Exploring Pseudo-Labeling for Reject Inference

Margarida Martins

Dissertation written under the supervision of professor Susana Brandão

Dissertation submitted in partial fulfilment of requirements for the MSc in Business Analytics, at the Universidade Católica Portuguesa, Jan 2024.

Exploring Pseudo-Labeling for Reject Inference

Margarida Martins

Jan 2024

Supervisor: Prof. Susana Brandão

Abstract

Banks use algorithms to estimate the credit risk of loan applicants. However, we need to retrain these models. When retraining, we only know the label, meaning whether the applicant defaulted or not, for those accepted for the loan. Retraining only with the accepted will result in biased models and losses for the bank due to selection bias. To counteract this issue, we can infer the labels of those rejected. This is known as reject inference. In this thesis, we will pursue pseudo-labeling to do reject inference, which needs two models, the first to create the pseudo-labels for the rejected and the second to make the final predictions. We will create the pseudo-labels by training a lightGBM on the available data. Afterward, we will apply a logistic regression as the final model. We will compare the results against a baseline, setting all rejected to a category (default /not default). In addition, we will compare to a scenario where the rejection results from random decision-making, experiment five rejection rates, and see the effect of setting to default vs. not default. We found that doing lightGBM to infer the labels had a lower F1 score, AUC, and profit for the bank. As such, the bank should set all rejected to a category. Additionally, we found that setting all to default has a higher recall in the rejected population and higher profit. Moreover, a lower rejection rate increases profits.

Keywords: Machine Learning, Pseudo-Labeling, Reject Inference, Selection Bias

Exploring Pseudo-Labeling for Reject Inference

Margarida Martins

Jan 2024

Supervisor: Susana Brandão

Resumo

Os bancos usam algoritmos para estimar o risco de crédito dos candidatos a empréstimos. No entanto, esses algoritmos necessitam de ser novamente treinados, mas para tal, é preciso possuir dados históricos com etiqueta. Neste caso, é necessário ter uma variável que indique se o candidato cumpriu na totalidade o pagamento do empréstimo. Nesta circunstância, só conhecemos a etiqueta de candidatos que foram aprovados para empréstimo. Ao treinar novamente apenas com estas observações, o modelo irá ser enviesado, resultando em perdas monetárias para o banco. De forma a impedir tais perdas, tentaremos apurar as etiquetas dos candidatos rejeitados. Nesta tese, iremos usar "pseudo-labeling" para inferir esta etiqueta. "Pseudo-labeling" funciona tendo dois modelos. Primeiro, criar-se-á "pseudo-labels" ao treinar o modelo "lightGBM". Após, iremos aplicar regressão logística. No final, estes resultados serão comparados com o cenário de classificação de duas categorias, analisando ambas. Concomitantemente, iremos comparar com o cenário da decisão de rejeição inicial resultante do acaso e experimentar cinco taxas de rejeição sobre a regressão logística. Ao usar o "lightGBM" obteve-se um "F1", "AUC" e lucro inferior. Como tal, o banco deverá classificar os rejeitados em uma das categorias. Sucede que se descobriu que classificar os rejeitados como incumpridores tem um "recall" superior na população rejeitada e leva a um lucro superior. E que uma taxa de rejeição inferior tem um lucro superior.

Keywords: Machine Learning, Pseudo-Labeling, Reject Inference, Selection Bias

Acknowledgements

I would like to thank Prof. Susana Dias Brandão for always being available and for helping me write this thesis. I learned so much from her. I am also thankful for the usefulness of the knowledge I acquired during this thesis.

I would like to thank all the professors of the master's degree for giving me the knowledge necessary to make this thesis possible.

I would also like to thank my family and friends for supporting and encouraging me to push through all the setbacks.

Contents

List of Tables

List of Figures

Glossary

MNAR - Missing Not At Random

MAR - Missing At Random

Log - Logistic

LGBM - lightGBM

1 Introduction

Like most organizations, banks' main goal is to generate profits. They achieve this by granting loans as they will earn interest on the borrowed money. However, the bank can only grant loans to some loan applicants. The bank will incur a loss if the borrower does not repay the loan. As such, the bank must assess the credit risk, meaning the likelihood that the loan applicant will not repay the loan in the agreed timeline. Currently, it is possible to use machine learning to estimate credit risk and decide whether to accept and grant a loan or reject it. Despite that, the bank would have to retrain these models as the conditions in which these loans take place change, making it necessary to update. Nonetheless, when attempting to retrain, a problem will arise. This problem occurs because historical labeled data is needed to retrain the models. In this specific case, the natural outcome of the loan borrower needs to be known.

