ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus School of Economics

MSc. Econometrics and Management Science

Master Thesis in Business Analytics & Quantitative Marketing

Less is more: improving customer churn prediction with split training data in the Penalized Logit Leaf Model

Author:

Martijn Hendriks

432592

Supervisor:

dr. E.P. O'Neill

Second assessor:

dr. M. Zhelonkin

August 1, 2022

Abstract

This study tries to improve data-based decision making by extending the Logit Leaf Model for customer churn prediction. Our goal is to create a high performing prediction model for insurance policy churn that is interpretable for businesses. We introduce the Penalized Logit Leaf Model that, in combination with *honest estimation*, is able to produce highly accurate unbiased probability estimates. We find that for imbalanced classification data, using split training data for hybrid tree model building benefits the predictive accuracy as well as model calibration. We benchmark the proposed models to the Logit Leaf Model, Random Forest and Elastic Net Logit using multiple datasets.

Contents

1 Introduction

Customer churn prediction, predicting the loss of customers to another party, is an essential element in customer relation management. Businesses need to gain insight in their customer base to efficiently maintain customers they have gained often using expensive marketing campaigns. This study is an extension of the research by De Caigny et al. (2018). We try to improve their method for customer churn prediction, the Logit Leaf Model (LLM). This novel method combines the interpretability and ease of a single decision tree in the first step and logistic regression within the leaves in a second step. This hybrid method outperforms various other methods in customer churn prediction and can also be used for other binary classification purposes. Another advantage of the LLM is its interpretability. Since it makes use of a single tree to subset the population into homogeneous groups and outputting parameter estimates for the attributes, it is far more useful for decision makers than high performing black-box machine learning methods. Our improvement in the model lies in an upgrade of the classification accuracy while maintaining the hybrid structure of the model and the interpretability advantages. This is done through an improved variable selection method for the logistic regression using Elastic Net based penalties. We also try to remove bias as well as provide ways to incorporate high dimensional categorical variables into the model, which there are plenty of in real life customer data.

The LLM is currently used in various predictive classification tasks, including student dropout (Coussement et al., 2020), bankruptcy (Pawełek & Pociecha, 2020) and disease classification (Nurrohman et al., 2020). Why do we want to improve specifically the LLM? The LLM is a promising method in terms of corporate usability, with a competitive predictive accuracy. There are some clear areas for potential improvement. The main area for improvement is where the LLM uses forward variable selection for logistic regression, which can be improved using more advanced logistic regression models. This research answers the following questions:

- What is the most accurate way, within a set of methods for an insurance dataset, to predict customer churn when keeping interpretability for business managers as a priority?
- Can the LLM be improved by substituting logit with forward variable selection with penalized logit?
- Can high cardinality categorical attributes be included efficiently without loss of interpretability of the model?
- Can bias be reduced by using honest estimation, without loss of predictive accuracy?

We perform our research using a business case for a dutch non-life insurance company.

The models are applied to a customer churn dataset supplied by the insurer, that consists of 80,000 policies in the year 2021. Several more datasets are used in the benchmark study. We find that our proposed model is a very accurate classifier for policy churn. This is useful for insurers to get knowledge of the size of the future portfolio. Furthermore, due to the focus on interpretability of the model, the business will gain useful insight in the clusters of policies. To explore the possibilities of the model beyond churn prediction, we include a credit card fraud dataset, alongside three other datasets in the benchmark study, which are publicly available.

This study concludes that the Logit Leaf Model can be improved using Elastic Net logit. We also find that honest estimation in this model does both remove bias as well as improve the model's predictive performance. For the insurance company, we find that the honest Penalized Logit Leaf Model in combination with *group means encoding* of categorical variables performs best. In general, the honest PLLM performs competitive with Elastic Net logit and Random Forest in terms of AUC, but achieves superior Brier scores, probability calibration and model interpretability. This model may also be useful for other binary classification purposes like fraud detection, customer attraction and customer scoring.

This paper is structured as follows. Section 2 elaborates on related previous studies and presents opportunities where existing literature can be improved. Section 3 describes the data used in the case study. Section 4 explains the methods of the models used along with the proposed extensions. The last sections present the results of the case study and give conclusions and notes for further research.

2 Literature Review

This section is a review of much of the related literature on the main topics of this study. For every improvent we try to make, there is an overview of what has been done on that aspect. Also a short summary of studies on customer churn prediction is given.

2.1 Classification models for customer churn and LLM

Since the turn of the century, there has been an enormous number of studies on the prediction of customer churn. And the topic is still very hot, as the number of studies on customer churn grows by the day[1]. Most of these studies focus on a select number of models; Logistic Regression, Decision Trees, Neural Networks, k-Nearest Neighbours, Random Forests, boosted trees or Support Vector Machines. Table 1 in De Caigny et al. (2018) shows a short overview of related studies from 2012 to 2017.

Xong et al. (2019) compare Logistic Regression (LR), k-nearest neighbours (kNN), Neural Networks (NN) and Support Vector Machines (SVM) for life insurance lapse risk. They assess performance using the Area under the Reciever Characteristics Operating Curve (AUC). They find $LR < kNN < NN < SVM$ in terms of accuracy. But the lowest AUC, for the Logit model, was above 0.80, so they conclude that all methods perform very well.

De Caigny et al. (2018) propose the Logit Leaf Model(LLM) and benchmark it against various established models on fourteen different datasets. They use AUC to define the performance of the model and find the LLM outperforms LR and Decision Tree (DT) and performs as least as well as Random Forest (RF) and Logistic Model Trees (LMT). The LLM has one very strong advantage which is its comprehensibility. As mentioned in section 1, the LLM has many use cases. The LLM makes use of logistic regression with forward variable selection in the leaves. Logistic regression with forward variable selection has some shortcomings, most notably its difficulty to deal with correlated regressors (Zou & Hastie, 2005; Whittingham et al., 2006).

Coussement et al. (2020) found the LLM outperformed all benchmark methods (LR, DT, Bagging, RF, XGBoost, LMT, NN, SVM) on the balance of predictive power and comprehensibility. This was tested on a small student dropout data set of 10 thousand observations with a dropout rate of 55 percent. Markapudi et al. (2021) improved the LLM model by applying gradient boosting in the second step of the LLM, called Boosted Leaf Model. It outperformed the LLM in terms of the AUC, but the use of booting in a tree-based model results in a loss of interpretability compared to a single tree.

[1] Google Scholar results for “customer churn prediction” show 2990 published articles in 2021

2.2 Honest estimation

Sampling is commonly used to reduce the bias of a model. Faraway (1998) proposed splitting the data for model selection and parameter estimation to obtain more 'honest' estimates. Faraway (2016) uses the first sample to construct the model after which both samples are used for estimation. Athey & Imbens (2016) introduce honest causal trees, a method where the data is split to eliminate bias caused by selection in a causal model tree. The first part of the data is used to construct the tree, whereas the second part is used separately to estimate the parameters within the subsets in the leaves. They note that sampling-splitting is essential for empirical work and that it gives an unbiased estimate of the treatment effect. This conclusion confirms Faraway (1998). Our study does not estimate treatment effects, but this method may work very well in a hybrid tree structure like the logit leaf model. A consequence of using honest estimation is that the model is left with a smaller sample to estimate the effects (Berk et al., 2021), this leads to a significant increase in the mean squared error (Athey & Imbens, 2016). The main consideration here is the bias variance trade-off. In section 4.2 we propose a solution to this issue.

2.3 Penalized Logit

Penalized logistic regression with Elastic Net penalties (Zou & Hastie, 2005) is a very popular improvement on logistic regression. The model combines the strengths of Lasso and Ridge regression, to prevent overfitting on the training data and perform variable shrinkage at the same time. Many studies have found that penalized logit outperforms basic logit models such as the one used in the LLM, with forward variable selection (Hastie et al., 2020; Teipel et al., 2017; Pereira et al., 2016; Makalic & Schmidt, 2011). An important drawback of the Elastic Net model is the added computing needed to cross-validate the two hyperparameters.

Penalized logistic regression has been applied in tree-based models. Dumitrescu et al. (2022) propose the penalized logistic regression tree (PLTR), which uses information from decision trees to improve the performance of logistic regression. Rules extracted from various short-depth decision trees built with original predictive variables are used as predictors in a penalised logistic regression model. The model is able to capture non-linear effects that can arise in credit scoring data while preserving the interpretability of the logistic regression model. They find that PLTR outperforms LR in credit risk prediction, and performs similarly to RF. The logit leaf model differs from this model in the way that the PLTR consists of a single logistic regression with predictors from a tree, and the LLM consists of an individual logistic regression for every leaf of a tree.

