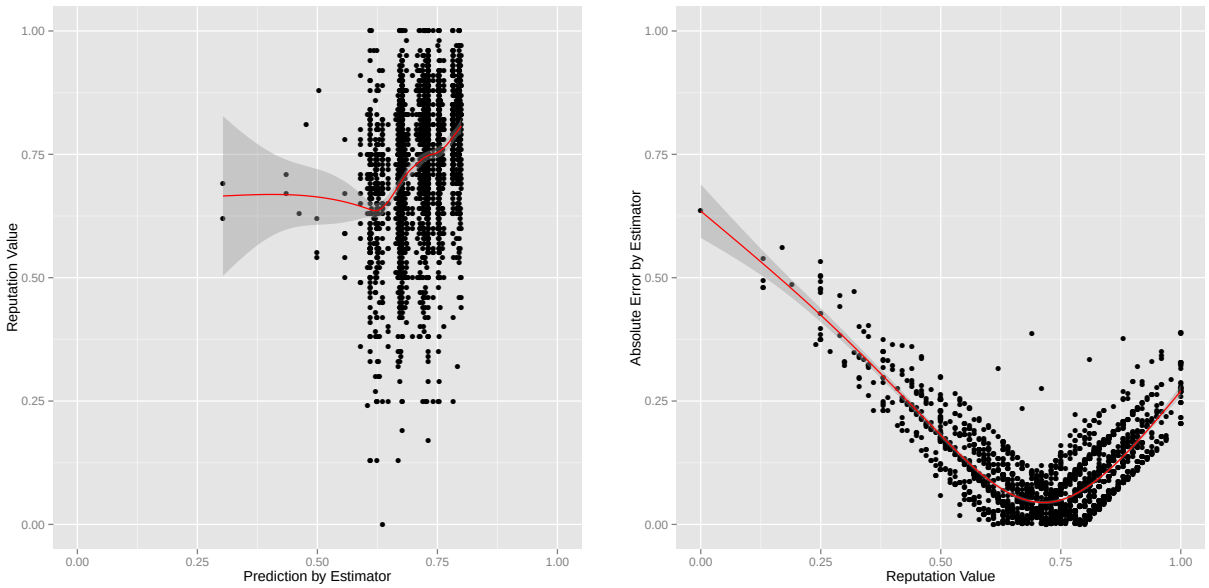
Figure 16. Predictive Performance and Error for Regression Random Forest (regRF, default, ntree=5)



(a) Prediction

(b) Absolute Error

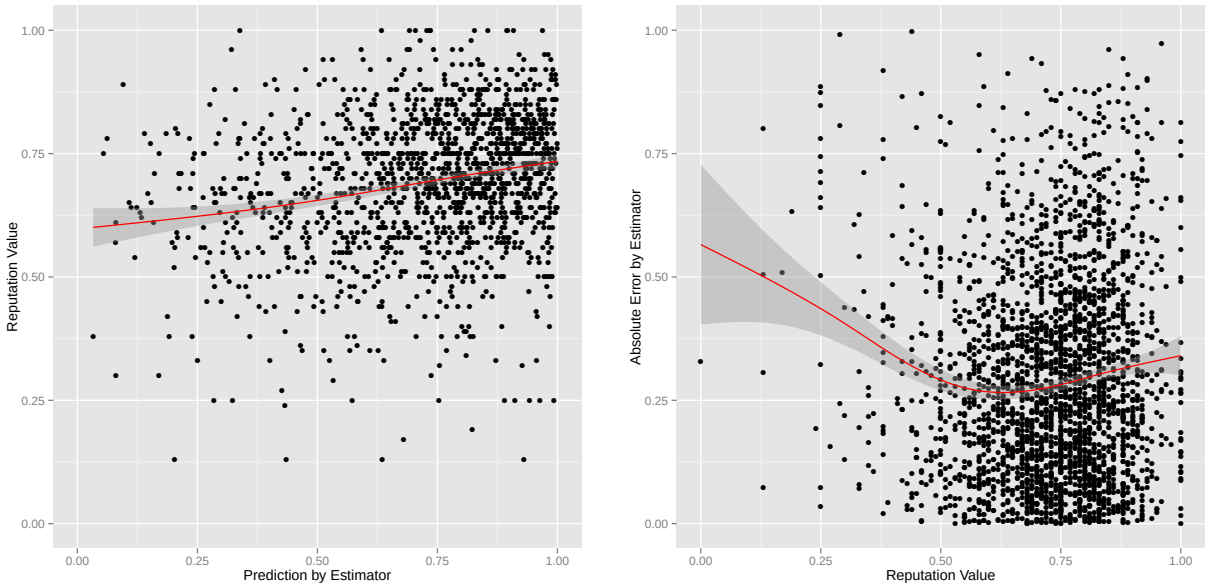
Figure 17. Predictive Performance and Error for M5 Model Decision Tree



(a) Prediction

(b) Absolute Error

Figure 18. Predictive Performance and Error for Classification and Regression Tree (CART)



(a) Prediction

(b) Absolute Error

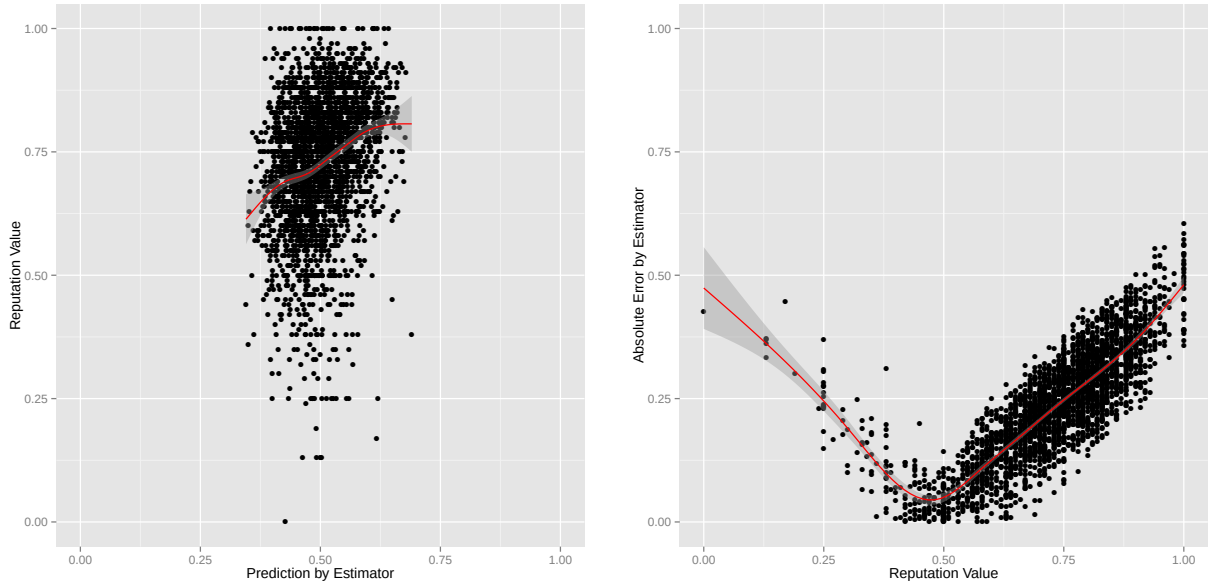
Figure 19. Predictive Performance and Error for Logistic Regression (logit)

D. Regression to Class Label, Undersampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Additionally, during the training process, the majority class $y = 1$ was undersampled, so that $|y = 1| == |y = 0|$. Results were generated from cross validated bootstrap samples.

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	-0.23	0.24	0.07	0.26	190.4	-31	1.9	0.47	-2.63	-1.23	0.44	0.29
regRF (default)	-0.23	0.25	0.08	0.28	202.8	-31.6	2.03	0.84	-3.11	-1.3	0.43	0.28
classRF (consistent)	-0.22	0.25	0.09	0.3	215.3	-30.7	2.15	1.36	-3.63	-1.32	0.44	0.29
classRF (default)	-0.23	0.24	0.08	0.27	199.2	-31.1	1.99	0.8	-2.97	-1.26	0.43	0.28
M5	-0.23	0.24	0.08	0.28	200.6	-30.9	2.01	0.83	-3.02	-1.26	0.42	0.28
CART	-0.23	0.25	0.08	0.28	200.3	-31.6	2	0.35	-3.01	-1.3	0.38	0.27
logit	-0.71	0.76	0.94	0.97	704.3	-96.8	7.04	4.92	-48.61	-6.14	0.18	0.13

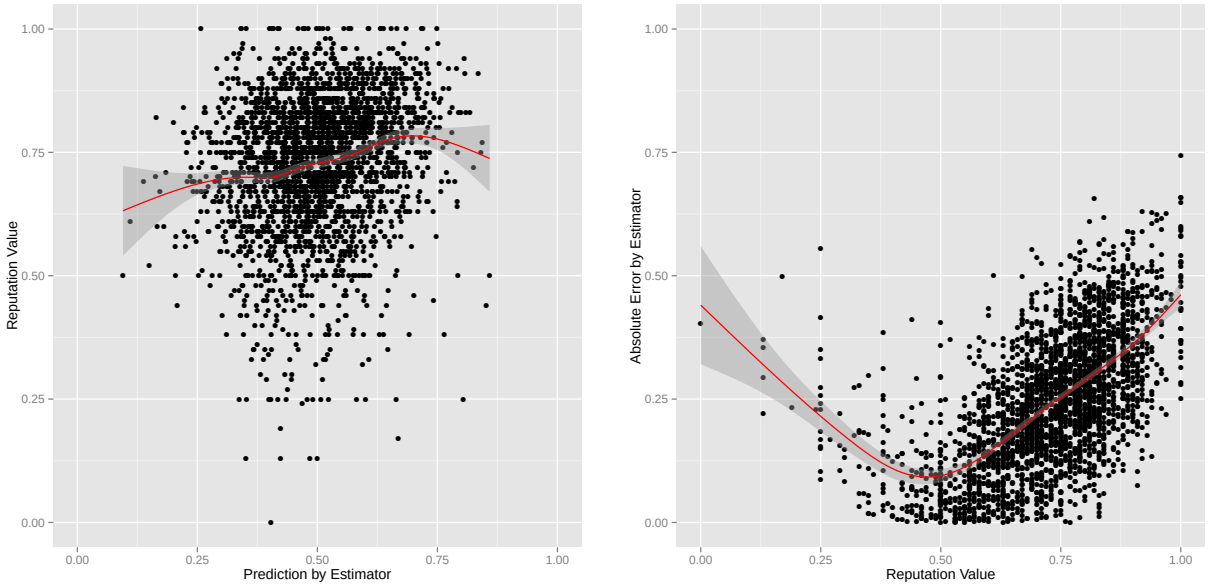
Table X
AVERAGE GOODNESS OF FIT FOR UNDERSAMPLED DATA AND CLASS LABEL AS REGRESSAND



(a) Prediction

(b) Absolute Error

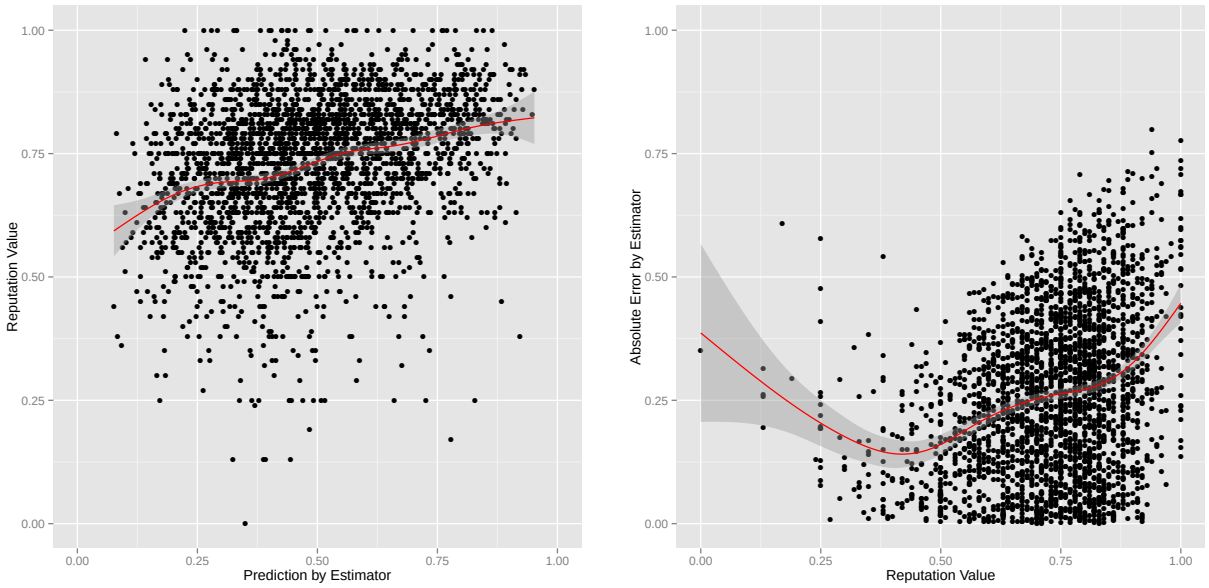
Figure 20. Predictive Performance and Error for Regression Random Forest (regRF, consistent, ntree=10%)