The natural outcome is whether the borrower defaulted or not. Unfortunately, this outcome is only known for those loan applicants who were accepted and granted loans. As for the rejected part, the natural outcome never actually occurred. Nevertheless, if the model is retrained only with the accepted part of the population, then the estimators will be biased. We expect the rejected group's distribution to differ from the accepted distribution, as the previous model rejected those loan applicants because they were likely to default. Additionally, if we do not train the model with the rejected part of the population, it could fail to find the previous patterns that led to the rejection of these loan applicants.

Estimating the labels of the rejected part of the population is known as reject inference. We will explore pseudo-labeling as a way of doing reject inference. Pseudo-labeling works by having two models. We will train the first model with the labeled, accepted data to predict the rejected part. Then, we will store these predictions as pseudo labels along with the labels for the accepted part of the population. Finally, a second model is trained with all the data and the pseudo labels to make predictions for unseen data. In our thesis, we will explore three ways of creating the first model in pseudo-labeling. The three ways of doing pseudo labeling are the following: 1) use as the first model, lightGBM, to create the pseudo labels and then apply a logistic regression; 2) set as pseudo labels all those rejected to default and then apply a logistic regression and lastly 3) set as pseudo labels all those that had been rejected not to default and then apply a logistic regression. The main goal of this thesis is to investigate the impact of using a model such as lightGBM to create pseudo labels versus setting all the rejected to a specific category. However, the rejection rate, meaning the percentage of loan applicants rejected by the final model (logistic regression), also influences pseudo-labeling performance. As such, in our thesis, we will additionally explore five rejection rates to guide banks' decision-making. Furthermore, given that in a missing not-at-random case, the distribution of the two groups rejected vs. accepted are different, while in the missing at random, it is not. We will compare our results

against the baseline of having the prior models' decisions result from random decision-making.

The research question I will answer is the following:

- What is the impact of using lightGBM to create pseudo labels versus setting all rejected to a category?

In this thesis, we will use machine learning to assess credit risk. We will start by exploring the available research on credit risk, reject inference, pseudo-labeling, and other relevant topics. As well as investigate the challenges of our approach versus rival approaches. We will then implement the models in the two scenarios with the five rejection rates with a Kaggle data set containing characteristics of loan applicants. Next, we will analyze the results of the six models, and lastly, we will present the implications of our research and future work.

2 Literature Review

2.1 Credit Risk

2.1.1 Credit risk and the use of machine learning methods to assess the risk

When banks decide whether to grant loans, whether personal, home, or corporate, they must assess the credit risk. Credit risk is the probability of default, meaning the likelihood that the borrower will fail to repay the credit at the time that was agreed upon (Crook and Banasik (2004)). Banks make this decision based on the characteristics of loan applicants. (Ehrhardt et al. (2021)).

Due to the positive evolution of computing power, big data, and data availability, it is possible to use machine learning models to assess loan default probability, assigning a credit score to each entry (Addo et al. (2018)). These models' main advantage over human decisions is that they can make quicker rulings (Maldonado and Paredes (2010)).

These algorithms must access the probability of default, also known as a score (Khandani et al. (2010)), and then define a cut-off threshold. If the score is below a certain threshold, it will be assigned to accept, as it is likely not to default; however, if it is above the cut-off point, it will be assigned to the rejected subgroup of the population, as the risk of default is high according to the model.

2.1.2 Ethical Issues

However, the data the model faces is not balanced, as both defaults versus non-defaults and rejected versus accepted ratios vary depending on several factors and might not appear in the

50/50 (Addo et al. (2018)). Additionally, the value of decisions also depends heavily on data quality (Addo et al. (2018)).

It is important to note that these decisions directly impact the economy and social world (Addo et al. (2018)). Financial distress for both banks and clients can arise from poor decision-making.

Data privacy is also a concern, especially when a machine decides, not a human being. Having the decision being made by an algorithm is a source of affliction for some people (Addo et al. (2018)). Therefore, we must be aware of the limitations of these algorithms and try to avoid possible discrimination that the models may find. Hence, it is crucial to be as transparent as possible in decision-making (Addo et al. (2018)).

2.2 The need to retrain

Nevertheless, these algorithms are not static (Wu et al. (2020)). Environments in which banks grant loans change with economic and financial factors. For example, the environment in which someone would find themselves now applying for a home loan is entirely different from the one they would find if they applied in 2019 when interest rates were still low. Consequently, a model working correctly in 2019 must be updated to consider these changes.

Moreover, along with the environmental changes mentioned in the previous paragraph, a significant amount of data will be added (Wu et al. (2020)). It is then necessary to update previous models. When attempting to retrain, the data scientist team would face a problem as there is selection bias because only the accepted part of the population is labeled and, therefore, available for training.