Zhang & Loh (2014) propose PLUTO, penalized unbiased logistic regression trees. This model is based on logistic model trees, and differs from the logit leaf model as it does not have a hybrid model structure. This model builds the decision tree based on logit regressions, whereas the LLM fits a tree before the logit in the leaves is applied. PLUTO uses Elastic Net penalties and it is able to capture the non-linear effects and interaction patterns. PLUTO controls selection bias by applying an adjusted chi-squared test to find the split variable instead of exhaustive search. A bootstrap calibration technique is employed to further correct selection bias. A drawback of PLUTO is that it excludes categorical predictors from the regression, thus unfortunately we can not include this model in our benchmarking study.

2.4 Categorical Variables

Johannemann et al. (2019) propose sufficient representations for categorical variables. Sufficient means that there is no information lost. They propose three methods, the most useful for our study is group means encoding: for every categorical variable, replace it with variables where value is the average of other continuous variables of observations within the same category. This method is useful when the number of continuous regressors is much smaller than number of categories. They further propose low-rank encoding, when number of continuous regressors is large. It makes use of matrix decomposition: singular value decomposition or PCA. An alternative is sparse PCA by Zou et al. (2006). The third method is encoding by multinomial LR coefficients. It results in the same number of new regressors as means encoding. The research found that the sparse low rank encoding performs best when the number of latent groups is very small. Multinomial and means encoding work better overall. Furthermore, all encoding methods are better for XGboost and Random Forest since it reduces dimensions.

Moeyersoms & Martens (2015) discuss the inclusion of high cardinality categorical variables in a predictive setting for churn. They propose multiple transformation methods, one of which is the *weight of evidence* encoding. Here, a categorical variable is transformed into one continuous variable that represents the ratio of churners of of the observations within the category. The methods are applied to C4.5 tree, Logit and SVM. Performance is measured by true positive rate and AUC. 10-fold CV is applied. They find the performance does not hold up relative to dummy encoding. Pargent et al. (2021) find that regularized versions of target encoding, using target predictions based on the feature levels in the training set as a new numerical feature, outperform traditional encoding methods when dealing with high cardinality categorical variables in different machine learning models such as Lasso, RF, kNN, Gradient Boosting and SVM. We have considered benchmarking the *weight of evidence* method to *means encoding* on our datasets, but we leave this for further research.

3 Data

3.1 Insurance dataset

This research is supplied with a policy dataset from a Dutch insurance company. Exact details of the data, along with the data itself, will be undisclosed. There is no direct churn variable in the dataset, therefore we need to construct it given some other variables. This dependent variable will indicate whether a policy has ended or not in 2021. First we take only policies which were active during 2021. If the policy terminated in 2021, it is classified as a churn. Then we look at the contract duration. If the end date of the contract minus the contract duration (the last renewal date) is in 2021 and after the inception date, the observation is a non-churn. All other observations are removed. This results in a dataset of eighty thousand observations with a churn rate of 11.3 percent.

The vast dataset consists of three main categories of insurance policies with very different churn behaviour. For example, motor insurance regularly has a shorter lifespan than property insurance. To try to better predict policy churn, the Motor and Property policies will be extracted to form separate datasets for this case study, next to the full dataset.

Available variable	Type
previous contract duration	number
age	number
gender	binary
private/business	binary
zipcode	category(90)
payment term	number
line of business	category(5)
policy contract duration	number
contract package	binary
number of coverages in policy	number
yearly premium	number
sum insured	number
number of claims	number
claim to premium ratio	number
MGA id	category(20)

Table 1: Available variables in the insurance company dataset

3.2 Other datasets

Other publicly available churn datasets will also be used in benchmarking tests. These datasets are found on Kaggle, and comprise of various binary classification examples alongside churn prediction. These datasets are used to benchmark models for differing binary classification tasks. This may lead to conclusions for further studies to apply the models on different topics.

Name	Type	#Obs.	Churnrate	#Features	#High cardinal cat. features	Source
Insurance	Insurance churn	80.000	11.3 %	15	1 (90 categories)	Company
Motor	Insurance churn	27.000	28.0 %	15	1 (90 categories)	Company
Property	Insurance churn	33.000	29.0 %	15	1 (90 categories)	Company
Telecom	Telecom churn	4.250	14.1 %	18	1 (52 categories)	Kaggle[2]
Bank	Bank churn	10.000	20.4 %	11	0	Kaggle[3]
Creditcard	Credit card fraud	285.000	0.17 %	29	0	Kaggle[4]
Insurance2	Insurance churn	30.000	13.2 %	15	0	Kaggle[5]

Table 2: Overview of all datasets used in this study

The telecom dataset is made available for a public Kaggle competition. The original source of the data is unknown. This dataset is useful for our study as it is relatively small, so we are able to test how the models are able to handle small datasets. The bank customer churn data, the credit card fraud data and the second insurance dataset are all sourced from Kaggle. These three datasets do not contain high cardinality categorical variables, thus means encoding is not considered. The credit card dataset is very large with more than 285 thousand transactions, of which there are only 492 frauds. The data comprises 29 anonimized numerical features, which are constructed using principal component analysis. This data is very useful to benchmark our models on an entirely different classification task. The second insurance data has similar properties to the company data, aside from not having categorical variables. This data is used to research whether the models' performance on this dataset and on the company data will differ significantly.

[2] https://www.kaggle.com/competitions/customer-churn-prediction-2020

[3] https://www.kaggle.com/code/kmalit/bank-customer-churn-prediction/data

[4] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[5] https://www.kaggle.com/datasets/k123vinod/insurance-churn-prediction-weekend-hackathon

4 Methodology

This section first describes the methods in the current literature related to this study. Next, the methodology for the contribution of this study is explained and reasoned. The Logit Leaf Model (LLM) can be decomposed into two parts. The first part uses a C4.5 decision tree to subset the data into homogeneous groups. This part contains two hyperparameters which need to be tuned using crossvalidation. The second part is ordinary logistic regression. This part uses forward variable selection based on the Akaike Information Criterion. The first part of the model uses hyperparameter optimisation to enhance the accuracy of the entire model. When regarding the second part as a model on its own for estimating the data within the leaves, this is a part that can be improved. The Logistic regression using forward variable selection tends to overfit the data and leads to low predictive accuracy (Song et al., 2013).

4.1 Penalized Logit Leaf Model

To improve the standard LLM, which uses forward variable selection in the logit regression within each leaf, we propose to use penalized logistic regression in the leaves. An Elastic Net places penalties on the size of the parameter estimates to reduce the number of parameters. The Elastic Net model optimizes the following function using maximum likelihood:

$$\begin{aligned}\ell\left(\beta, y_i\right) = &\sum_{i=1}^{n}\left\{y_i \log \pi\left(\tilde{x}_i\right) + \left(1-y_i\right)\log\left(1-\pi\left(\tilde{x}_i\right)\right)\right\} \\ &+ \lambda_1 \sum_{j=1}^{p}\left|\beta_j\right| + \lambda_2 \sum_{j=1}^{p}\beta_j^2\end{aligned} \tag{1}$$

Where $\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij}-\bar{x}_j\right)^2}}$ is the normalization of variables to use penalized regression, and $\pi\left(x_j\right) = p\left(y_i = 1 \mid x_{ij}\right) = \frac{e^{x'_j\beta}}{1+e^{x'_j\beta}}, j = 1, 2, \ldots, p$.

Similar to the Logit Leaf Model, the Penalized Logit Leaf Model (PLLM) first constructs a C4.5 decision tree for a given pruning threshold and minimal leaf size, after which the observations in each leaf are estimated using a logit regression. The performance of the model is assessed in its entirety, meaning that the hyperparameters of both the tree and the logit are tuned based on the performance of the whole model. The use of Elastic Net logit introduces two additional hyperparameters to the model, making the total count four. This generates a problem where the number of possible hyperparameters can get really high, really fast. As the hyperparameters are tuned based on the performance of the whole model, this results in the model having to be fitted thousands of times for only six options per hyperparameter and 5-fold crossvalidation. There is a consideration to be made for the trade-off between optimization and computing time.

Algorithm 1 shows the proposed penalized logit leaf model, and algorithm 2 shows the PLLM with honest estimation applied. See section 4.2 for honest estimation.