(a) Prediction

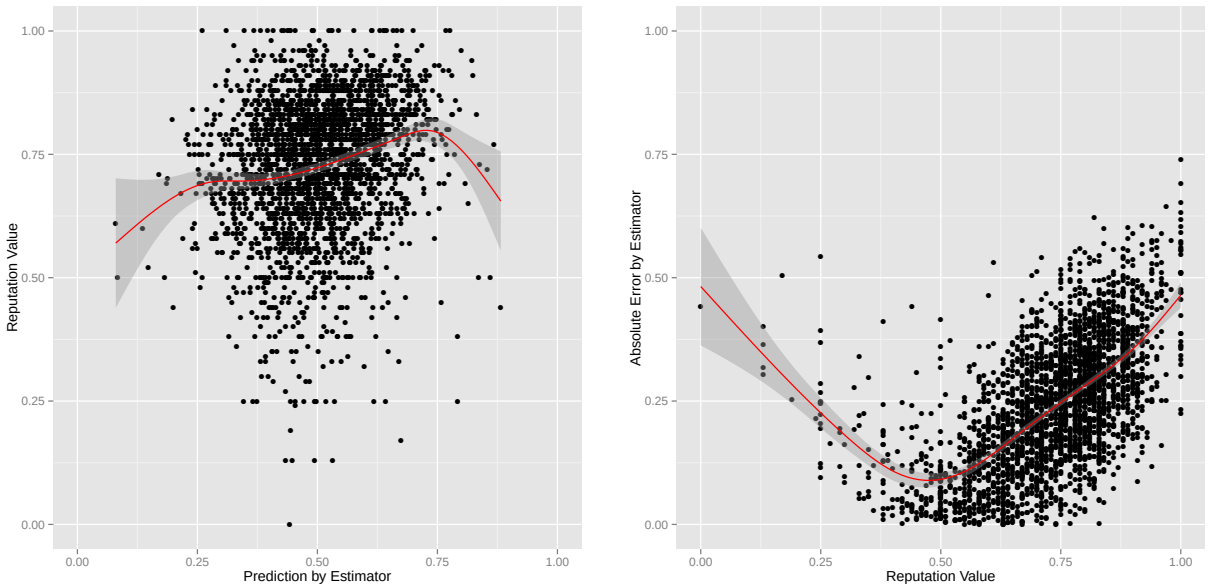
(b) Absolute Error

Figure 21. Predictive Performance and Error for Regression Random Forest (regRF, default, ntree=5)



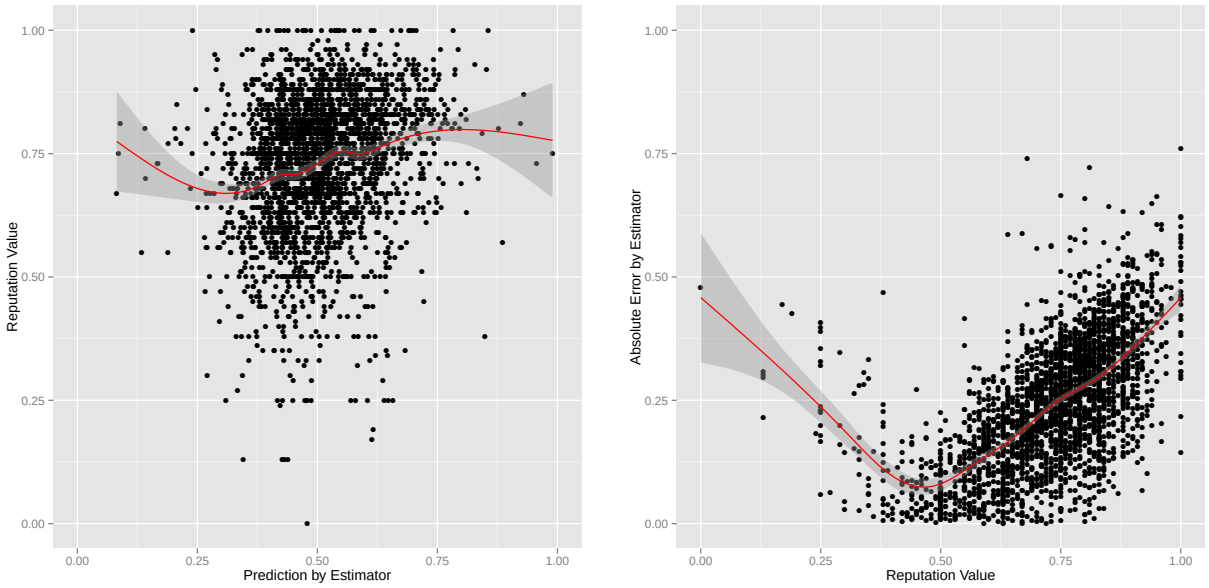
(a) Prediction (b) Absolute Error

Figure 22. Predictive Performance and Error for Classification Random Forest (classRF, consistent, ntree=10%)



(a) Prediction (b) Absolute Error

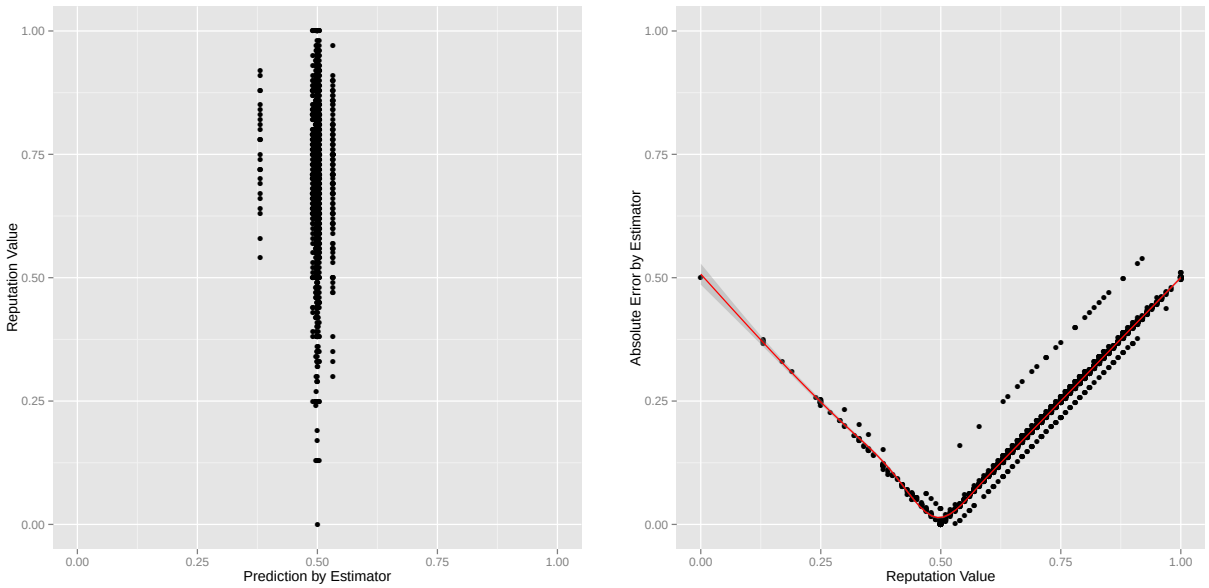
Figure 23. Predictive Performance and Error for Classification Random Forest (classRF, default, ntree=1)



(a) Prediction

(b) Absolute Error

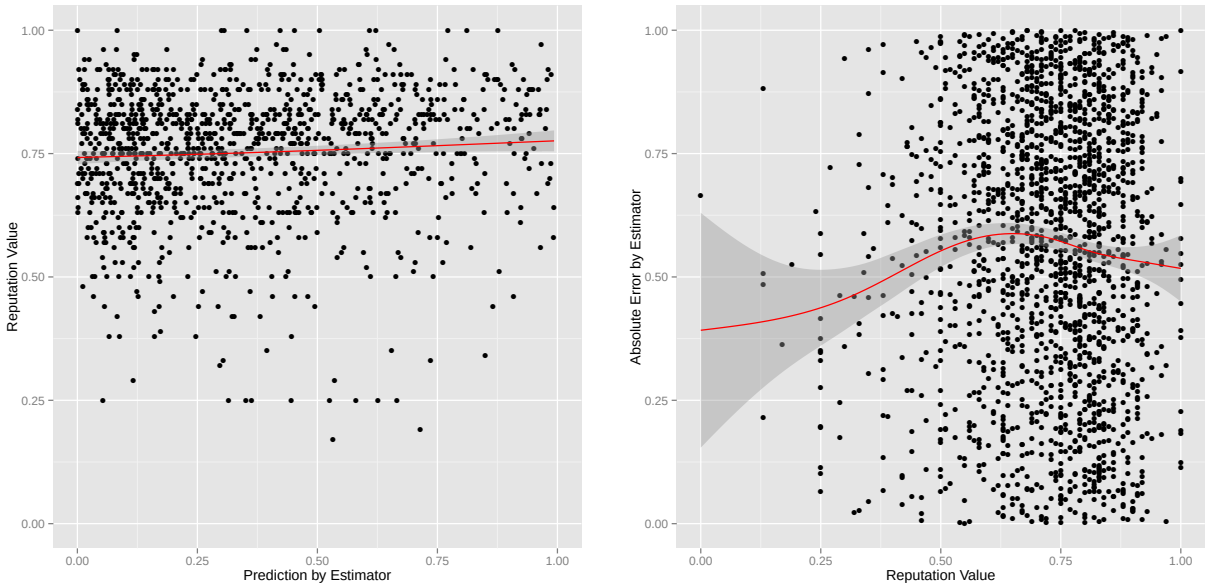
Figure 24. Predictive Performance and Error for M5 Model Decision Tree



(a) Prediction

(b) Absolute Error

Figure 25. Predictive Performance and Error for Classification and Regression Tree (CART)



(a) Prediction

(b) Absolute Error

Figure 26. Predictive Performance and Error for Logistic Regression (logit)

Classification performance of different estimators for dichotomised lass labels of the hotel data set. For each hotel, a new dichotomous response variable $y \in 0; 1$ was computed by using a binomial random number generator with the hotels recommendation score as the corresponding probability.

E. Regression to Reputation Score, no Sampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Results were generated from cross validated bootstrap samples.

	avg AUC	**MIN**	**MAX**	**± SD**
regRF (consistent)	0.590	0.563	0.604	± 0.012
regRF (default)	0.585	0.565	0.599	± 0.014
M5	0.582	0.565	0.6	± 0.012
CART	0.56	0.543	0.575	± 0.01
logit	0.582	0.563	0.603	± 0.014

Table XI
AVERAGE CLASSIFICATION PERFORMANCE FOR UNBALANCED DATA AND RECOMMENDATION SCORE AS REGRESSAND

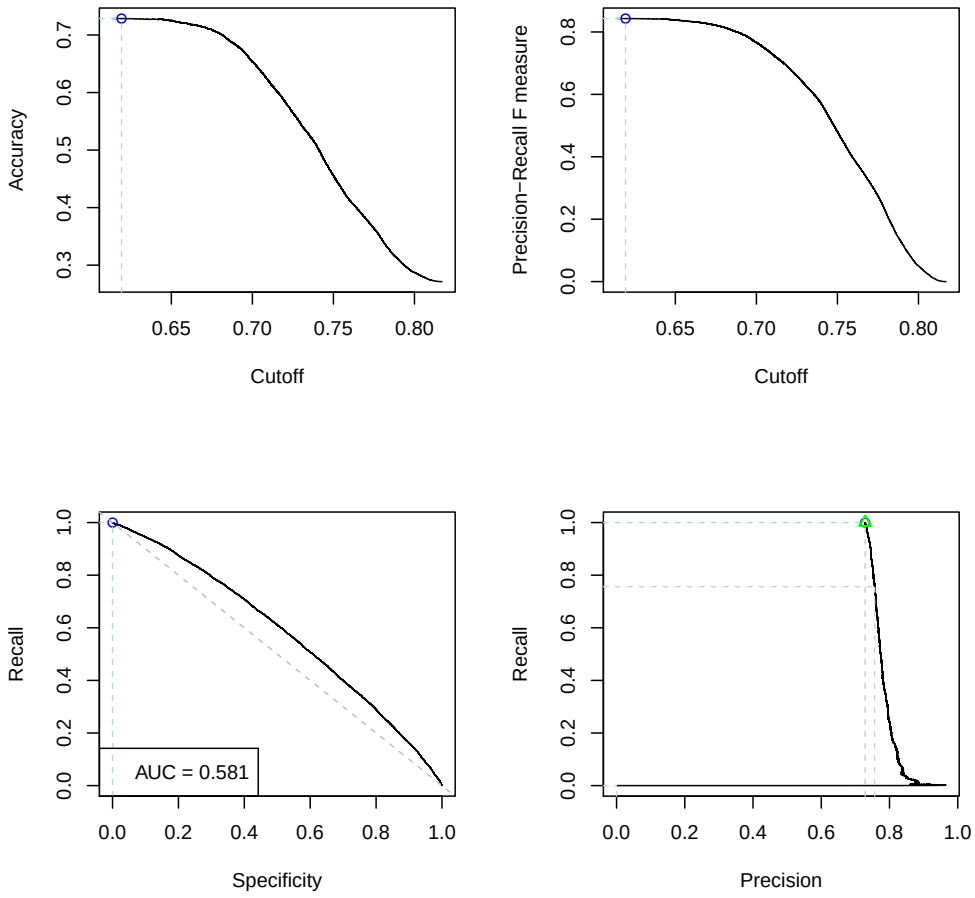


Figure 27. Regression Random Forest, ntree = 10% of number of samples (hotels)