2.3 Reject Inference

This problem arises because the sample on which we will train the model differs from the population on which the model will feed when deployed (Ehrhardt et al. (2021)). The reason behind this is that the natural outcome, whether it defaulted or not, is only known for the part of the population that the prior model accepted. For those whom the model rejected, it is unknown whether they would default, as that never occurred. If a model were trained by the bank using only the part of the accepted population and therefore labeled, the sample would be biased. More specifically, with selection bias (Maldonado and Paredes (2010)). Additionally, this would mean that some characteristics between applicants would be more prevalent than others (Crook and Banasik (2004)). It could also happen that previous decisions were made based

on variable/s that could be either no longer present or no longer relevant (Crook and Banasik (2004)). As such, it is impossible to use a model, for example, logistic regression, using only the accepted subgroup because there will be biased parameters.

Another problem worth mentioning is that the data are not missing at random in most cases (Ehrhardt et al. (2021)). A part of the data is unlabelled because a previous model found that the probability of defaulting met some threshold, and therefore, they were likely to default. However, in some specific cases, it could be possible that the bank made the decision at random, but that case is different than what we expect to find.

Some researchers consider this issue a selection bias problem and an omitted variable problem (Banasik and Crook (2007)) because, as previously mentioned, some characteristics could be more or less present due to the biased sample.

In addition to the apparent statistical issue, why do banks need to do reject inference? It matters because the bank could lose money by not taking advantage of the rejected subgroup. After all, not all accepted would default, and not all rejected would default (Maldonado and Paredes (2010)). By minimizing those that were rejected and would not default and maximizing those that were accepted and would not default, the bank would maximize its profits. Another vital aspect to note is that the previous model decided to reject or not a part of the population based on patterns it found in the training data. However, our new model will no longer find these patterns if we do not feed the rejected part of the population to it. As such, the new algorithm could accept these, and the bank will be subject to losses as they are likely to default.

To counteract this issue, reject inference can be done. Reject inference occurs when the labels for the part of the population that was rejected and are therefore missing are inferred (Maldonado and Paredes (2010)). If we do reject inference correctly, we can turn unlabelled data into labeled data, and selection bias would no longer be an issue.

It is essential to consider that reject inference is not expected to be 100 % accurate. If the natural outcome never occurred, then it is impossible to say with 100 % certainty that a specific applicant would fail to repay the loan.

There are some cases in which reject inference is more critical than others. When there are low to medium approval rates (high rejection rate) and low approval rates (low default rate), or even in cases where the model is performing well but could be better (Maldonado and Paredes (2010)), then reject inference would be optimal in these cases. Moreover, if the proportion of defaults/non-defaults differs significantly depending on whether they belong to the accepted

or rejected subgroup, then reject inference would yield better results (Maldonado and Paredes (2010)).

Nevertheless, in cases with high confidence and high approval rates (low rejection rates) or cases with low confidence, but the decision was random (Maldonado and Paredes (2010)), reject inference would be less valuable. As in the second case, it is now a missing at-random case, and in the first case, the model would learn very little from doing reject inference.

Furthermore, in cases where the characteristics for those rejected are not available, reject inference would not be possible (Mancisidor et al. (2020)).

One paper, Crook and Banasik (2004), found that doing reject inference slightly differed in performance for the new model. Furthermore, the importance of reject inference also depends on the rejection rate(Crook and Banasik (2004)). In cases of high rejection rates, doing reject inference had modest improvements. Moreover, with a low rejection rate, the improvement is even more minor (Crook and Banasik (2004)). As such, in this thesis, instead of using the regular 0.5 cut-off threshold, we will select n % of observations with the highest scores to be predicted as default to investigate the rejection rate that provides the best performance on the chosen KPIs. In addition, we will test several n values to find the optimal rejection rate.

Finally, to better understand the issue, it is essential to note that in a real-world scenario, it is vital to find the scores and determine the amount of credit to give each applicant (Banasik and Crook (2007)). Although we know it is crucial, this will not be further studied in this thesis as we are aware of the amount of credit for each loan applicant in our case.

2.3.1 Semi-Supervised Learning

We can teach machine learning models in different ways. Some of the most known are supervised and unsupervised learning. Labeled data is used to train supervised learning models, whereas in unsupervised learning, we use unlabelled data to train the models. Semi-supervised learning is a technique that exists between the two because it uses both labeled and unlabelled data (Kim et al. (2023)). This idea is worth exploring as labeled data are scarce, while unlabelled data are abundant (Kim et al. (2023)).

Additionally, obtaining labeled data requires financial costs and human resources (Kim et al. (2023)). Therefore, semi-supervised learning that utilizes unlabelled data would relax human supervision (Arazo et al. (2020)). In essence, semi-supervised learning is a potential method for reject inference.