Algorithm 1 Penalized Logit Leaf Model

```
Model Creation Phase
Split $D_{\text{tot}}$ into $D_{\text{tr}}$ and $D_{\text{val}}$
Input: training data $D_{tr} = \{(X_i, Y_i)\}_{i=1}^{N}$
Calculate initial decision tree on $D_{tr}$ spanning the total space $S$ using the C4.5 method for tree construction.
Define subspaces $S_t$ from $D_{tr}$ based on a set of terminal nodes $T$ for which $S = \cup_{t \in T} S_t, \forall t \neq t' : S_t \cap S_{t'} = \emptyset$
For $i = 1$ to $T$ do:
    Train an Elastic Net logistic regression model on subset $S_t$.
End for;
Combine results $M_i$ in model $M$
Output: model $M$

Prediction phase
Input: (new) data $D_{val} = \{(X_i, Y_i)\}\,{}_{i=1}^{N}$
Apply decision rules of model $M$ on $D_{\text{val}}$ spanning the total space $S$, resulting in subspace $S_t$ based on a set of terminal nodes $T$ for which $S = \cup_{t \in T} S_t, \quad \forall t \neq t' : S_t \cap\ S_{t'} = \varnothing$
For $i = 1$ to $T$ do:
    Apply model $M_i$ specific for $S_t$
    For $j = 1$ to $n_i$ do:
        Calculate predictions for all $n_i$ instances in $S_t$
    End For;
End For;
Combine predictions
Output: one prediction for every instance in $S$
```

4.2 Honest estimation and data imbalance

When the dependent variable distribution of the data is imbalanced, e.g. the churn rate is low, the predictions of the model will be biased towards the majority class. To remedy this, the training data is undersampled to reduce the size of the majority class to the size of the minority class (Burez & Van den Poel, 2009). Using undersampling to balance the training data results in the loss of a vast part of the dataset if the churn rate is very low. This is especially problematic when the dataset is already small. In the LLM, the training data needs to be balanced for the construction of the decision tree. The decision tree has the purpose to make the subsets in the leaves used for logistic regression as homogeneous, e.g. imbalanced, as possible. For logistic regression (LR) the training data does not need to have classes of equal size (Burez & Van den Poel, 2009). Oommen et al. (2011) even state it is important for LR to keep the class distribution

Algorithm 2 Penalized Logit Leaf Model with Honest Estimation

```
Model Creation Phase
Split $D_{\text{tot}}$ into $D_{\text{tr}}$ and $D_{\text{val}}$
Input: training data $D_{tr} = \{(X_i, Y_i)\}_{i=1}^{N}$
Split $D_{\text{tr}}$ into $D_{\text{tr}-1}$ and $D_{\text{tr}-2}$
Perform undersampling on $D_{tr-1}$ to balance the data.
Calculate initial decision tree on $D_{tr-1}$ spanning the total space $S$ using the C4.5 method
  for tree construction.
Define subspaces $S_t$ from $D_{tr-2}$ based on a set of terminal nodes $T$ for which $S =
  \cup_{t \in T} S_t, \forall t \neq t' : S_t \cap S_{t'} = \emptyset$
For $i = 1$ to $T$ do:
     Train an Elastic Net logistic regression model on subset $S_t$.
Obtain maximum likelihood estimates for the penalized logit leaf.
End for;
Combine results $M_i$ in model $M$
Output: model $M$

Prediction phase
Input: (new) data $D_{val} = \{(X_i, Y_i)\}_{i=1}^{N}$
Apply decision rules of model $M$ on $D_{\text{val}}$ spanning the total space $S$, resulting in subspace
  $S_t$ based on a set of terminal nodes $T$ for which $S = \cup_{t \in T} S_t, \quad \forall t \neq t' : S_t \cap S_{t'} = \varnothing$
For $i = 1$ to $T$ do:
    Apply model $M_i$ specific for $S_t$
    For $j = 1$ to $n_i$ do:
        Calculate predicted probabilities for all $n_i$ instances in $S_t$
    End For;
End For;
Combine predictions
Output: one prediction for every instance in $S$
```

of the sample as close to the original population as possible.

We propose to remedy the loss of data by dividing the training data into two equivalent parts. The first part is undersampled and then used to construct the decision tree. The second part is not undersampled. This data is divided into subsets according to the nodes of the established decision tree, and used as training data for the logistic regressions within the leaves. This method is based on the idea of *Honest Estimation* by Athey & Imbens (2016).

Using less data for estimation to improve accuracy seems paradoxical. However, in the original LLM, the size of the sample within each leaf equals $D_{leaf} = D_{train} * (ChurnRate * 2) * LeafProportion$ due to undersampling. In the proposed method the estimation sample size equals $D_{leaf} = D_{train} * 0.5 * LeafProportion.$ It is clear that when the dataset consists up to 25 percent of the minority class, the size of the training data in the leaves will be larger using the proposed method. Strictly speaking, using this method likely results in a different tree, making this comparison not valid in general. But, as the purpose of a decision tree is to create homogeneous subsets, the minority class size within each leaf will likely be below 25 percent, thus validating this idea. This method is compared to several other cases: the case in which there is a split in training data and both sets are undersampled and the case of no split in training data with undersampling, which is used in the original LLM.

Zheng & Jin (2019) researched the effect of class imbalance and training data size on logistic regression and the logit leaf model. They found that the LLM and LR performances increase faster with respect to boosting methods when the classes become more imbalanced. For the LLM and LR, given the skew is larger than 0.5, that is, the minority class ratio is higher than 33 percent, they find the effect of class imbalance is eliminated through hyperparameter tuning. Lastly, all used models are susceptible to the size of training data, as the maximum performance is achieved with the largest training dataset for all models. We expect the LR and non-honest LLM to achieve higher AUC scores for more imbalanced datasets.

Sowah et al. (2016) propose a novel cluster undersampling technique for class imbalance in C4.5 trees. The method focuses on eliminating the least useful data from the majority class to let the remaining sample contain as much information as possible about the majority class. The cluster undersampling technique outperformed the most used undersampling techniques in terms of AUC performance based on 16 datasets. We have considered this method, but chose to leave this for further research.

Lastly we address a small fix in our benchmarking study. For data with a heavily skewed class distribution, the problem may arise where a sample does not contain any minority class observations. This is prevented by using stratified sampling instead of random sampling, for every sampling occurrence in the model.

4.2.1 Honesty level

The above proposed method uses half of the training data for tree creation and the other half for logit estimation. It may be interesting to see if varying this ratio may lead to an increase in performance. In Athey & Imbens (2016), the option to vary the sample sizes exists, but has not been studied in detail. When honest estimation is applied and the estimation data is not undersampled, increasing the proportion of the estimation sample will increasingly make the logit leaf model more similar to a regular logit model. Thus, we expect similar results of the model with a large sample for estimation and a regular logit model. The train/estimation ratio is tested for the values: 25/75, 30/70, 35/65, 40/60, 45/55, 50/50, 60/40.

We analyse the honest PLLM given these seven ratios on a small number of datasets. The tree training data is undersampled to balance out the data (see section 4.2), the leaf estimation sample remains imbalanced.

4.3 Data preparation

Correctly preparing the dataset for churn prediction can result in an AUC increase of up to 14.5% (Coussement et al., 2017). The methods for data preparation are similar to the approach of De Caigny et al. (2018), with some slight adjustments. First, we make an improvement for the missing value imputation. For variables with more than 5% missing values, they impute missing values. For continuous variables, they use zero, whereas the missing values of categorical variables are an additional category. For each variable for which they imputed missing values, they create a dummy variable to trace back imputation positions. For variables with less than 5% missing values, these observations are removed to reduce the impact of the imputation procedures. In our research, we replace missing continuous values with the mean instead of a zero, but still with an additional dummy indicating the observation has been treated for missing values. As the Kaggle datasets do not contain missing data, missing value treatment only applies to the insurance dataset. Another point where we do not follow De Caigny et al. (2018) is the undersampling of the entire training data beforehand. Instead, we do the undersampling as an additional step within the (P)LLM, to make the model applicable to use a full estimation sample for honest estimation. Lastly, outliers are corrected to be within three standard deviations of the mean, using Winsorization.

4.3.1 Sufficient representations for categorical variables

This research tries to incorporate high cardinality categorical attributes into the model. Since the data contains many categorical variables, which in their place contain many categories such as zip-code, the classical method of encoding group membership as dummy variables will result in an enormous amount of parameters to estimate. There are some options next to traditional dummy encoding, one of which is group means encoding by Johannemann et al. (2019), that we use in our study. There are some drawbacks to the use of this method. In the first case, it removes a large part of the interpretability of the model when categorical variables are encoded into latent variables. Also, the splits in the decision tree may not be able to correctly split the data when a split is on one of the newly constructed variable values. To use sufficient representation in decision trees, we run 5 fold stratified crossvalidation on the datasets to avoid the case where there are categorical variables in the test set which are not contained in the training set.