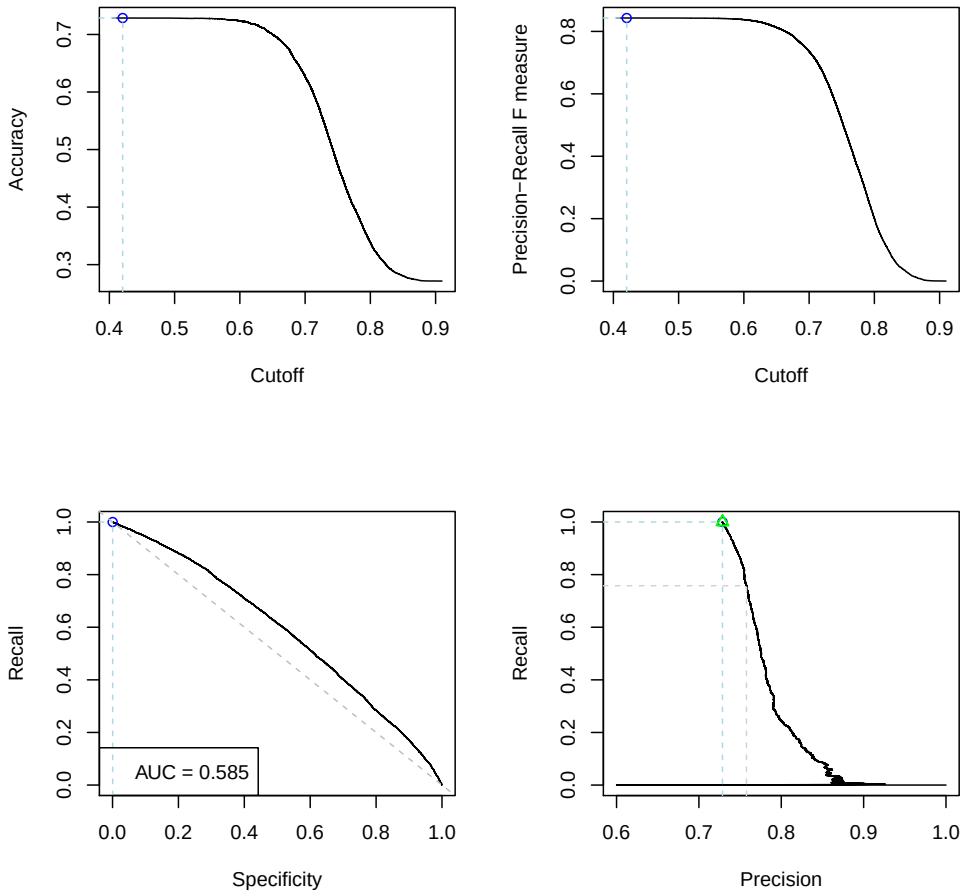


Figure 28. Regression Random Forest, ntree = 5, package default

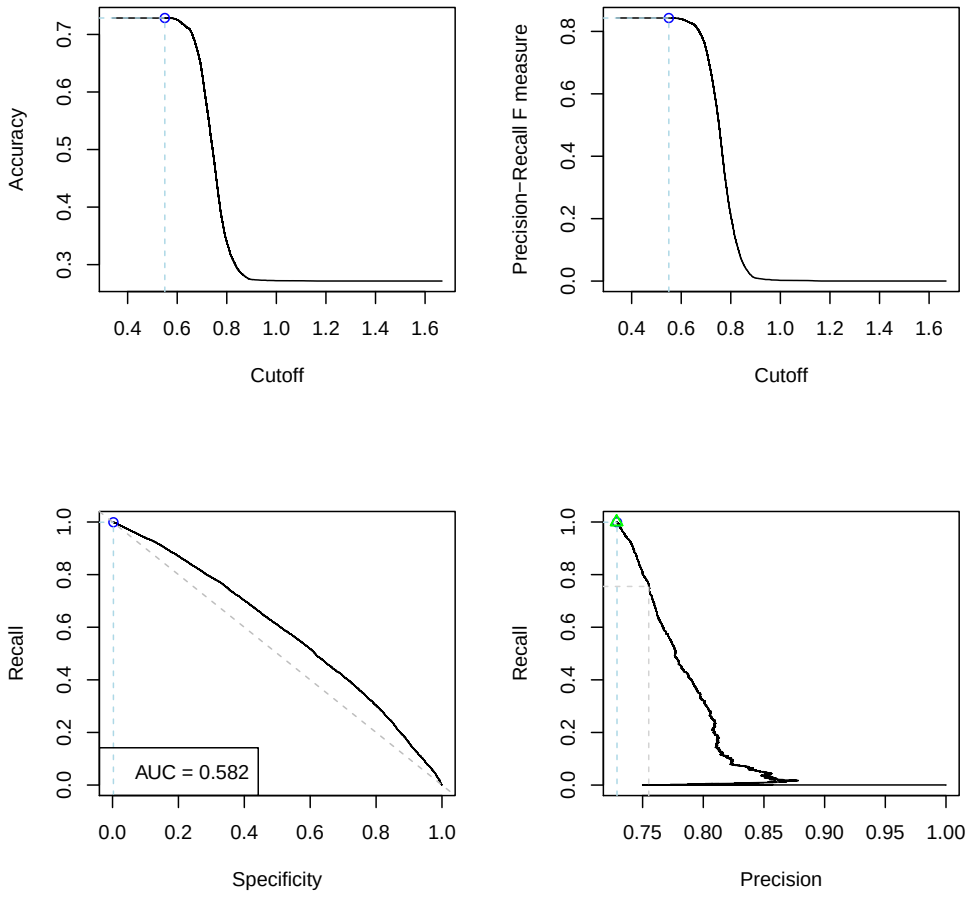


Figure 29. M5 Model Decision Tree

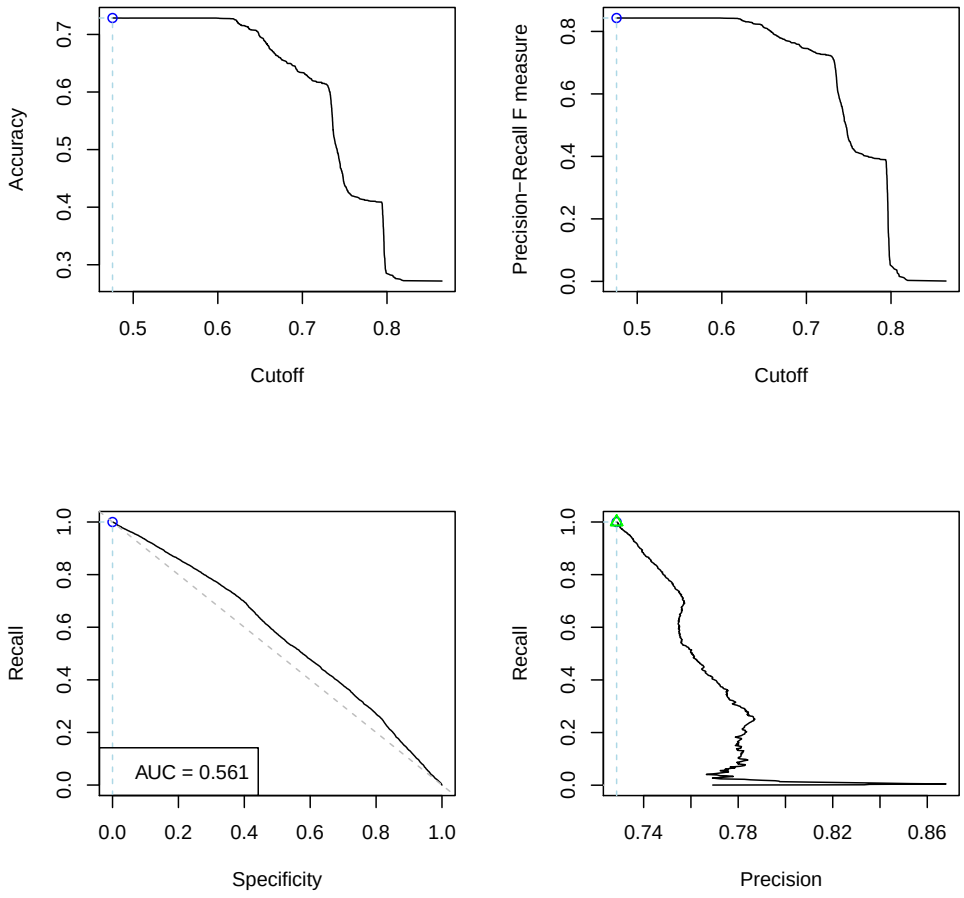


Figure 30. Classification and Regression Tree (CART)

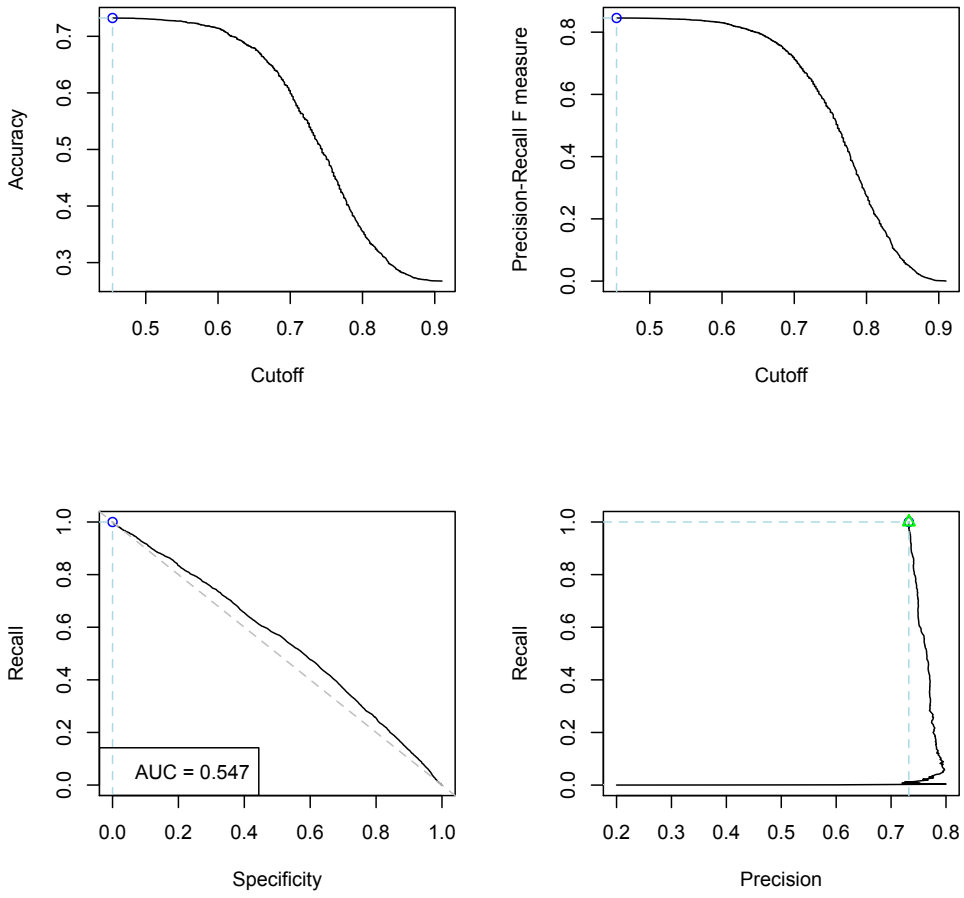


Figure 31. k-Nearest Neighbour (kNN)

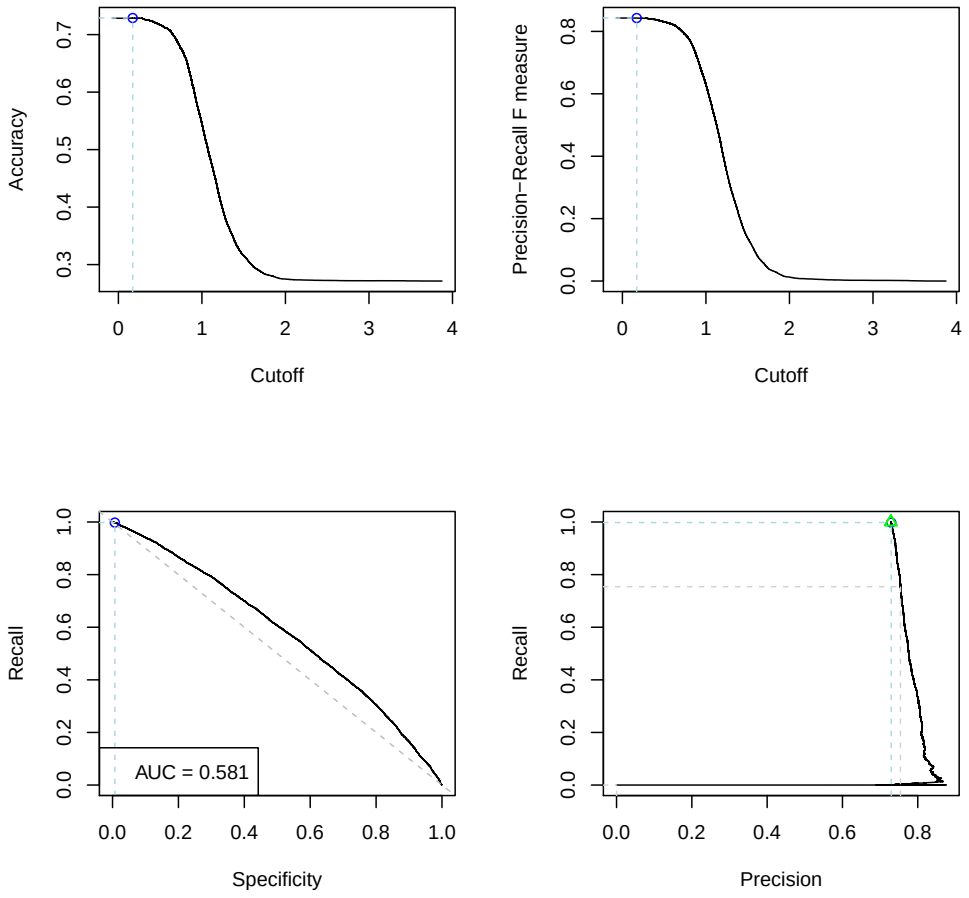


Figure 32. Logistic Regression

F. Regression to Class Label, no Sampling

Classification performance of estimators trained on hotel label, i.e., regressor is dichotomous variable in $\{0; 1\}$. Results were generated from cross validated bootstrap samples.