One issue with Semi-Supervised Learning is that some models assume that labeled and unlabelled data come from the same distribution(Ehrhardt et al. (2021)). However, this is different for home loans. The characteristics of the accepted population differ from those of the rejected population (Ehrhardt et al. (2021)). Additionally, semi-supervised learning works best in cases where unlabelled data exist in a larger quantity than labeled data. A high quantity of unlabelled data is only sometimes valid in our case, as it depends on the bank, and economic state, among others. Furthermore, paper Ehrhardt et al. (2021) shows that the semi-supervised learning models perform worse as the rejection rate increases. Moreover, most semi-supervised learning models also assume that the data is missing at random, which is not the case (Ehrhardt et al. (2021)).

Semi-supervised learning has mainly been used for image classification problems and is not used extensively in tabular data (Kim et al. (2023)). The lack of use in tabular data is critical because images and videos are homogeneous, whereas tabular data are heterogeneous (Kim et al. (2023)). Fundamentally, semi-supervised learning will be more challenging to apply to tabular data. The success of semi-supervised learning will depend heavily on the composition of the data set, for example, feature types and the imbalances they may find, depending on the label (Kim et al. (2023)). Paper Cascante-Bonilla et al. (2021) also mentions how semi-supervised learning has had great results in image classification but still needs to be used more for tabular data.

The semi-supervised learning also needs to ensure the cluster assumption, meaning that two data points in the cluster should have the same label (Yang et al. (2022)). Another critical assumption in semi-supervised learning is the low-density assumption, which states that the decision boundary must be in a low-density region (Yang et al. (2022)). However, most semi-supervised learning would perform poorly in overlapping Gaussian distributions, where the decision boundary will now be a dense region (Zhu (2005)). Additionally, the manifold assumption states that if two points are closely located in a cluster but in the low-dimensional region, they should have the same label (Yang et al. (2022)).

2.3.2 Transductive support vector machines

We attempt to optimize the gap between the decision boundary and data points in transductive support vector machines. We increase this gap by using the distance of the unlabelled data to the boundary (Yang et al. (2022)). However, these models might only work if the low-density assumption holds (Zhu (2005)). Therefore, in a complex case, as in loan defaults where trying to identify patterns and characteristics is unclear, the low-density assumption would pose

a problem. Due to not being able to ensure this assumption, we will further investigate other methods.

2.3.3 Pseudo-Labelling

Pseudo-labeling is the most relevant method we found to do rejected inference as it is a method that works by having two models. One will be the student, while the other will be the teacher (Pham et al. (2021)). The teacher generates pseudo labels on the unlabelled data by using the labeled data to learn. Then, the pseudo labels are combined with the labeled data and used to train the students who will make predictions for unseen data. We can use it for structured data. In addition, it can work with models that are not neural networks, which would relax the assumptions mentioned in the semi-supervised learning section and ensure the transparency needed to lessen the ethical concerns.

Despite fitting well with the problem in theory, the main problem with pseudo-labeling is that the teacher might generate incorrect labels. Then, the student is going to learn from these inaccurate predictions. (Pham et al. (2021)). Overfitting to these incorrect labels is known as confirmation bias (Arazo et al. (2020)).

Several solutions have been proposed to address the confirmation bias. The following are some of the most exciting findings:

- In paper Arazo et al. (2020), they attempted to tackle this issue using augmentation and dropout. They also used soft labels instead of complex ones, claiming that this outperformed other models. They also stated that reducing the confidence of the pseudo-labels reduced confirmation bias. They showed that oversampling with labeled data reduces confirmation bias and improves generalization. They showed that dropout with augmentation reduces confirmation bias. However, the data was unstructured in their case, which is not the case here. We will use undersampling, a method similar to oversampling, to reduce the confirmation bias in this thesis.
- In paper Kim et al. (2023), the scores provided by pseudo-labels do not usually ensure the cluster assumption. As we have said before, this assumption states that two data points that are close together will belong to the same cluster and, as such, the same class. In addition, the decision boundary should be in the low-density region. In essence, a pseudo-label in a high-density region is more reliable, but these are not necessarily those with high confidence scores. Additionally, pseudo-labelling needs help identifying whether the pseudo-labels are in high-density regions. They use curriculum labeling, meaning they use percentile scores instead of a fixed threshold to decide the label. They show that

the performance increases using curriculum labeling, similar to pseudo-labeling with regularized confidence. Curriculum labeling also ensures the cluster assumptions. However, this depends on the data. Based on this paper, we will reject or accept a loan applicant based on a certain percentage of the highest predicted scores.

- In paper Rizve et al. (2021), they discussed how pseudo-labeling also makes incorrect decisions with poorly calibrated networks. However, they found that including low- and high-confidence predictions improves generalization. Despite this, we will not use neural networks in this thesis, but all predictions from lightGBM, with low and high confidence, will be kept.

In this thesis, we will use percentages and undersampling to prevent confirmation bias.