4.4 Hyperparameter tuning

Classifier	No. models per algorithm	Hyperparameter	Candidate values
Elastic Net	500	α	0, 0.2, 0.4, 0.6, 0.8, 1
		λ	100 lambda's determined by *glmnet*
LLM	36	Confidence threshold for pruning	0.01, 0.05, 0.1, 0.15,0.25,0.3
		Mininal leaf size	n*[0.01, 0.025,0.05,0.1,0.25,0.4]
PLLM	1080	Confidence threshold for pruning	0.01, 0.05, 0.1, 0.15,0.25,0.3
		Mininal leaf size	n*[0.01, 0.025,0.05,0.1,0.25,0.4]
		α	0, 0.2, 0.4, 0.6, 0.8, 1
		λ	0, $1e^{-5}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, $5e^{-2}$, 0.1

Table 3: Hyperparameter settings for this study

Two hyperparameters are added in the PLLM: λ_1 and λ_2 ,see eq. (1). Here we have a few options. We set the hyperparameters equal across all leaf regressions. There is an option to differ the hyperparameter values across parts of the tree or across all individual regressions. However, this option is infeasible when considering that the decision tree in the final model is likely to have a different structure to the trained trees within the validation. Thus, it is not possible to assign hyperparameters to parts of the tree. For the ability to set an honest level (section 4.2.1), an extra hyperparameter is needed. It is interesting to study whether using more hyperparameters would improve the overall accuracy of the model, especially when considering the use of more hyperparameters exponentially increase the computing time and model complexity (Mantovani et al., 2018). The crossvalidation of hyperparameter is done the same way as in De Caigny et al. (2018).

4.5 Benchmark methods

Following De Caigny et al. (2018) we compare the performance of the PLLM to Random Forest, along with the LLM and Elastic Net Logistic regression. Note we do not include DT and LMT as benchmarks, as De Caigny et al. (2018) have proven that these methods are outperformed by the LLM on every dataset.

The Elastic Net regression is tuned using grid search for alpha values {0, 0.2, 0.4, 0.6, 0.8, 1}. For each value of alpha, the `glmnet` function optimizes the lambda parameter using 5-fold crossvalidation. Training data is not undersampled. The Random Forest is also trained using 5-fold cv. The parameter for the number of candidate variables is set equal to the square root of the number of variables, rounded down to the nearest integer. For RF, the training data is undersampled.

4.6 Performance statistics

As a model performance measure, this study uses the area under the receiver operating characteristics curve (AUC). The ROC curve is used to analyse correct and incorrect classification for either balanced or imbalanced data sets (Pagels Fick, 2019). As the validation and test sets consist of heavily skewed class distribution, measuring the performance of models using the area under the ROC curve is very practical (Burez & Van den Poel, 2009). Next to the AUC, a calibration plot gives insight in the probability calibration and the Brier score is used as a performance measure for the estimated probabilities. A well calibrated model has the property that the estimated probability for positives equals the expected rate of positives. The Brier score can be used to quantify this calibration.

Wallace & Dahabreh (2012) argue that the Brier score is unsuitable for imbalanced data. They propose the stratified Brier score, which essentially is assessing the Brier scores of the majority and minority classes separately. This method is useful to check whether a well calibrated model is actually well calibrated for both classes, but it cannot be represented by a single performance measure.

4.7 Model setup

All coding is be done in `R`. For the LLM, the standard source code supplied by the original authors is improved to deal with empty leaves and pure leaves. The original LLM R code[6] deals with pure leaves by regressing the dependent variable on a constant within the leave. We impute a simpler method by setting the probability of all observations within such leaf to 0 or 1. This method is applied on leaves which consist of zero or one observation of a class. This method prevents overfitting on a single odd observation. Next the model is adjusted to deal with empty leaves in various situations. First, it can happen that for a given set over hyperparameters, a tree cannot be built, and the resulting model will be a regular penalized logit model. In this situation, the model terminates and returns an AUC statistic of zero. A second situation can arise with honest estimation, when the estimation subsample in a leaf is empty. Regular honest trees prevent this situation by pruning, but this method is not possible in a hybrid model like the LLM. In this case, the leaf is filled with the resulting leaf sample from the trained tree. This way, the parameters for the leaf can be estimated and predictions for observations in the leaf can be done, thus not having to discard the model.

The `LLM` function is then adjusted to be able to perform undersampling of the class distribution on each of the folds of the cross validation. This is done using a hand-built function. Next,

[6] https://cran.r-project.org/web/packages/LLM/LLM.pdf

the `llm.predict` function is optimized in such a way that it performs up to 15 times faster than the original function.

The train-test set ratio is 70/30, the most commonly used ratio. In all of the models, stratified sampling is used to ensure equal class ratio's across all samples and to avoid pure samples. We perform 5-fold stratified cross validation of the hyperparameters, based on the AUC statistic for all models. The crossvalidation is parallelized by using a future mapping function. This is available for a given number of cores in a pc.

To build the PLLM model, the `glmnet` function is used with the `modified newton` option to improve computation speed. For the possibility of publishing the code as a package, many warning and debugging structures are implemented in all functions.

5 Results

This section presents the findings of the benchmark results of all models on the available datasets.

5.1 Performance analysis

Tables 4 and 5 show the results of the benchmarking experiment, with the best performing models for each dataset underlined.

	Elastic Net	Random Forest	LLM	LLM_{honest}	PLLM	PLLM_{honest}
Motor_D	0.731	0.745	0.690	0.756	0.717	0.723
Motor_M	0.734	0.745	0.705	0.754	0.712	0.767
Property_D	0.835	0.799	0.525	0.830	0.772	0.714
Property_M	0.830	0.808	0.514	0.828	0.784	0.837
Insurance_D	0.742	0.641	0.623	0.730	0.631	0.724
Insurance_M	0.732	0.632	0.617	0.741	0.625	0.766
Telecom_D	0.799	0.893	0.853	0.884	0.841	0.856
Telecom_M	0.798	0.881	0.855	0.863	0.850	0.845
Insurance2	0.879	0.919	0.893	0.893	0.890	0.893
Creditcard	0.980	0.959	0.861	0.981	0.875	0.957
Bank	0.757	0.847	0.841	0.838	0.839	0.848

Table 4: Results of the benchmarking experiment using the AUC performance criterion. *D=dummy encoded, M=means encoded*

	Elastic Net	Random Forest	LLM	LLM_{honest}	PLLM	PLLM_{honest}
Motor_D	0.166	0.297	0.327	0.160	0.293	0.181
Motor_M	0.171	0.295	0.323	0.161	0.292	0.173
Property_D	0.130	0.245	0.278	0.132	0.241	0.193
Property_M	0.129	0.237	0.271	0.132	0.209	0.129
Insurance_D	0.154	0.433	0.403	0.156	0.433	0.162
Insurance_M	0.156	0.434	0.406	0.155	0.431	0.149
Telecom_D	0.099	0.117	0.121	0.075	0.130	0.066
Telecom_M	0.099	0.118	0.117	0.065	0.119	0.064
Insurance2	0.077	0.121	0.144	0.075	0.143	0.071
Creditcard	$5e^{-4}$	0.087	0.129	$4e^{-4}$	0.340	0.013
Bank	0.143	0.145	0.152	0.114	0.152	0.108

Table 5: Brier scores of the benchmarking experiment

Looking at table 4, the honest PLLM scores competitively with the Random Forest and Elastic Net. But more importantly, the regular PLLM outperformed the regular LLM on the insurance company data.

Appendix A shows the visualisation of the honest LLM for the Telecom_D data. This is the best performing logit leaf model for the Telecom dataset. It shows that the logit regressions are very different for each leaf.

Table 4 further shows that the leaf models with honest estimation achieve respectable results, with the honest PLLM outperforming non-honest LLM in almost every case. The benefit of applying honest estimation best showcases itself when looking at the Brier scores in table 5. The honest models perform the best overall, with Brier scores half of their non-honest counterparts for most of the datasets. The Brier scores of RF, regular LLM and regular PLLM for the company datasets are almost all above 0.25. This indicates that these models have worse performance than a bench-sitting model that estimates 0.5 probability for every observation. This seems to be the result some property of the company dataset itself, rather than the fact that the data is imbalanced, as the Kaggle data does not encounter this problem. Although the Brier scores are bad, the AUC for these models are mostly not surprisingly bad. Across the board, the Elastic Net logit performs very steadily. The model's Brier scores are among the best for every dataset, but looking at Figures 12, 13 and 15 it is clear that the Elastic Net logit is ill-calibrated for some data.