	avg AUC	**MIN**	**MAX**	$\pm$ **SD**
regRF (consistent)	0.568	0.552	0.585	$\pm$ 0.013
regRF (default)	0.547	0.523	0.579	$\pm$ 0.019
classRF (consistent)	0.529	0.503	0.545	$\pm$ 0.012
classRF (default)	0.55	0.527	0.579	$\pm$ 0.02
M5	0.554	0.523	0.584	$\pm$ 0.019
CART	0.529	0.505	0.544	$\pm$ 0.012
kNN	0.548	0.529	0.564	$\pm$ 0.014
bNN	0.541	0.505	0.564	$\pm$ 0.024
logit	0.557	0.535	0.583	$\pm$ 0.016

Table XII
AVERAGE CLASSIFICATION PERFORMANCE FOR UNBALANCED DATA AND CLASS LABEL AS REGRESSAND

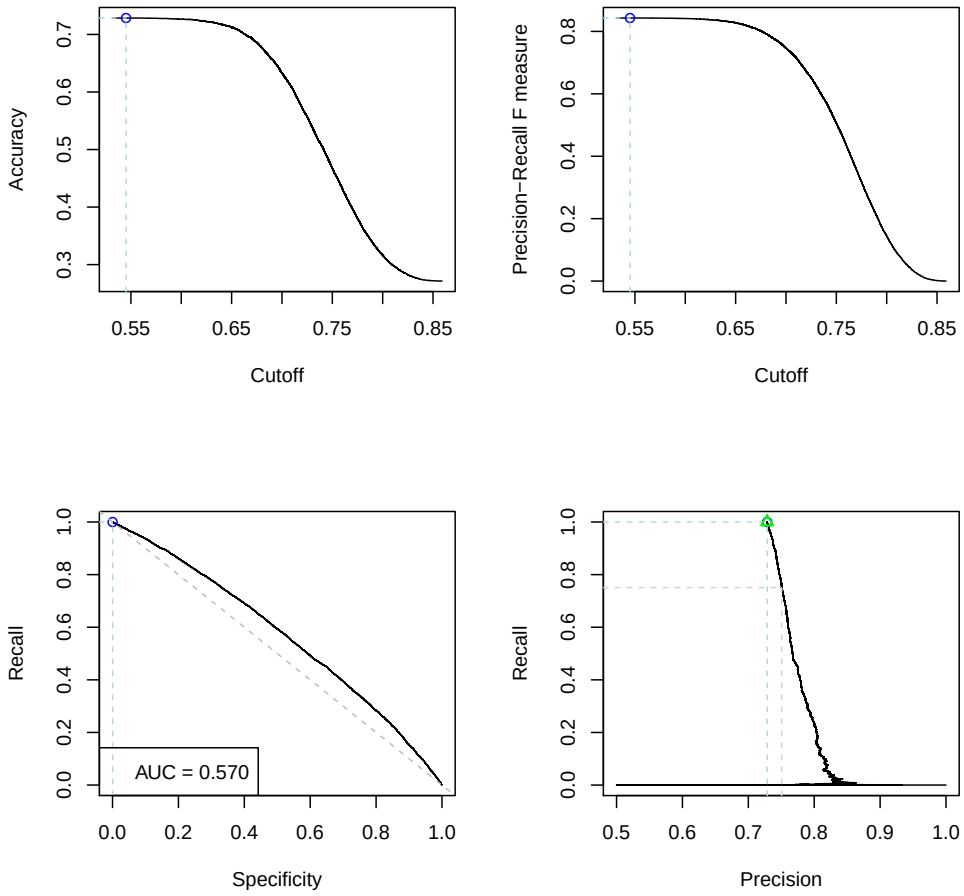


Figure 33. Regression Random Forest, ntree = 10% of number of samples (hotels)

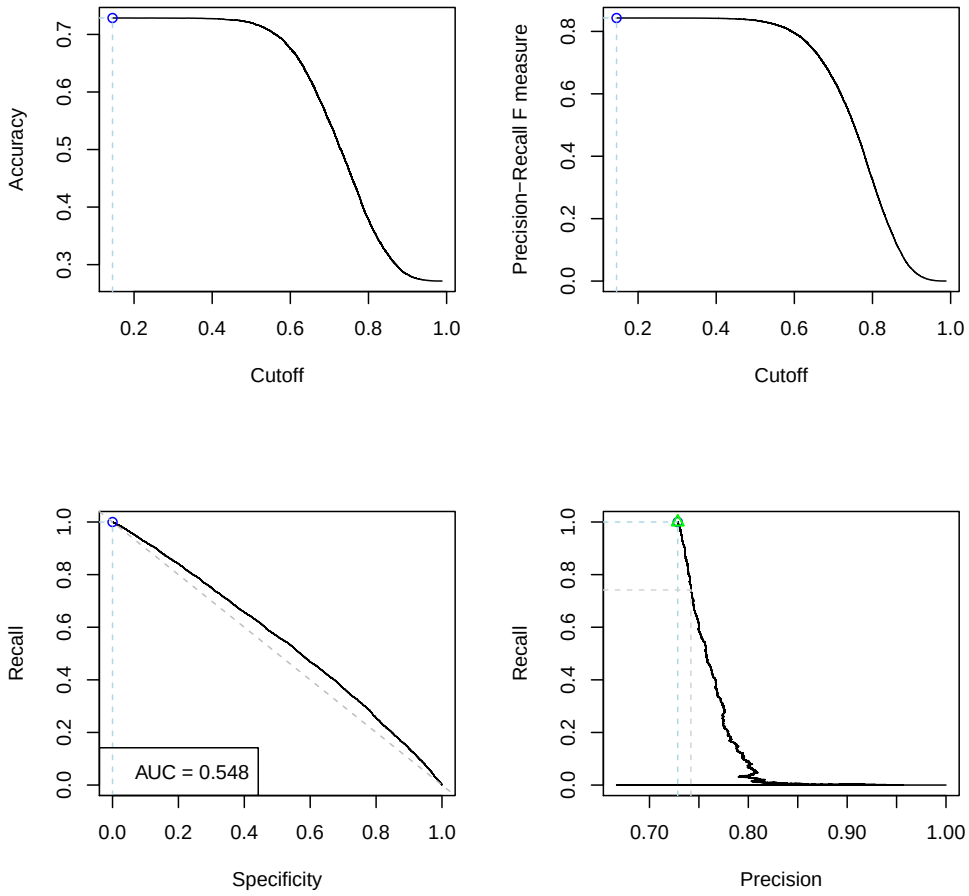


Figure 34. Regression Random Forest, ntree = 5, package default

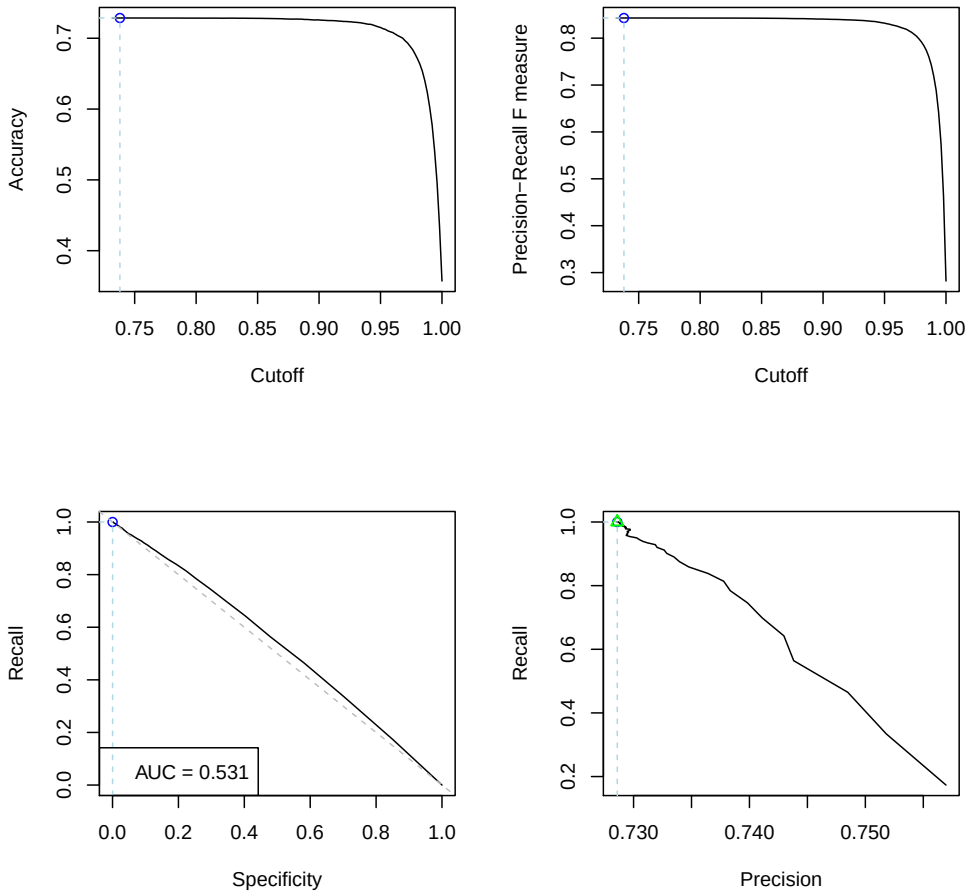


Figure 35. Classification Random Forest, ntree = 10% of number of samples (hotels)