Two other important things to note about pseudo labeling are that if the proportion between default/not default differs significantly between the two subgroups of the population, then reject inference provides better results using all data (Maldonado and Paredes (2010)). Furthermore, pseudo-labeling can be expensive with many variables (Maldonado and Paredes (2010)).

2.3.4 Incremental Learning

Another way to do reject inference is to use incremental learning. Continual learning occurs when a model keeps learning from an infinite data stream (De Lange et al. (2021)). The goal is to acquire and update knowledge continuously. Despite that, these models suffer from catastrophic forgetting, as the performance of previously learned tasks will continue to deteriorate as new knowledge is added (De Lange et al. (2021)). In addition, it is expensive to store all data and update the model (De Lange et al. (2021)). Given the potentially high costs and catastrophic forgetting, in this thesis, we will continue to pursue pseudo-labeling.

2.3.5 Label Noise

Another possibility is to use the concept of label noise. In the paper Natarajan et al. (2013), the writers attempted to determine the impact of random classification noise on the accuracy of machine learning models. They found that even with 40 percent of corrupted labels, they still had an accuracy of over 88 percent. However, in this case, the labels are not corrupted. They are missing, and not randomly so.

In paper Song et al. (2022), they showed that the difference between the accuracy of the test data when using clean data and noisy data is significant even with regularization techniques.

These two papers show that there will be a different impact depending on whether the noise is random or not random. As such, in this thesis, we will further inspect the impact on the key performance indicators chosen depending on whether the rejection decision is random.

2.3.6 Control Groups and Augmentation

Another option would be to experiment with control and treatment groups (Ehrhardt et al. (2021)). However, this would purposely require the bank to lose money. It would also have an ethical concern, as loans would be granted to loan applicants who are likely to default, potentially causing financial distress. Therefore, neither the bank nor the regulators of banks would find this option acceptable.

Another option is augmentation and sample selection (Banasik and Crook (2007)). Sample selection could help include variables that would otherwise not appear(Banasik and Crook (2007)). In addition, augmentation would synthesize a sample to represent the rejected sample (Banasik and Crook (2007)). However, the same paper found that augmentation and sample selection had negligible benefits, both in cases where they were applied together and separately (Banasik and Crook (2007)). Due to the negligible effects, it is not an option worth exploring.

2.4 LightGBM:

In order for pseudo-labeling to work, two models have to be defined. Paper Addo et al. (2018) compared several models to do reject inference in corporate loans. They found that tree-based methods were more stable and had the transparency needed to ensure ethical concerns, such as discrimination against minorities. It is important to note that it differs from this thesis as it uses only one model, not pseudo-labeling, and uses corporate loans. As such, we will use a tree-based method for one of the first models to create the pseudo labels.

Additionally, most methods expect the data to be normally distributed. However, tree-based methods do not make such assumptions about the data distribution. Since the rejected part of the population is missing, we will not assume it to be normally distributed. In addition, the lack of this assumption could lead them to perform better when missing not at random compared to logistic regression, which would have biased estimators. Moreover, another reason I chose to use a decision tree-based model is that it provides better results when compared to linear regression for cases where the relationships are complex and non-linear (Gareth James (2013)), which is what we expect to find.

We will be using light gradient-boosting machine. This technique is more efficient, trains faster, uses less memory, is more accurate than other techniques, and can handle large amounts of data. This technique is also supposed to be faster than eXtreme Gradient Boosting, which made it more appealing for us to use in this thesis. The reason why it is faster is because it uses leaf-wise, vertical growth rather than level-wise growth. The trade-off, however, is that it could be less robust and overfit. As the computational power was a limitation, we chose to use LightGBM over xGBoost at the expense of potentially overfitting the training data.

The light gradient-boosting machine is a gradient-boosting decision tree with gradient-based one-sided sampling (Ke et al. (2017)). Gradient boosting decision tree is a model that ensembles decision trees, meaning we train several decision trees sequentially using the residuals of the previous tree. Gradient-based one-sided sampling is a technique where we select only a part of the data for training. It works by filtering the data using a loss function; the instances with more significant gradients are all used, while we only keep some instances with smaller gradients. We base this decision on a random selection for a fixed number of instances.

2.5 Other first models

It is possible to do pseudo-labeling by assigning all that are missing to a category instead of using a model. Two possible decisions can make sense in labeling data: Label all missing data as 0, meaning they did not default. When most loan owners do not default, the proportion of not defaulting is exceptionally high. It could make sense to label all the rejected as not defaulting. Although this solution is simple, it is easy to implement.

It is also possible to set all to default. It is important to remember that the data were not missing at random but because a previous model decided that it had a high score and was, therefore, likely to default. Labeling all as default would be better in cases where reject inference is impossible, approval rates and default costs are high, or random decision-making cases. We can use logistic regression after setting everything to default in these cases. These scenarios are why testing different rejection rates and random decision-making is essential.