Means encoding for categorical variables does not improve the AUC in general, except for the honest PLLM. Surprisingly, the consistently best performing model for the insurance company data is the honest PLLM using means encoding. This result is very promising and confirms a success in improving the Logit Leaf Model. Regarding dummy encoding, the honest LLM consistently outperformed the honest PLLM in terms of both AUC and Brier score. Note that for the Telecom data, means encoding mostly worsened the AUC scores. This may result from the fact that the categorical variable in this dataset has 52 categories, which is under three times the number of variables in the data. However for the insurance company data has a categorical variables with the number of categories equal to six times the number of variables. This result is in line with Johannemann et al. (2019), who state that means encoding only works when the number of categories is significantly larger than the number of variables in the data.

Figure 2 shows the excellent performance of the estimated probability calibration when implementing honest estimation into the LLM (blue) and PLLM (red). The Elastic Net (grey) is also very well calibrated. The poor calibration of the regular LLM (green) and PLLM (orange) is comparable to the RF (purple). In fig. 1, we see a significant dip in the calibration curve of the honest PLLM for the Motor data around 50%. For both the dummy and means encoding, the dip occurs at the same bin of probabilities and the point lies in the direction of the churn rate (28%) which may be due to a large proportion of the data being in this bin. Looking at figure 3, it is clear that this is the case. A remedy for this can be the use of the Brier score for hyperparameter tuning instead of assessing performance based on the AUC. We deal with an imbalanced dataset and when tuning the model based on the AUC, the estimated probabilities

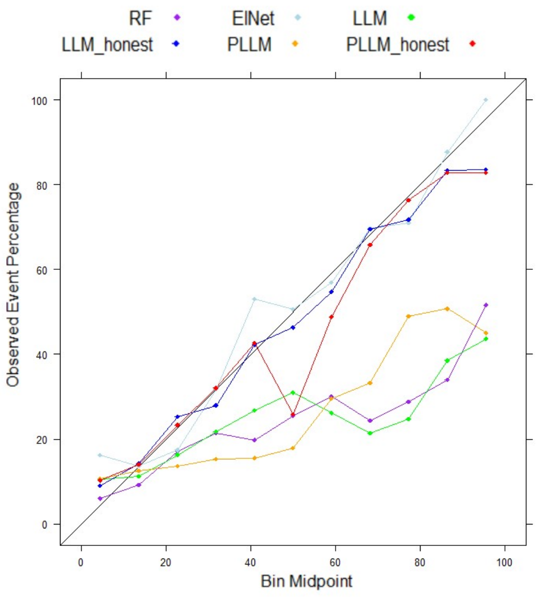

Figure 1: Calibration plot for Motor$_M$

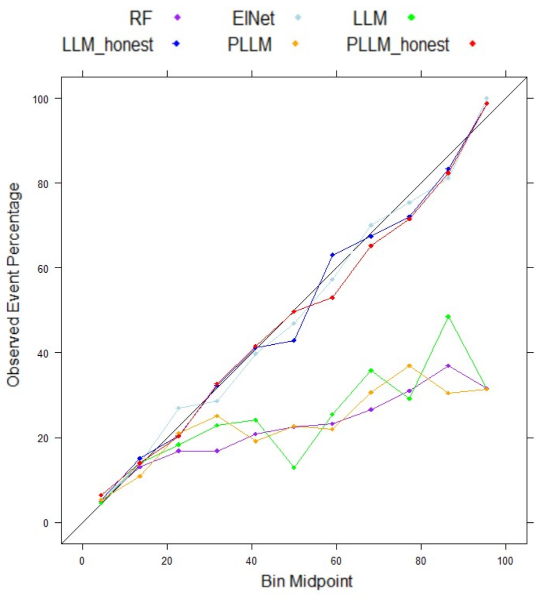

Figure 2: Calibration plot for Insurance$_M$

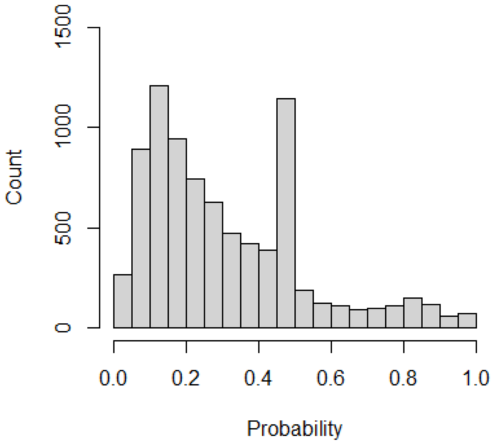

Figure 3: PLLM$_{honest}$ for Motor$_M$: Proportion of observations in bins of probability estimates

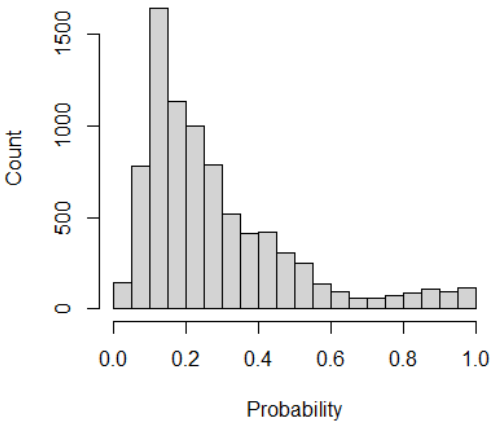

Figure 4: Proportion of observations in bins of probability estimates, using the brier score for hyperparameter crossvalidation

may be left skewed. As the threshold of estimating a positive is at 0.5, the model is likely to optimally train the model to classify positives just over this threshold, to have the majority of the data being correctly classified as negatives just under this threshold. Figure 4 shows the proportions of observations per bin for the same model when being trained using the Brier score. It is clear that in this case, the large proportion of observations at the halfway mark has disappeared. Figure 5 shows that the drop in calibration for this model is not present.

The good calibration of the Elastic Net is in line with literature, as logistic regression naturally returns well calibrated probabilities (Cearns et al., 2020). Tree based methods naturally

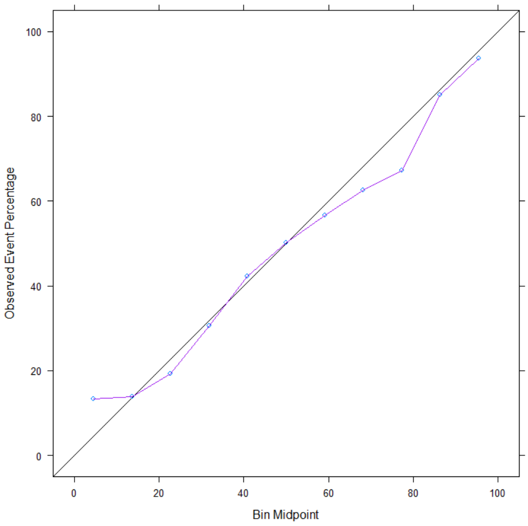

Figure 5: Calibration plot of the honest PLLM on the Motor$_M$ data, using the Brier score for crossvalidation

result in ill-calibrated estimates as the leaf subsets are very homogeneous thus the estimated probabilities are shifted towards zero and one (Zadrozny & Elkan, 2001). The tree based methods are indeed ill-calibrated as we see the RF, LLM and PLLM being underestimated for all datasets. See appendix B for calibration plots of every dataset. Looking at all calibration plots RF, LLM and PLLM models are consistently underestimated towards the majority class for the more imbalanced datasets. This bias is removed in the elastic net and honest models.

5.2 Handling data imbalance for honest estimation

	LLM	PLLM
$Motor_D$	0.682 (0.328)	0.723 (0.297)
$Motor_M$	0.709 (0.322)	0.716 (0.290)
$Telecom_D$	0.851 (0.130)	0.822 (0.180)
$Telecom_M$	0.827 (0.158)	0.826 (0.156)
$Bank$	0.832 (0.160)	0.841 (0.161)

Table 6: AUC results of honest logit leaf models with all training data undersampled. The Brier score is in brackets.

Table 6 shows the results of the honest LLM and honest PLLM where both the training and estimation sample are undersampled. Comparing these results to the non-honest LLM and PLLM (table 4 & 5) respectively, we see that the AUC performance is comparable, but the undersampled honest models have much worse Brier scores.