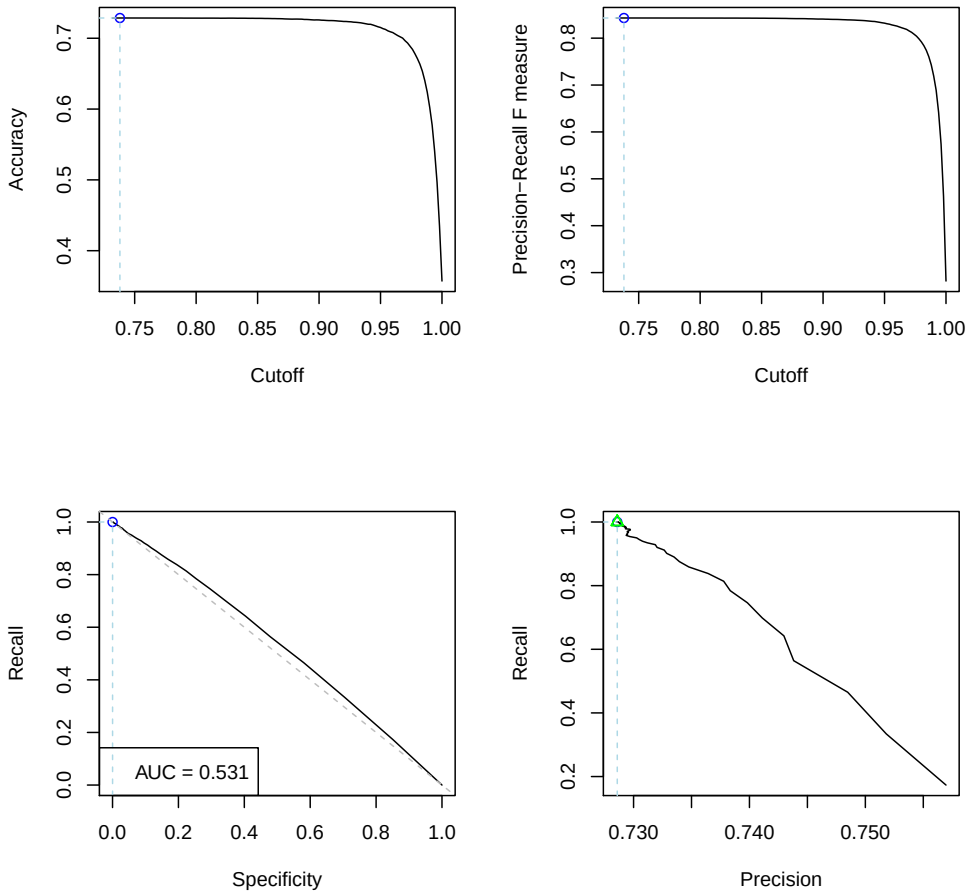


Figure 36. Classification Random Forest, ntree = 1, package default

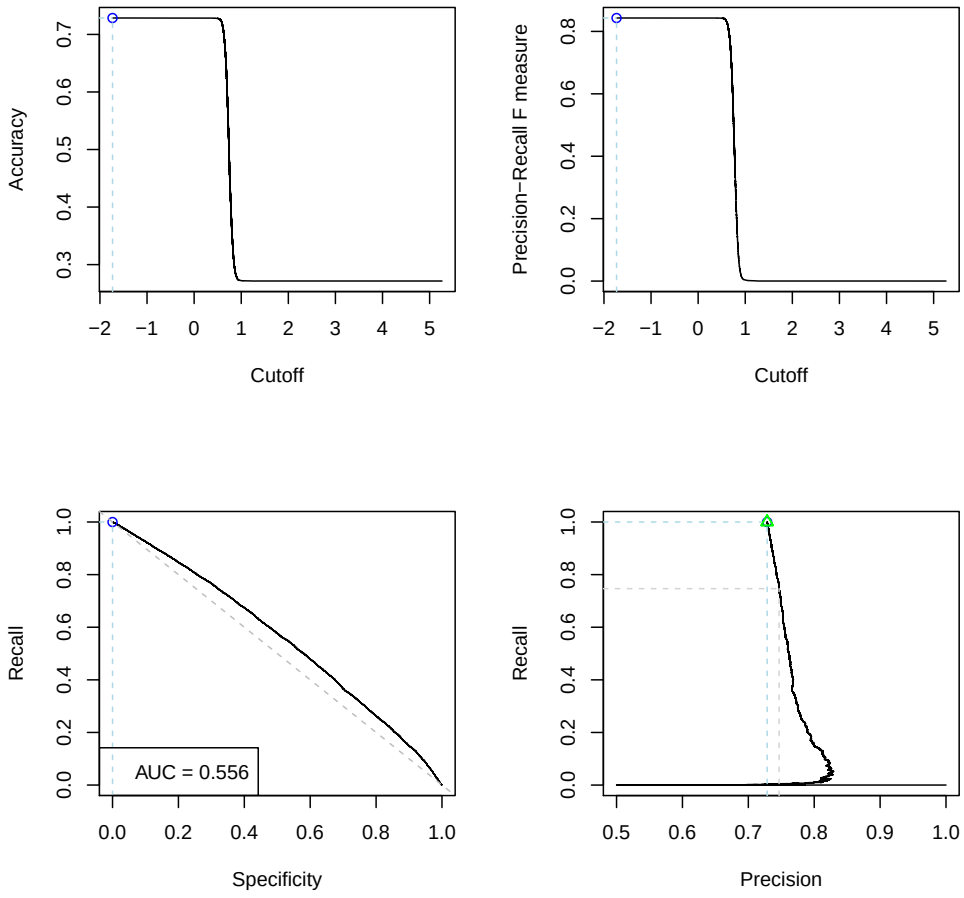


Figure 37. M5 Model Decision Tree

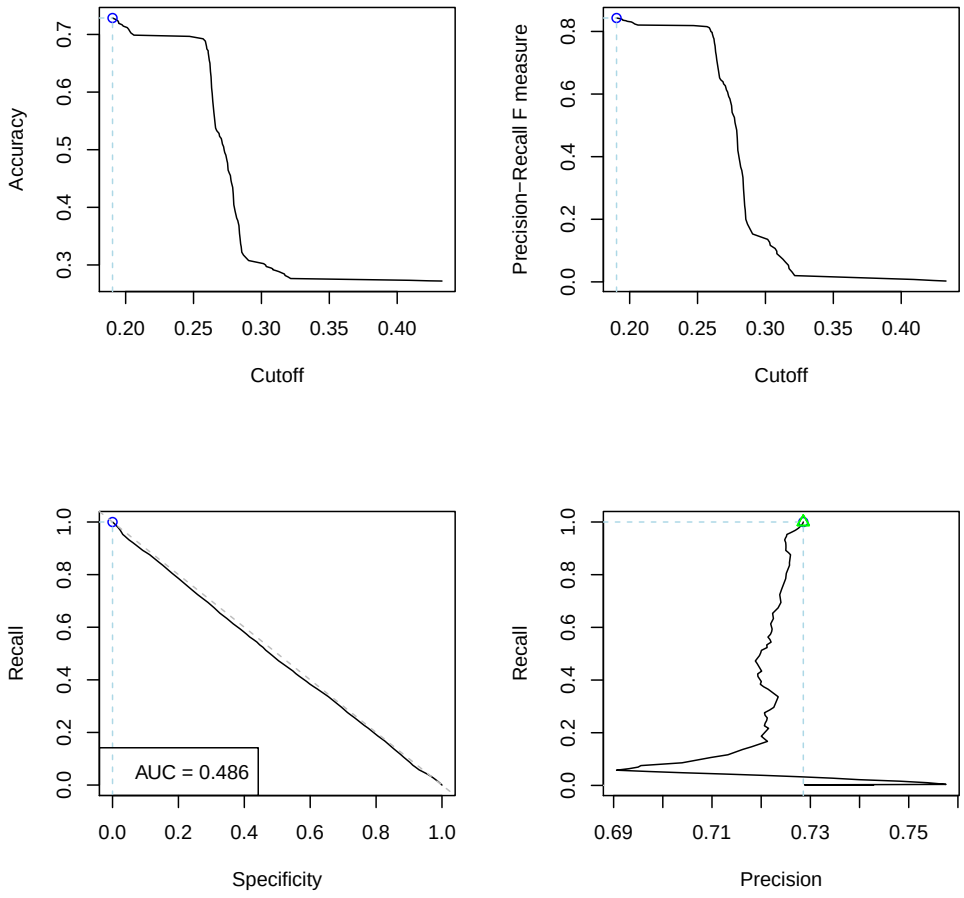


Figure 38. Classification and Regression Tree (CART)

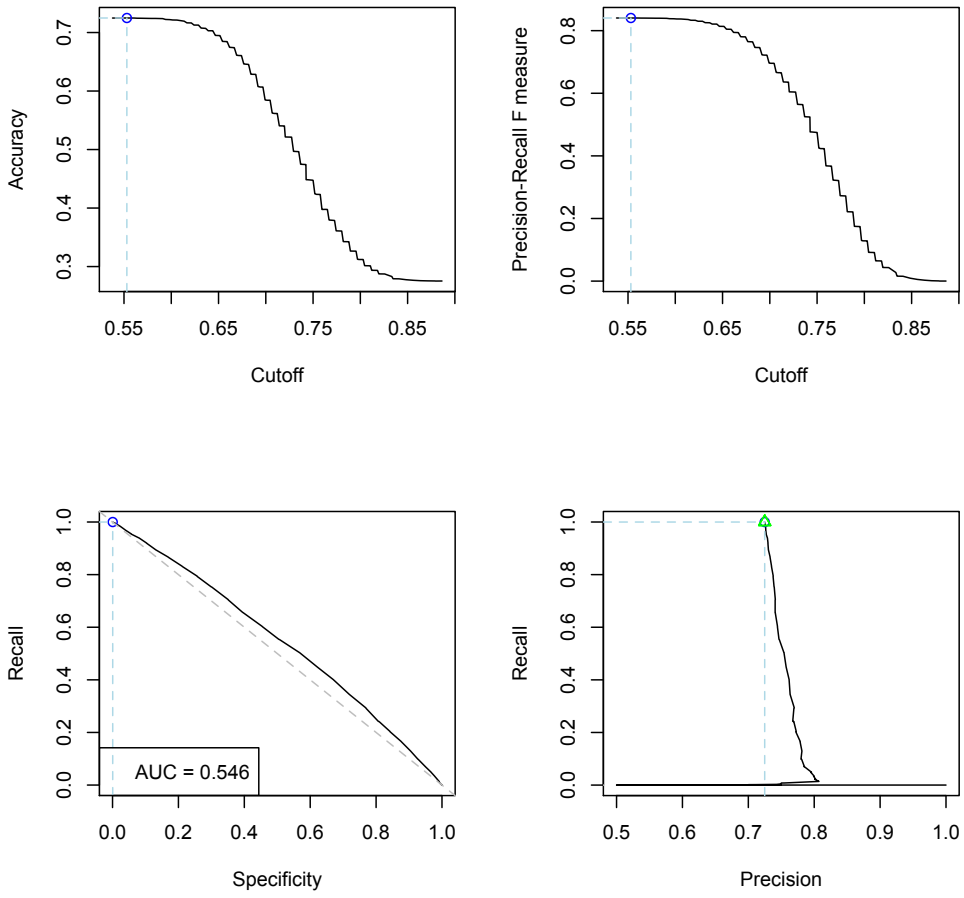


Figure 39. k-Nearest Neighbour (kNN)

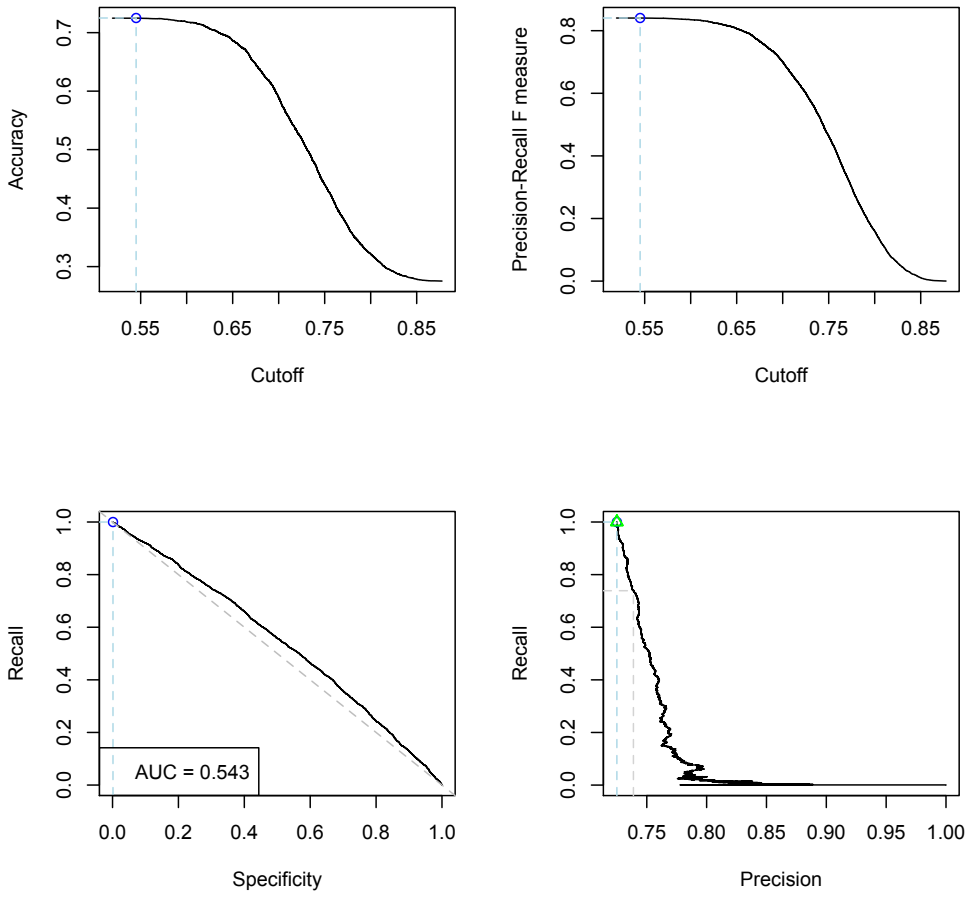


Figure 40. Bagged 1-Nearest Neighbour (bNN)