As we plan on doing pseudo-labeling for reject inference, we are interested in learning if and to what extent a lightGBM could outperform these two decisions. Based on our own short experience, we have noticed that in some cases, business decisions, human logic, and empiricism outperform complex algorithms; as such, we will be further investigating this in this thesis.

2.6 Logistic Regression:

In order to do pseudo-labeling, we need two models. One is to create the pseudo labels for the unlabelled part of the population, and the second is to train on the complete data set. Given that the goal of this thesis is to mainly study the first step, which is setting all to default / not default or teaching a first tree-based model, our goal was to have a second model that did not necessarily need to be the most complex, just one that would perform well in predicting a binary outcome with tabular data. Considering that using linear regression is not optimal for a binary problem as probabilities could be negative or superior to one, we considered other models.

We use logistic regression when the goal is to predict a categorical variable, usually binary, like in this case. It predicts the probability that a certain observation will belong to a certain category (Gareth James (2013)). Other variables, also known as predictors, are then used to attempt to forecast the outcome, the target (Nick and Campbell (2007)). Since this case has multiple predictors, we will use multiple binary logistic regression.

$$\text{P=(Target} = 1) = \frac{\exp(\beta_0 + \beta_1 \text{X1} + ...\beta_n \text{Xn})}{1 + \exp(\beta_0 + \beta_1 \text{X1} + ...\beta_n \text{Xn})} \tag{1}$$

In logistic regression, the algorithm chooses the estimators to maximize the likelihood function, meaning the objective is to have an estimated probability of the outcome occurring as close as possible to the actual outcome (Gareth James (2013)). In other words, if one loan applicant's actual outcome is the default, the expected probability of default should be as close to one as achievable.

The description of logistic regression shown above supports our ruling of using it as a second model.

2.7 Other Applications

As mentioned, most previous models were applied to image classification problems, where they can take advantage of the high amount of unlabelled data. However, semi-supervised learning and other approaches can be used in other cases with many unlabelled data, such as healthcare and biomedicine (Kim et al. (2023)). We can apply the models used in this thesis to other types of loans; here, we focus only on home loans, but corporate and personal loans also suffer from the same issues.

2.8 Conclusions

As a result of evaluating different research on reject inference and semi-supervised learning, pseudo-labeling is the most interesting, as it is possible to turn the unlabeled data into labeled data using the first model and then train a second model, logistic regression afterward on the entire training data. Pseudo-labelling allows us to use the full data set as all data will be now labeled and selection bias will no longer be a problem turning our case into a simple binary classification problem.

We want to test three options: 1) setting all to not default. Although it is a simple option, it is practical and could yield good results. 2) Setting all to default is another simple option. However, it would be interesting to see whether there is a difference in performance between the two: setting all not to default or default. 3) The more complex option is to train a supervised learning model, lightGBM, on the already labeled data to make predictions on the missing data and store these labels as pseudo labels to be used along with the labeled data to train the second model. All the models would then have a second model applied, logistic regression, the final one that could make predictions for unseen data.

Additionally, we wish to better understand what impacts and what drives the performance of pseudo-labelling approaches. For that purpose, we will test also the impact of having the rejected be missing at random instead of being the result of a decision done on top of classification model. We will also investigate which n value for percentage we should use when selecting n % of observations with the highest scores to be rejected by the final model. The need to compare the two scenarios arises because 1) when randomly allocating observations to the two groups, the distribution of those sub-groups is expected to remain similar, while in the missing not at-random case, we expect that they differ, and 2) in the label noise section the corruption of labels had a different impact depending on whether they had added the noise randomly or not. As such, we expect the performance to differ in the two cases. In addition, the rejection rate is a decision banks need to make, and as such, we will explore several to help guide this decision.

3 Descriptive Statistics

The data set came from a Kaggle source containing information regarding the characteristics of applicants for home loans. The data set includes the following information: characteristics regarding the loan, the home, the neighborhood and city, and other credits of the loan applicant, among others. It consists of tabular data with both numerical and categorical variables. As this thesis aims to investigate pseudo-labeling when retraining, we altered the data

set by applying a prior model. As a result, we used part of the data set to train and validate this prior model, and the other part resulted from its decisions. It is important to note that we cannot access this prior model, and this thesis will focus only on retraining. We apply all the models after this prior model. Despite not having a time variable, the variable containing the IDs is a proxy for this.