Taking a critical look, the procedure of undersampling the estimation sample is very illogical in practice. There are two options, one could perform undersampling before or after the data is guided through the tree. As the purpose of the decision tree is to create homogeneous groups, the first option will still result in imbalanced leaf subsets, but with smaller leaves for the majority class. The other option, undersampling the leaf subsets, defeats the purpose of the decision tree to create homogeneous subsets, as well as empties the pure leaves.

The regular honest LLM and PLLM score definitively better than their undersampled counterparts on every dataset, on both AUC and Brier scores. This indicates that it is crucial to leave the estimation sample in honest estimation as-is. Building on this result, it is safe to say that actually, *less is more*.

5.3 Honesty level

Can we improve the honest PLLM even more by adjusting the proportion of the training and estimation samples? We tested this on three datasets where the honest PLLM performed best.

	PLLM$_{25\%}$	PLLM$_{30\%}$	PLLM$_{35\%}$	PLLM$_{40\%}$	PLLM$_{45\%}$	PLLM$_{50\%}$	PLLM$_{60\%}$
$Motor_M$	0.764	0.763	0.768	0.757	0.721	0.767	0.757
	(0.159)	(0.160)	(0.158)	(0.160)	(0.173)	(0.173)	(0.161)
$Property_M$	0.836	0.837	0.838	0.774	0.836	0.837	0.835
	(0.129)	(0.129)	(0.129)	(0.141)	(0.129)	(0.129)	(0.130)
Bank	0.838	0.836	0.831	0.837	0.835	0.848	0.835
	(0.111)	(0.112)	(0.117)	(0.111)	(0.113)	(0.108)	(0.111)

Table 7: AUC results of the honest PLLM with differing sample ratio's for tree construction

Table 7 shows the performance of the honest PLLM using different proportions for the training sample. The best results for each dataset are underlined. The models are applied to datasets where the honest PLLM scored the highest in the benchmarking test of section 5.1. We see that the performance of the already top-of-class honest PLLM can be improved a little by reducing the sample size for tree construction.

5.4 Results comparison with Kaggle competition

For the telecom churn dataset, the highest known AUC score on Kaggle is for a CatBoost model with a score of 0.907, which achieved third place in the competition. Our best performance on this data was an AUC of 0.893, which was achieved by the Random Forest with dummy encoding.

The highest known AUC score for the Insurance2 data on Kaggle is 0.893 using a LightGBM model. This score is equivalent to the AUC of the LLM, honest LLM and the honest PLLM.

6 Conclusion

This study started with the hypothesis that the PLLM would improve the LLM in terms of predictive accuracy. Next we assumed the incorporation of honest estimation would decrease the predictive power due to smaller training samples, but in return would produce unbiased probabilities.

The hypothesis is proven to be right in some way by this research. For many datasets, the honest models also improved their non-honest counterparts in predictive accuracy. Previous research has encountered a similar result for honest ensemble methods (Wager & Athey, 2018), but this is an unexpected result for single tree based methods. We find that both penalized logit and honest estimation are improvements to the LLM. The results show that the combination of the two does not guarantee optimal performance. However, we find that the combination of the two improvements does improve the Logit Leaf Model in every aspect to what it was before. With this "two steps forward, one step back" result, the model is definitely a useful contribution to the current literature.

The results of the models on the credit card data contradict Zheng & Jin (2019), who state that the LR and LLM cannot handle extremely imbalanced data. This paves the way for further research into honest logit leaf models focused on fraud data.

For insurance data, penalized logit based models are superior. For all datasets in general, the honest PLLM outperforms all other models in terms of Brier score, but not always in terms of AUC. Thus, in real world applications, the choice between the models will likely be based on a bias-variance tradeoff.

6.1 Probability calibration

When probabilities are calibrated, the estimated probabilities are in line with the real distribution of the outcome. Then, we can say with some certainty that the estimated probability is the correct probability of an individual for churning. Furthermore we can then predict the total churn rate for a given set of variables very accurately. This is mostly irrespective from the overall prediction performance. In practice this is a very useful property of the new model. This property is only useful to predict the churn rate itself, not for individual specific predictive tasks, as the AUC performance is not necessarily high in this case.

Another unmentioned outcome of the results is that the honest PLLM can be used in risk modelling for insurance companies. Thanks to the high calibration, its ability to handle high number of features and the high interpretability, one can test its ability to assess risk for insurers.

6.2 Answers to the research questions

From the results we can not conclude a single best model for insurance churn prediction. What we can say, is that the models based on penalized logistic regression perform very well across all datasets in terms of both AUC, Brier Score and calibration. The great benefit from both the Elastic Net and the PLLM is the retained model interpretability. This means these models are ready to be applied by business analysts of the insurance company for data-driven decision making.

The improvement of the Logit Leaf Model using Penalized Logistic regression is a success, but with a side note. For most of the data sets, the PLLM is superior over the LLM. But, the extra computing time due to the added hyperparameters in the PLLM causes the model not to achieve its full potential, as we are able to search through only a small number of candidate hyperparameters. As the Elastic Net is much quicker than the other models and less complex, the Elastic Net logit is also a viable option for these type of predictive tasks.

Means encoding for high cardinality categorical variables is a good alternative to dummy encoding, but it does not guarantee improved predictive accuracy. Using means encoding you lose the direct interpretability of the transformed variable, but it is not lost entirely. After model fitment, the continuous values of each category can be retrieved to reinstate the category values into the model. This way the interpretability of the model is never totally lost.

We can conclude that honest estimation drastically reduces the bias for the LLM and PLLM for every dataset. It even does so with an improvement in predictive accuracy for most of the data.

The results further indicate that honest logit leaf models have strengths beyond churn modelling and may be very strong fraud predictors. In the example of fraud detection, the data is extremely unbalanced by nature. The *less is more* principle for honest estimation is very well visible in this case, as the honest estimation improves the AUC by double digits.

Following the results of section 5.3, it is clear that the honest PLLM can further be improved by incorporating an honesty hyperparameter setting the proportion of training and estimation samples. But further research has to find out whether the added computing time for tuning this hyperparameter is worth the slight improvement of model performance.

6.3 Notes on limitations and further research

This research was limited by the small computing power of the company laptop. When using faster computers, the parameter tuning can be extended with more parameter options. This may result in better performance of the PLLM. Data preparation is mostly kept similar to the original LLM study. One can test more advanced data preparation methods to improve model performance. If the PLLM will be used on data that may contain a large proportion of outliers, one can try to further improve the model by using a robust estimator for the Elastic Net regression. A robust estimator eliminates bias caused by outliers. For example, a Mallows type M estimator is suited for a GLM like binary logistic regression (Cantoni & Ronchetti, 2001). Algamal & Lee (2015) applied correlation based Elastic Net penalties to binary logistic regression. They found that for high dimensional data, it outperformed Elastic Net, Lasso and Ridge regression. We have considered studying the performance of this method in a hybrid tree structure like the LLM. However, due to implementation issues in our programming language, we leave this for further research. As we conclude honest logit leaf models are a significant improvement to the current literature, more research is needed to compare these models to honest ensemble trees. Finally we suggest to research an bagged edition of the honest PLLM. And even though the model interpretability will be lost, this has the potential to result in very high performance.

References

Algamal, Z. Y., & Lee, M. H. (2015). Applying penalized binary logistic regression with correlation based elastic net for variables selection. *Journal of Modern Applied Statistical Methods*, *14*(1), 15.

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360.

Berk, R., Olson, M., Buja, A., & Ouss, A. (2021). Using recursive partitioning to find and estimate heterogenous treatment effects in randomized clinical trials. *Journal of Experimental Criminology*, *17*(3), 519–538.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626–4636.

Cantoni, E., & Ronchetti, E. (2001). Robust inference for generalized linear models. *Journal of the American Statistical Association*, *96*(455), 1022–1030.

Cearns, M., Hahn, T., Clark, S., & Baune, B. T. (2020). Machine learning probability calibration for high-risk clinical decision-making. *Australian & New Zealand Journal of Psychiatry*, *54*(2), 123–126.

Coussement, K., Lessmann, S., & Verstraeten, G. (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. *Decision Support Systems*, *95*, 27–36.