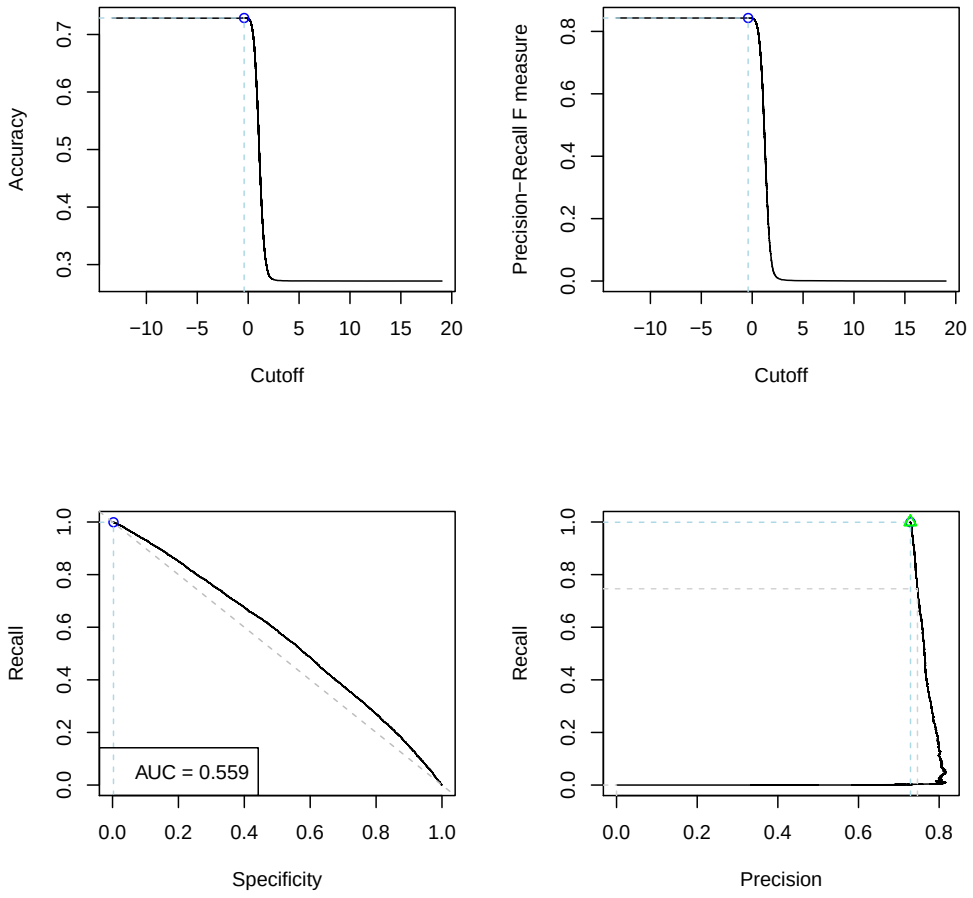


Figure 41. Logistic Regression

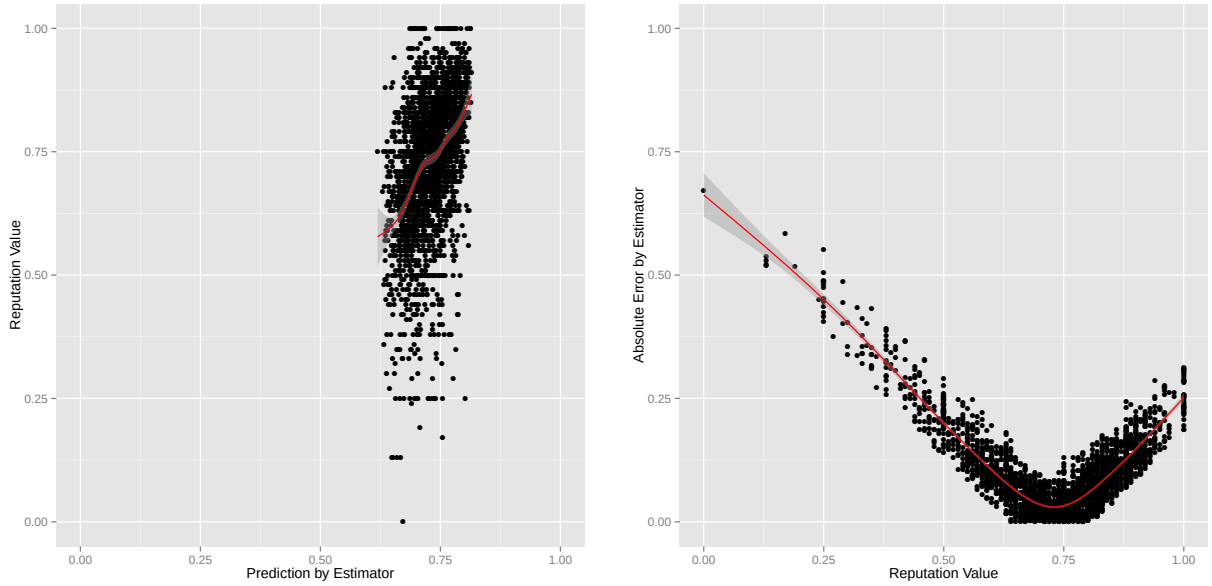
Predictive performance of regression and probability machines. Performance measured based on the error function between predicted value $\hat{Y}$ and reputation score Y.

G. Regression to Reputation Score, no Sampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Results were generated from cross validated bootstrap samples.

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	0	0.1	0.02	0.13	91.8	0	0.92	0.29	0.16	0.11	0.43	0.32
regRF (default)	0	0.09	0.02	0.12	89.6	-0.6	0.9	0.44	0.2	0.14	0.56	0.41
M5	0	0.09	0.02	0.13	91.8	0	0.92	0.44	0.16	0.11	0.53	0.38
CART	0	0.1	0.02	0.13	94	0	0.94	0.39	0.12	0.08	0.47	0.35
kNN	0	0.11	0.02	0.14	103.4	-0.3	1.03	0.52	-0.07	-0.02	0.45	0.34
logit	0.29	0.33	0.17	0.42	301.2	39.3	3.01	2.37	-8.07	-2.12	0.39	0.26

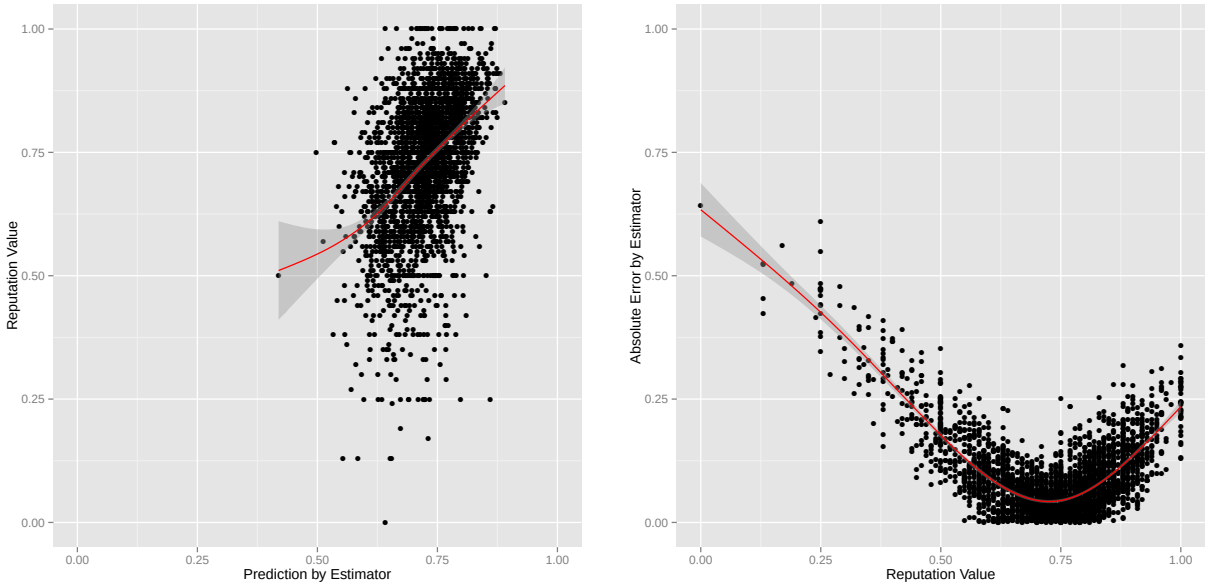
Table XIII
AVERAGE GOODNESS OF FIT FOR UNBALANCED DATA AND RECOMMENDATION SCORE AS REGRESSAND



(a) Prediction

(b) Absolute Error

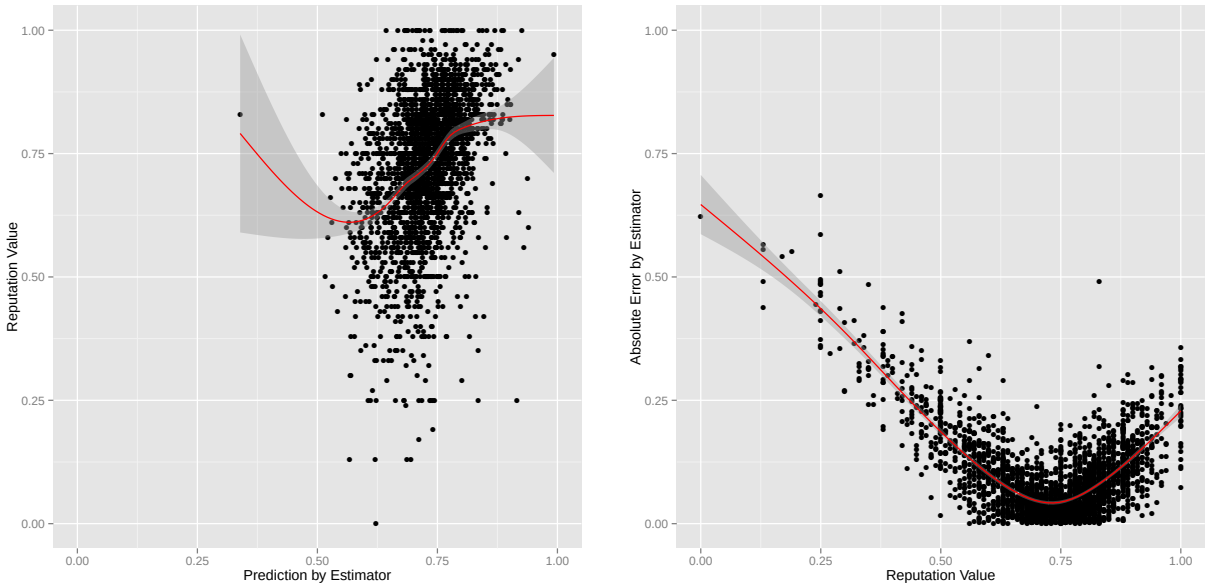
Figure 42. Predictive Performance and Error for Regression Random Forest (regRF, consistent, ntree=10%)



(a) Prediction

(b) Absolute Error

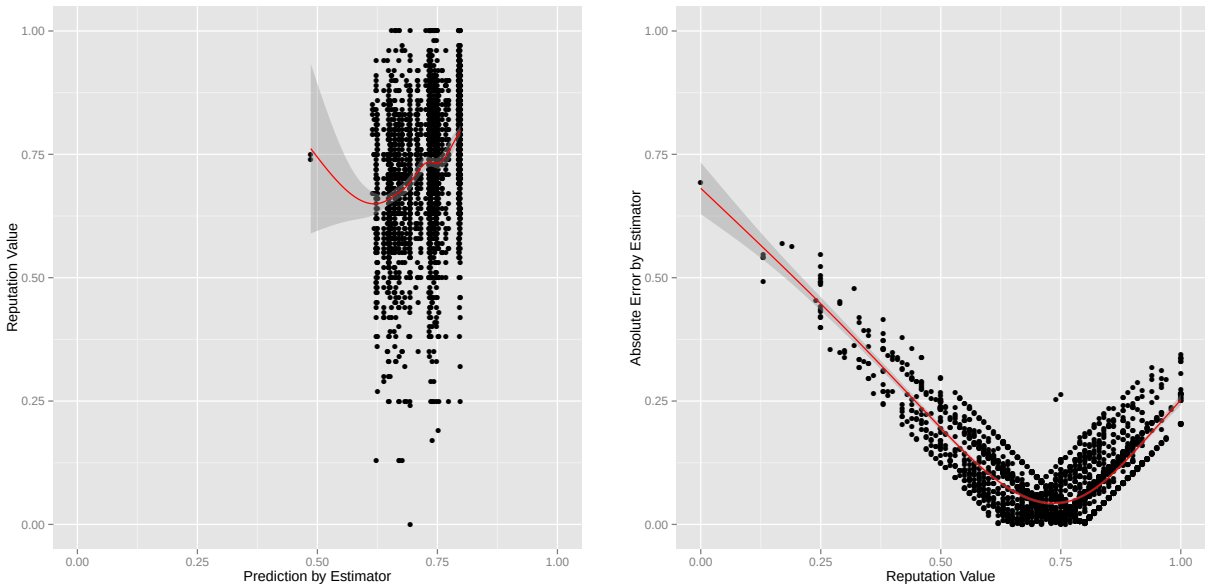
Figure 43. Predictive Performance and Error for Regression Random Forest (regRF, default, ntree=5)



(a) Prediction

(b) Absolute Error

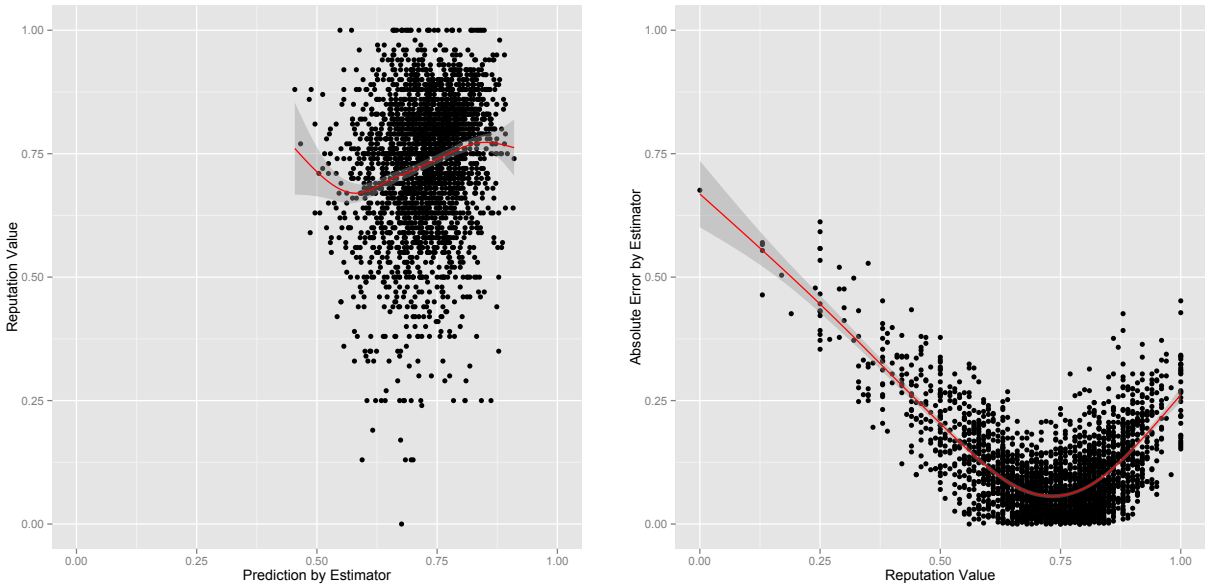
Figure 44. Predictive Performance and Error for M5 Model Decision Tree