Two variables in the data set are most important to understand. Target that represents whether the borrower would default on their loan or not. The target variable is what we are aiming to predict. In addition, the variable rejected is also essential as it represents whether the observations were rejected by the bank, accepted, or used to train and validate the prior model.

$$Target = \begin{cases} 0, & \text{if does not default, fully repays loan} \\ 1, & \text{if does default, does not repay loan} \end{cases} \tag{2}$$

$$Rejected = \begin{cases} 0, & \text{if accepted} \\ 1, & \text{if rejected} \\ -1, & \text{if used to train or validate the prior model} \end{cases} \tag{3}$$

The data set consists of 52.699 observations used for the training of the prior model, 26.349 for its validation, and 184.443 observations as a result of the decisions of the first model. Of the 184.443 observations, the model rejected 4451, while 179.992 were accepted, resulting in a rejection rate of approximately 2 % and an imbalance in this variable. The 2 % indicates that the rejection rate is low, and according to the literature review, when we use lightGBM to predict the missing labels, it is not expected to yield better results than setting all to default/not default.

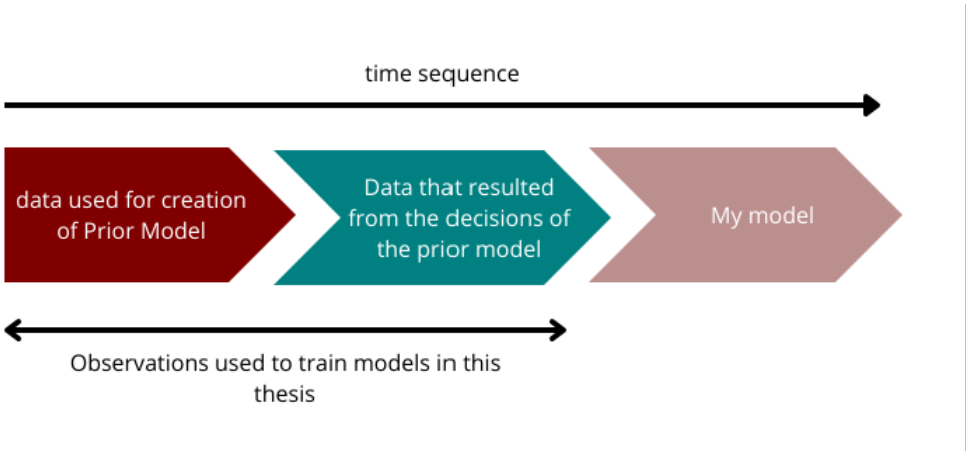


Figure 1: Time Diagram

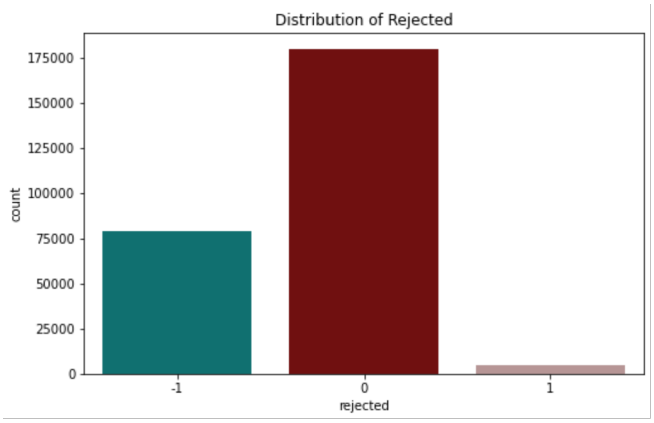


Figure 2: The Distribution of the Rejected Variable

The data set has 263.491 applicants in total and 124 columns. Of all 263.491 applicants, only 20368 defaulted on their loans, representing around 8 percent. The low ratio indicates that the target variable is unbalanced.

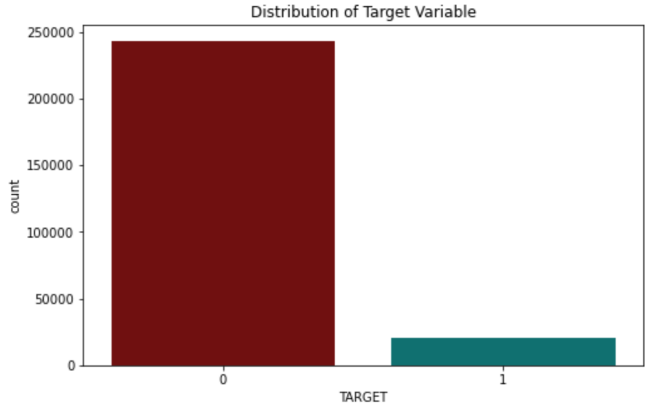


Figure 3: The Distribution of the Target Variable

Additionally, given that according to this literature review, the distribution is expected to differ between the groups, rejected and accepted, we computed the percentage of observations that default in each group. In the rejected group, 39 % of observations default, whereas in the accepted group, 7 % of observations default. As expected, the percentages differ between groups.