Coussement, K., Phan, M., De Caigny, A., Benoit, D. F., & Raes, A. (2020). Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. *Decision Support Systems*, *135*, 113325.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, *269*(2), 760-772. doi: https://doi.org/10.1016/j.ejor.2018.02.009

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, *297*(3), 1178-1192. doi: https://doi.org/10.1016/j.ejor.2021.06.053

Faraway, J. J. (1998). Data splitting strategies for reducing the effect of model selection on inference. *Comput Sci Stat*, *30*, 332–41.

Faraway, J. J. (2016). Does data splitting improve prediction? *Statistics and computing*, *26*(1), 49–60.

Hastie, T., Tibshirani, R., & Tibshirani, R. (2020). Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. *Statistical Science*, *35*(4), 579–592.

Johannemann, J., Hadad, V., Athey, S., & Wager, S. (2019). Sufficient representations for categorical variables. *arXiv preprint arXiv:1908.09874*.

Makalic, E., & Schmidt, D. F. (2011). Logistic regression with the nonnegative garrote. In *Australasian joint conference on artificial intelligence* (pp. 82–91).

Mantovani, R. G., Horváth, T., Cerri, R., Junior, S. B., Vanschoren, J., & de Carvalho, A. C. P. d. L. F. (2018). An empirical study on hyperparameter tuning of decision trees. *arXiv preprint arXiv:1812.02207*.

Markapudi, B., Latha, K. J., & Chaduvula, K. (2021). A new hybrid classification algorithm for predicting customer churn. In *2021 international conference on innovative computing, intelligent communication and smart electrical systems (icses)* (pp. 1–4).

Moeyersoms, J., & Martens, D. (2015). Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector. *Decision Support Systems*, *72*, 72-81. doi: https://doi.org/10.1016/j.dss.2015.02.007

Nurrohman, A., Abdullah, S., & Murfi, H. (2020). Parkinson's disease subtype classification: Application of decision tree, logistic regression and logit leaf model. In *Aip conference proceedings* (Vol. 2242).

Oommen, T., Baise, L. G., & Vogel, R. M. (2011). Sampling bias and class imbalance in maximum-likelihood logistic regression. *Mathematical Geosciences*, *43*(1), 99–120.

Pagels Fick, S. (2019). Will you stay or will you go? churn prediction for an app-delivered international calling service. *Thesis*.

Pargent, F., Pfisterer, F., Thomas, J., & Bischl, B. (2021). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *arXiv preprint arXiv:2104.00629*.

Pawełek, B., & Pociecha, J. (2020). Corporate bankruptcy prediction with the use of the logit leaf model. In *Studies in classification, data analysis, and knowledge organization* (pp. 129–146). Springer International Publishing. doi: 10.1007/978-3-030-52348-0_9

Pereira, J. M., Basto, M., & da Silva, A. F. (2016). The logistic lasso and ridge regression in predicting corporate failure. *Procedia Economics and Finance*, *39*, 634–641.

Song, L., Langfelder, P., & Horvath, S. (2013). Random generalized linear model: a highly accurate and interpretable ensemble predictor. *BMC bioinformatics*, *14*(1), 1–22.

Sowah, R. A., Agebure, M. A., Mills, G. A., Koumadi, K. M., & Fiawoo, S. Y. (2016). New cluster undersampling technique for class imbalance learning. *International Journal of Machine Learning and Computing*, *6*(3), 205–214.

Teipel, S. J., Grothe, M. J., Metzger, C. D., Grimmer, T., Sorg, C., Ewers, M., ... others (2017). Robust detection of impaired resting state functional connectivity networks in alzheimer's disease using elastic net regularized regression. *Frontiers in aging neuroscience*, *8*, 318.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242.

Wallace, B. C., & Dahabreh, I. J. (2012). Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 ieee 12th international conference on data mining* (pp. 695–704).

Whittingham, M. J., Stephens, P. A., Bradbury, R. B., & Freckleton, R. P. (2006). Why do we still use stepwise modelling in ecology and behaviour? *Journal of animal ecology*, *75*(5), 1182–1189.

Xong, L. J., Xong, L. J., & Kang, H. M. (2019). A comparison of classification models for life insurance lapse risk. *International Journal of Recent Technology and Engineering*, *7*(5), 245–250.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml* (Vol. 1, pp. 609–616).

Zhang, W., & Loh, W.-Y. (2014). Pluto: Penalized unbiased logistic regression trees. *arXiv preprint arXiv:1411.6948*.

Zheng, W., & Jin, M. (2019). Effects of training data size and class imbalance on the performance of classifiers. In *Conference on artificial intelligence and natural language* (pp. 3–17).

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301–320.

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, *15*(2), 265–286.

Appendix

A Logit leaf model visualization

Segment	1	2	3	4
Rule1 Rule2 Rule3	Var.A <= 244.9 Var.B <= 3 Var.C =0	Var.A <= 244.9 Var.B <= 3 Var.C > 0	Var.A <= 244.9 Var.B > 3 . . .	Var.A > 244.9 . . . . . .
LR Coefficients	(Intercept) -11.41 . . Var.D 0.1 Var.E -135.6 Var.A 23.07 . . . . Var.H 0.1 Var.K 5.35 Var.L -4.35 Var.M -15.45 Var.N -15.21 Var.R 0.13 Var.T -118.69	(Intercept) -6.14 . . Var.D 0.17 . . . . . . Var.G 1.84 . . . . Var.L -6.89 Var.P 16.88 Var.Q 2.95 Var.W 18.45 . .	(Intercept) 14.56 Var.C 5.89 . . Var.E -0.28 . . Var.F -0.03 Var.G -242.32 Var.H 65.16 Var.K -0.01 . . Var.B 1.35	(Intercept) -64.74 Var.C 3.58 . . Var.E -185.19 Var.A 31.64 Var.F 0.08 . . . . Var.K 0.04 Var.L -8.32 . . . .

Table 8: Visualization of the honest Logit Leaf Model on the Telco$_D$ dataset. Variables are renamed for better readability.