(a) Prediction

(b) Absolute Error

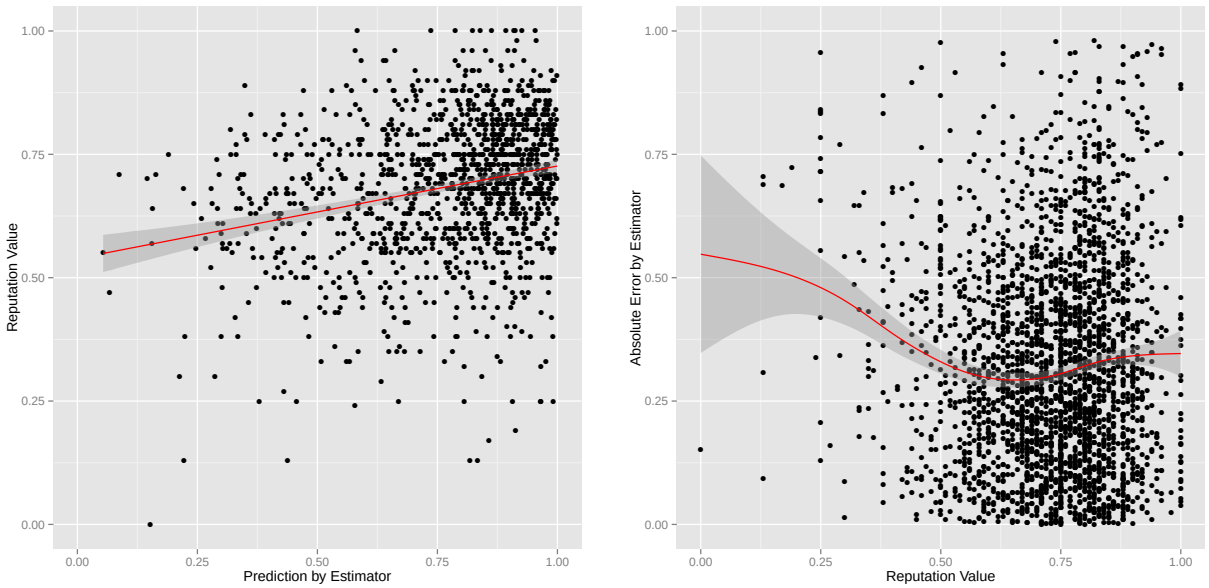
Figure 45. Predictive Performance and Error for Classification and Regression Tree (CART)



(a) Prediction

(b) Absolute Error

Figure 46. Predictive Performance and Error for k-Nearest Neighbour (kNN)



(a) Prediction

(b) Absolute Error

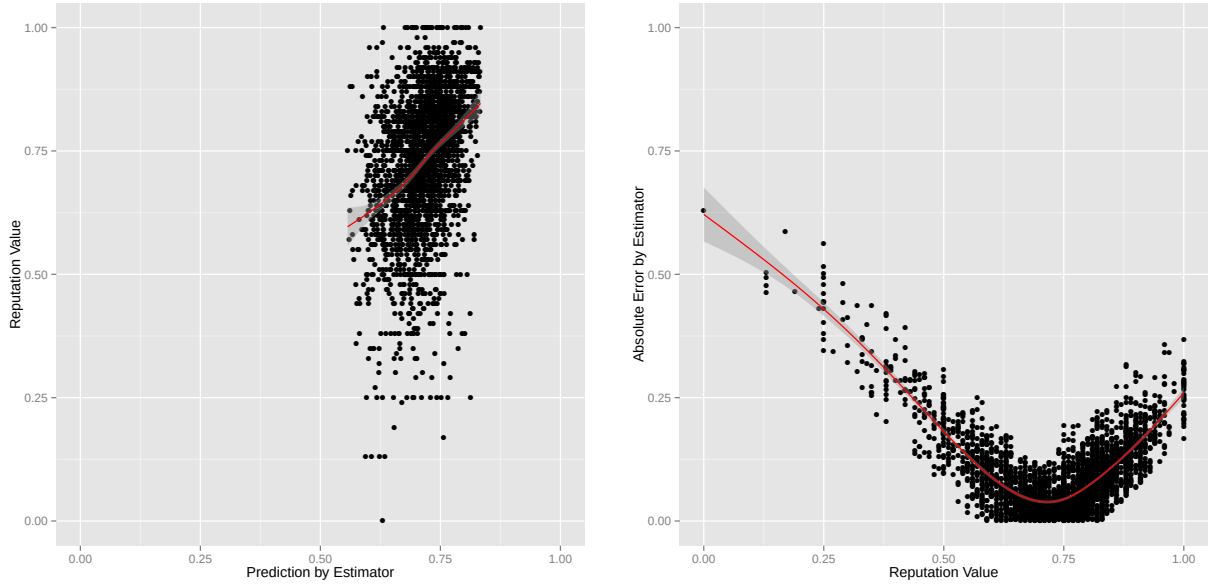
Figure 47. Predictive Performance and Error for Logistic Regression (logit)

H. Regression to Class Label, no Sampling

Classification performance of estimators trained on hotel reputation score, i.e., regressor is a continuous variable in $[0; 1]$. Results were generated from cross validated bootstrap samples.

	ME	MAE	MSE	RMSE	NRMSE %	PBIAS %	RSR	rSD	NSE	mNSE	d	md
regRF (consistent)	0	0.1	0.02	0.13	94.2	-0.1	0.94	0.36	0.11	0.08	0.45	0.33
regRF (default)	-0.02	0.12	0.02	0.15	110.6	-2.9	1.11	0.75	-0.22	-0.11	0.52	0.38
classRF (consistent)	0.26	0.26	0.09	0.29	212.8	35.6	2.13	0.13	-3.53	-1.43	0.39	0.29
classRF (default)	-0.01	0.11	0.02	0.15	107.6	-1.7	1.08	0.7	-0.16	-0.07	0.52	0.38
M5	0	0.1	0.02	0.14	102.2	-0.1	1.02	0.59	-0.04	0.03	0.49	0.37
CART	-0.46	0.46	0.23	0.48	347.6	-62.8	3.48	0.17	-11.08	-3.3	0.3	0.19
kNN	-0.01	0.1	0.02	0.13	96.8	-1.2	0.97	0.28	0.06	0.04	0.36	0.27
bNN	-0.01	0.1	0.02	0.13	96.8	-1.1	0.97	0.3	0.06	0.04	0.37	0.28
logit	0.3	0.41	0.35	0.59	427.8	41.5	4.28	3.8	-17.3	-2.84	0.27	0.23

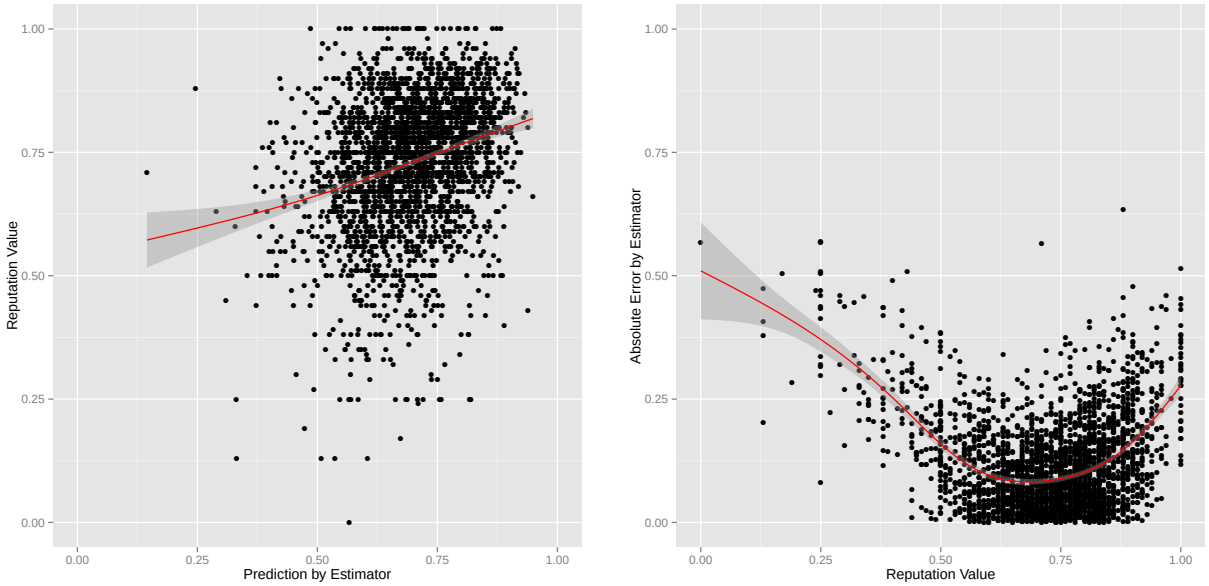
Table XIV
AVERAGE GOODNESS OF FIT FOR UNBALANCED DATA AND CLASS LABEL AS REGRESSAND



(a) Prediction

(b) Absolute Error

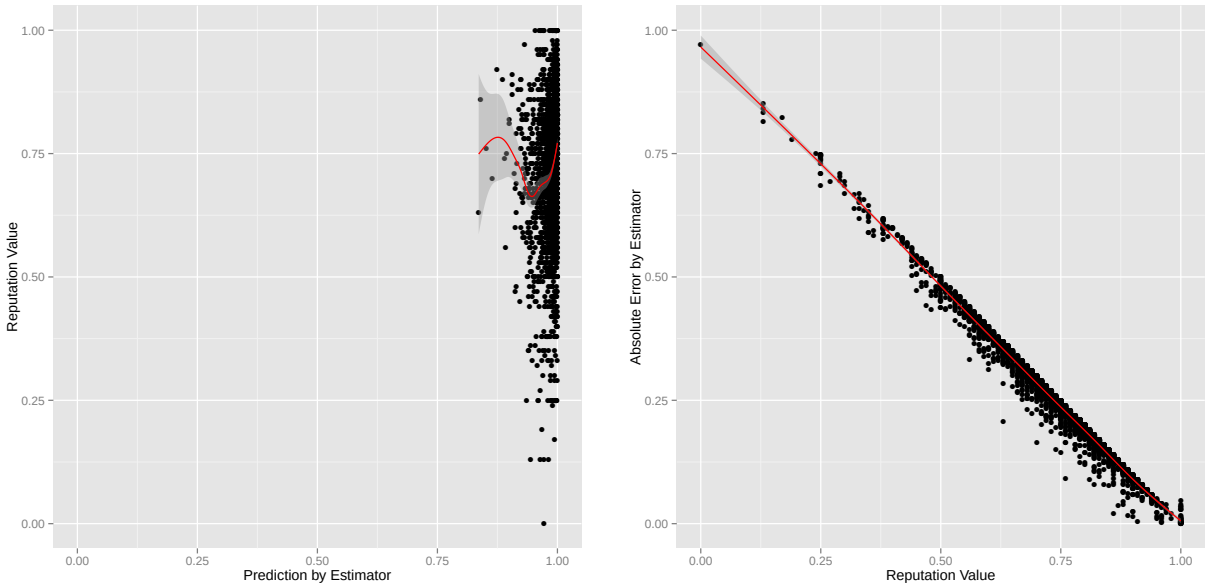
Figure 48. Predictive Performance and Error for Regression Random Forest (regRF, consistent, ntree=10%)



(a) Prediction

(b) Absolute Error

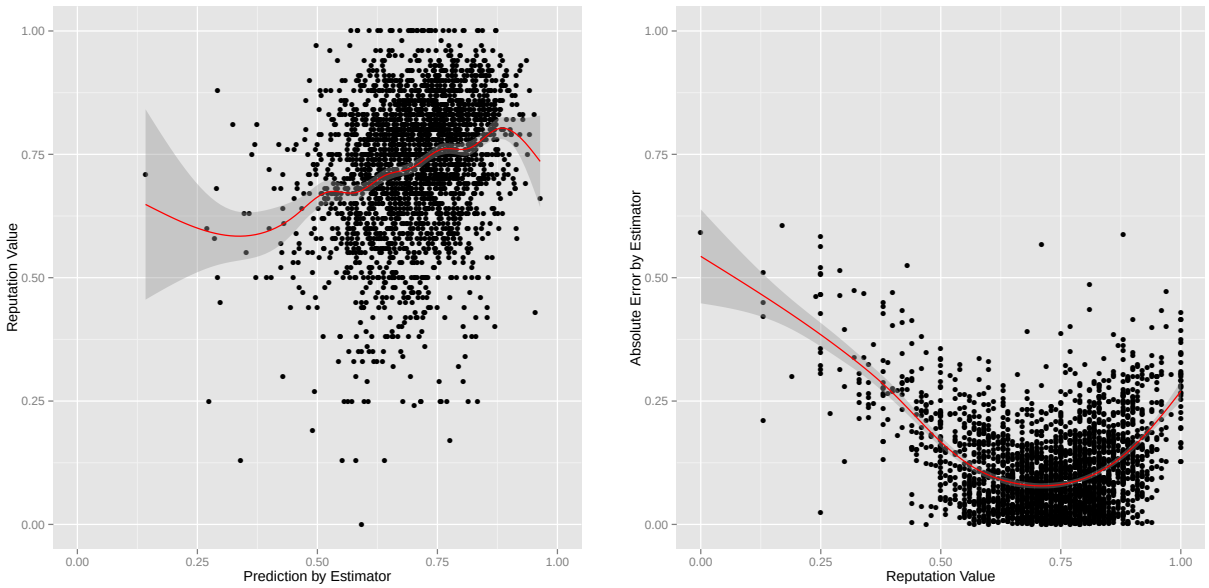
Figure 49. Predictive Performance and Error for Regression Random Forest (regRF, default, ntree=5)