B Calibration plots

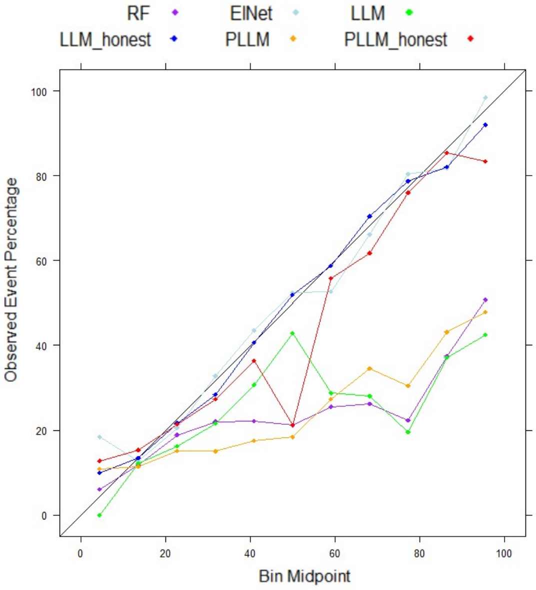

Figure 6: Motor$_D$

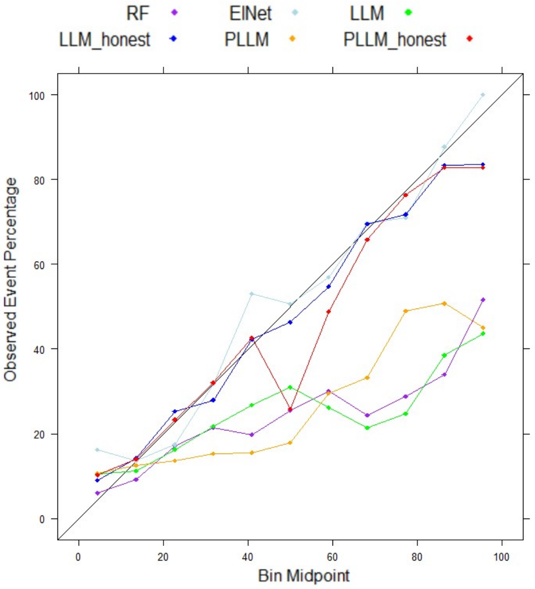

Figure 7: Motor$_M$

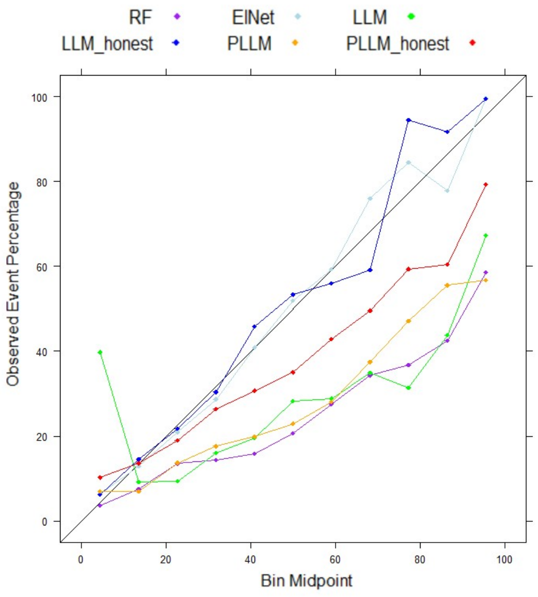

Figure 8: Property$_D$

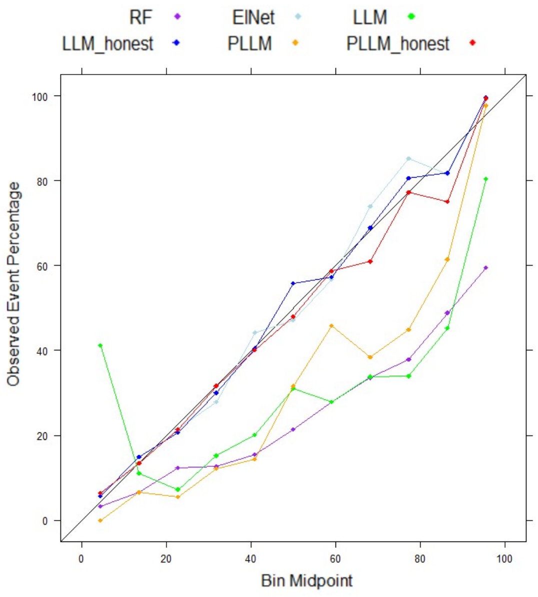

Figure 9: Property$_M$

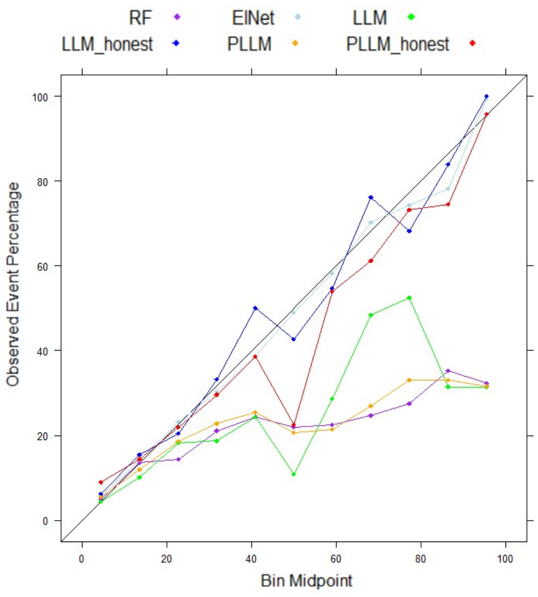

Figure 10: Insurance1$_D$

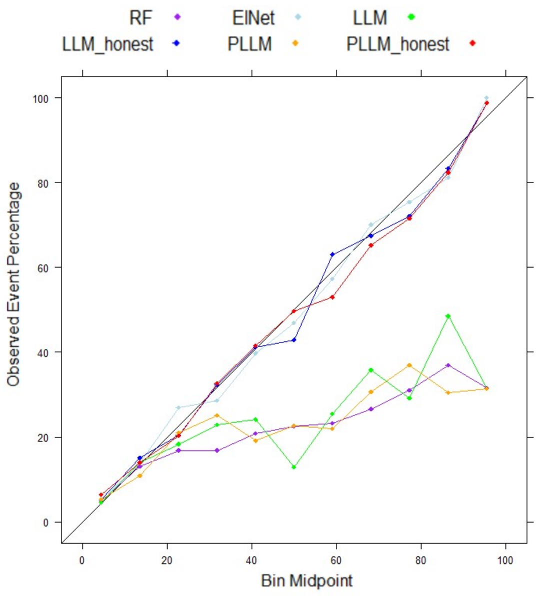

Figure 11: Insurance1$_M$

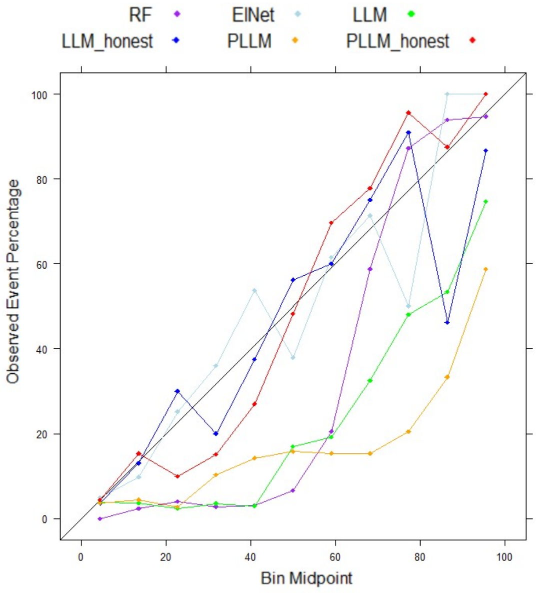

Figure 12: Telecom$_D$

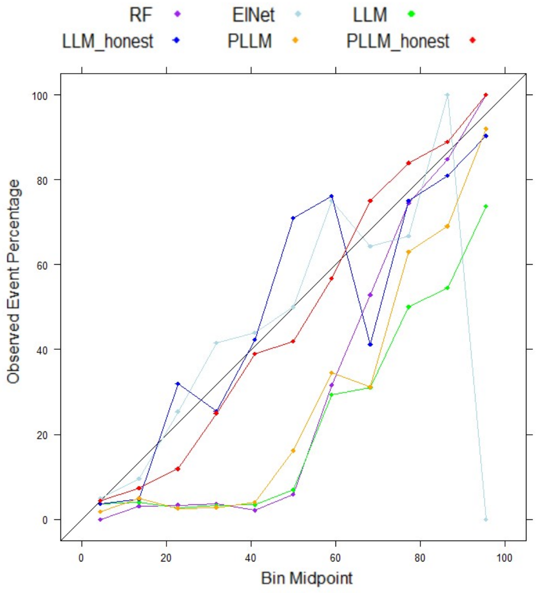

Figure 13: Telecom$_M$

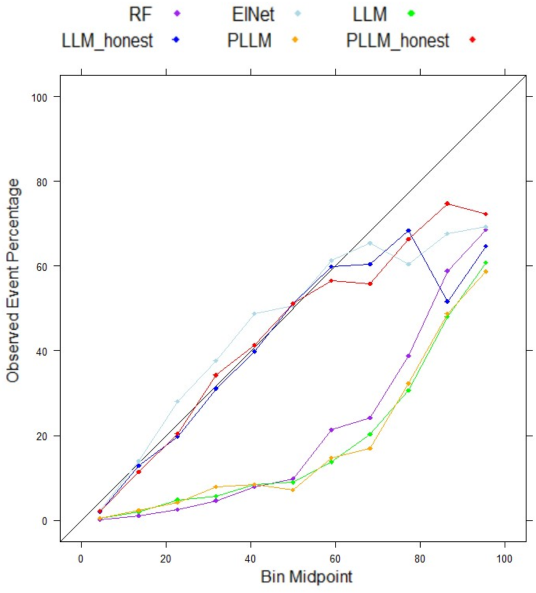

Figure 14: Insurance2

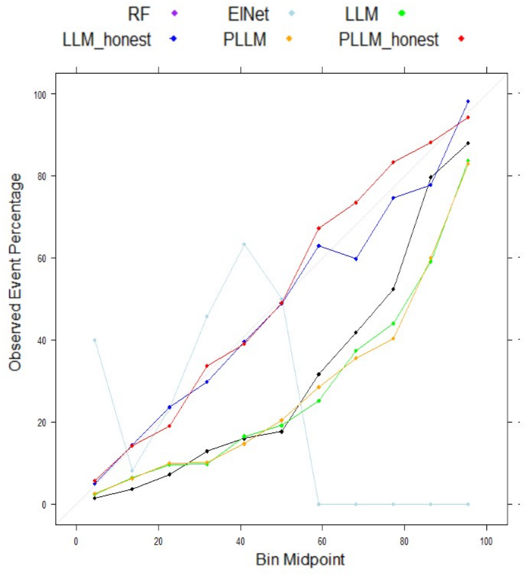

Figure 15: Bank