(a) Prediction

(b) Absolute Error

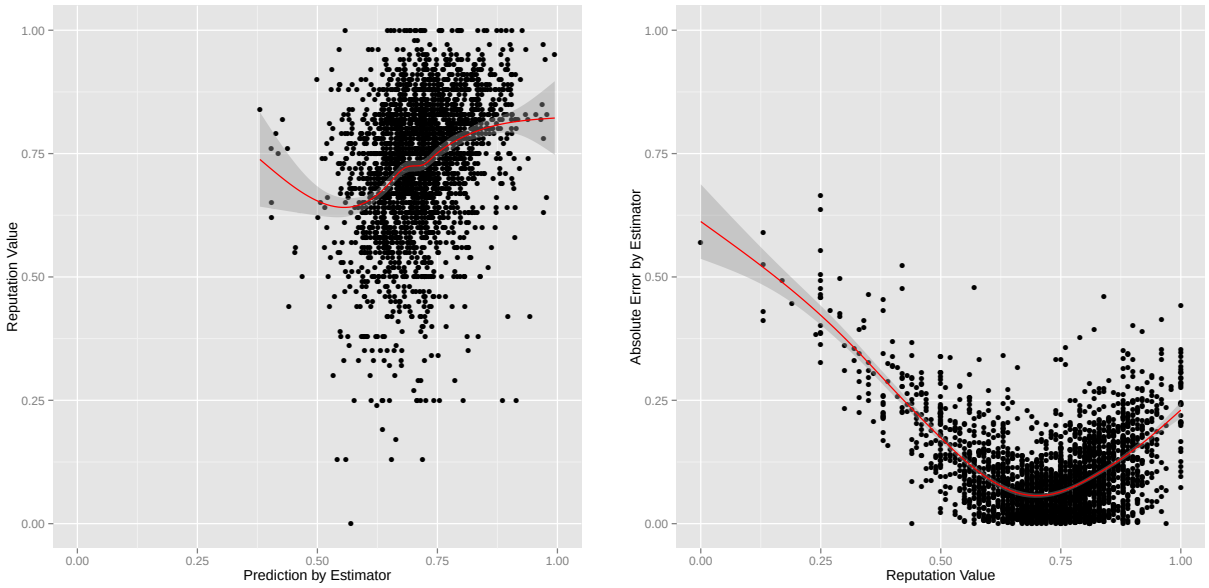
Figure 50. Predictive Performance and Error for Classification Random Forest (classRF, consistent, ntree=10%)



(a) Prediction

(b) Absolute Error

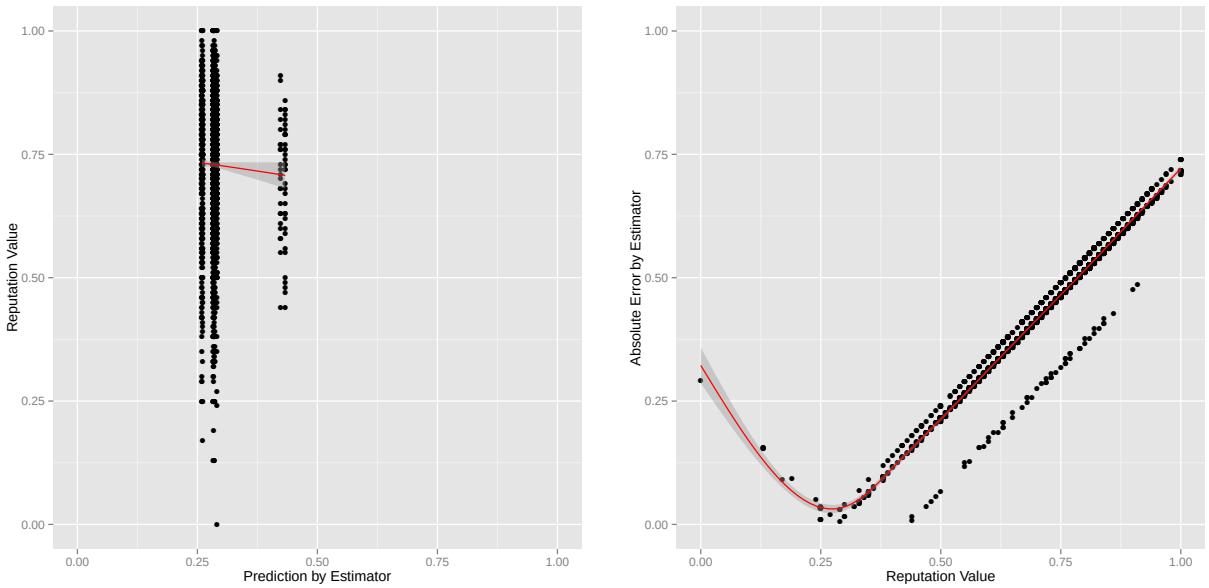
Figure 51. Predictive Performance and Error for Classification Random Forest (classRF, default, ntree=1)



(a) Prediction

(b) Absolute Error

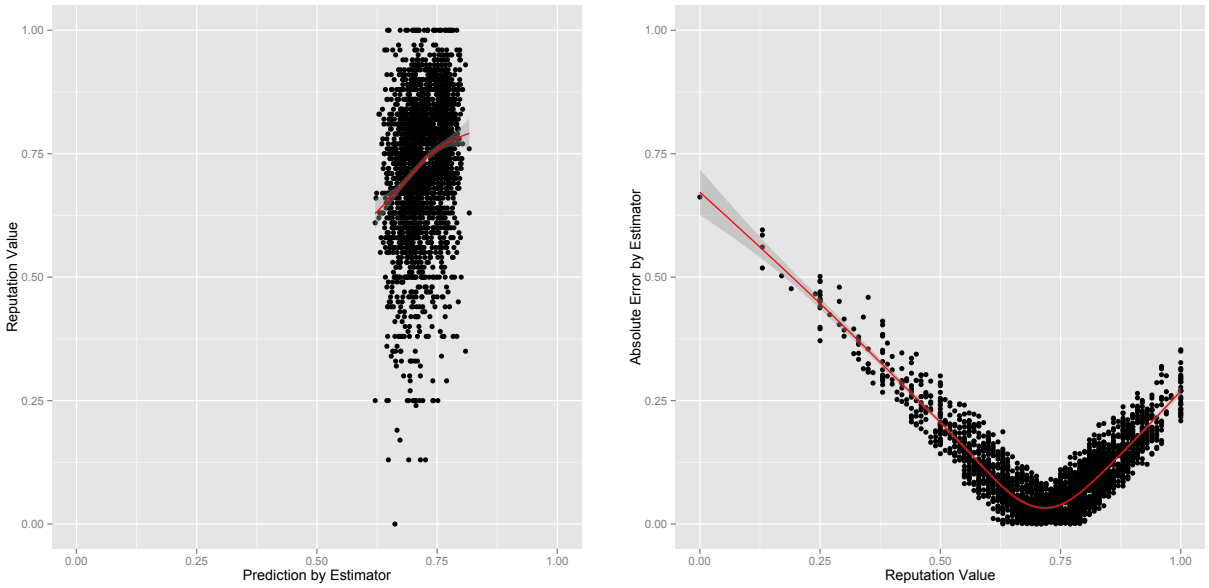
Figure 52. Predictive Performance and Error for M5 Model Decision Tree



(a) Prediction

(b) Absolute Error

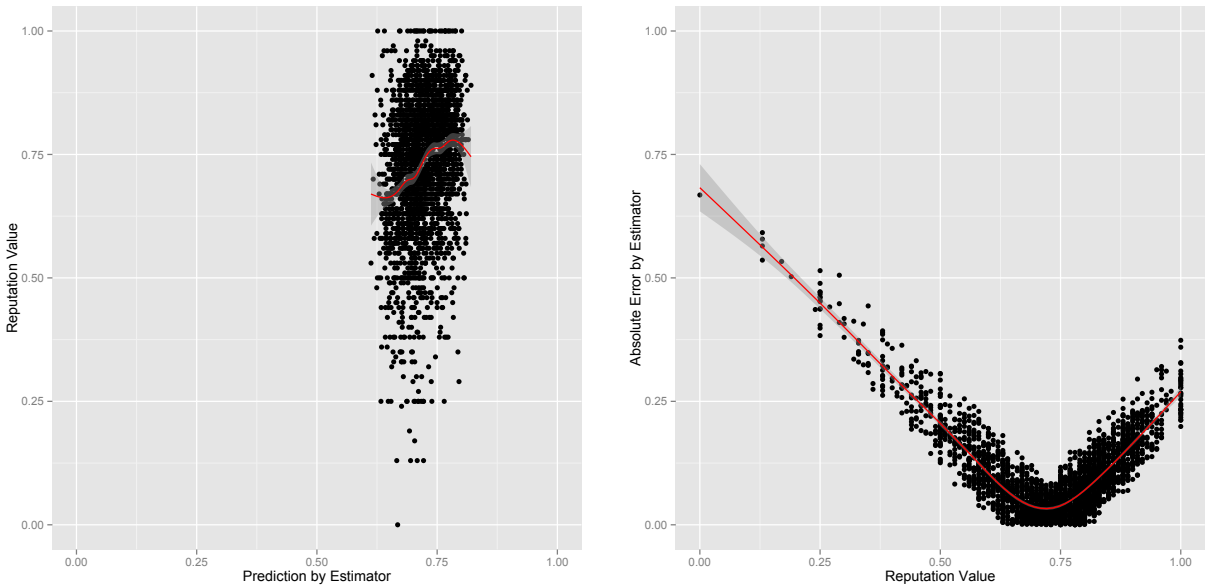
Figure 53. Predictive Performance and Error for Classification and Regression Tree (CART)



(a) Prediction

(b) Absolute Error

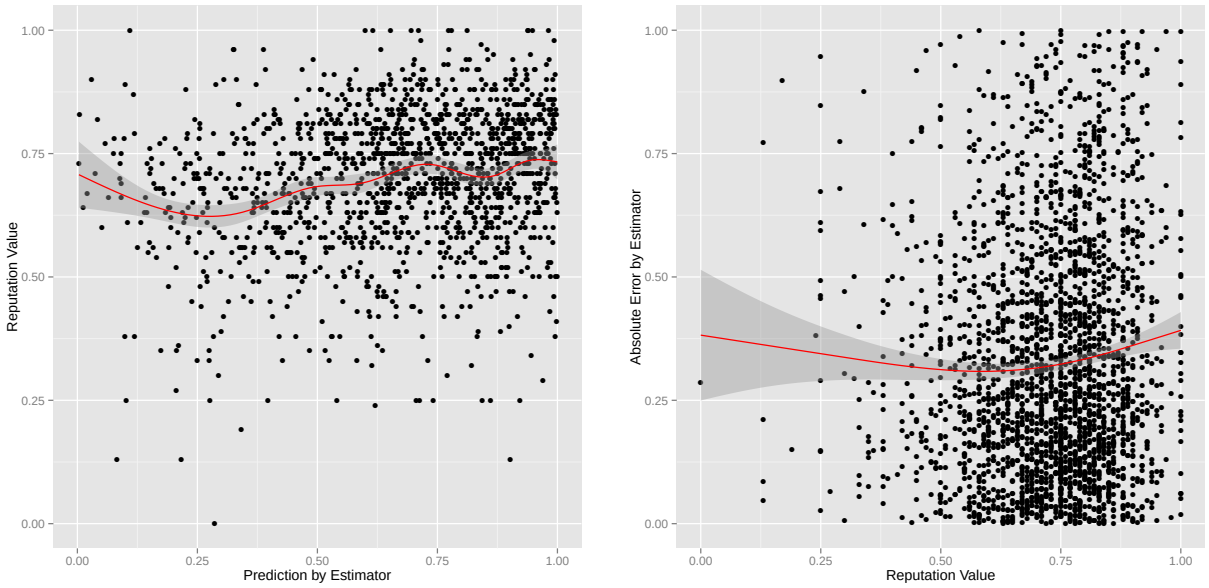
Figure 54. Predictive Performance and Error for k-Nearest Neighbour (kNN)



(a) Prediction

(b) Absolute Error

Figure 55. Predictive Performance and Error for bagged 1-Nearest Neighbour (bNN)



(a) Prediction

(b) Absolute Error

Figure 56. Predictive Performance and Error for Logistic Regression (logit)