Moreover, those who default have an average income of 170.189 a year and ask on average for 566.007 in the total amount of credit, resulting in a credit/income ratio of 3.3, while those who do not default have a slightly higher average, 171.503, ask for 609.786 in credit amount resulting in a credit/income ratio of 3.6. The average number of days employed is higher for those who do not default, 64935 versus 41929. Lastly, the average age of those who do not default is 44 versus 41 for those who default. Thus, those who default have a slightly lower

income, ask for less credit, have a smaller credit/income ratio, have been employed for less time, and are younger.

4 Methodology

In this thesis, we will use pseudo-labeling for reject inference. As explained, pseudo-labeling works by estimating the pseudo-labels for the rejected group, which we then join with the labels of the accepted part and finally train a model with all the data. We will explore pseudo-labeling by having three different models/decisions (LightGBM, setting all to default, and setting all to not default) to create pseudo-labels, which we will then use to train a logistic regression and compare the final performance of the models. The main goal is to investigate if lightGBM can outperform setting all rejected to a given category. We will also conduct an experiment by creating a missing at-random scenario to evaluate if the decision being the result of random allocation is a factor that influences the performance. As such, we created six models as seen in figure 4: lightGBM missing not at random, lightGBM missing at random, logistic missing not at random setting all to default, logistic missing not at random setting all to not default, logistic missing at random setting all to default and logistic missing at random setting all to not default. Furthermore, we will experiment with five rejection rates to guide the banks.

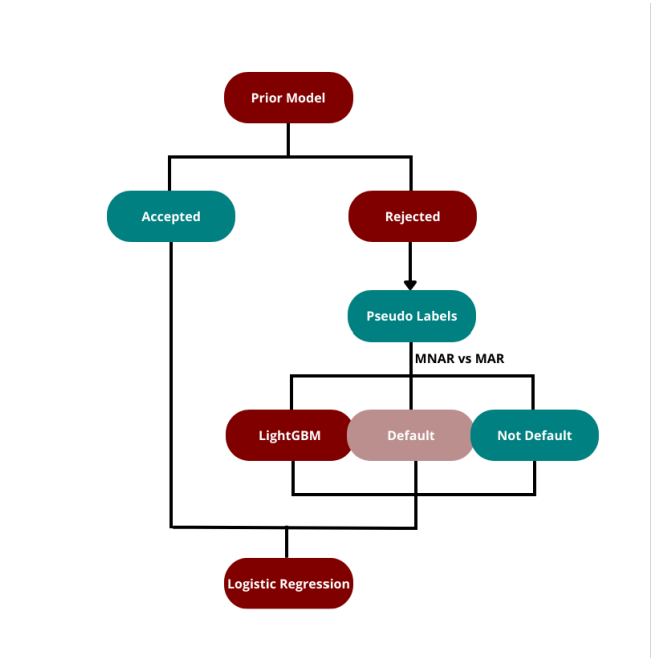


Figure 4: Diagram of Models

It is important to note that the missing not-at-random case is represented by the actual rejected/accepted distribution in our data set, as it was the result of the prior model that made

decisions not at random. For the missing at the random case, we will have to create this scenario by randomly allocating observations to rejected in the same quantity as they appear in the actual case, two % of the sample as the expected proportion of the population is the proportion of the sample if no evidence points otherwise.

4.1 Data Treatment:

For all the models, we used the same data treatment.

- Missing Values: We filled in all the missing values. We used the median to fill the numerical variables as we were not assuming the data would be normally distributed. The median could better represent the distribution over the mean. For categorical variables, a string, 'missing,' was used.
- Perfect Multicollinearity: All the variables that correlated higher than 0.95 with other variables were removed from the training set to prevent this issue.
- Standardized: All variables used to train the models were standardized.
- Categorical variables: We created Dummy variables for all the categories in the categorical variables. Additionally, one variable, organization type, had over ten different categories. We used a function to reduce the number of categories to prevent issues.
- Variables removed from the training set: We needed to remove some variables from the training set: id, the target variable I am trying to predict, the variable that indicates whether the model rejected that loan applicant or not, the variable for random rejection in the missing at random cases I created, the fake targets and a population variable that indicated whether that observation we used for the training, validation or a result of the prior model.

4.2 Experimental Setup

4.2.1 LightGBM:

We chose LightGBM based on the research; not only does it ensure transparency, which we need to relax ethical concerns, but it is also a tree-based method, which, based on the literature review, we expect to perform well in this case.

For the two models we will create using lightGBM, it is essential to note that they will result from lightGBM and logistic regression, which will still be applied afterward. We will start by training a lightGBM using both the accepted part of the sample and the data used to

train and validate the prior model. The reason behind the decision to use the data used to train and validate the prior model was that we had access to it, and it was computationally possible, so we chose to take advantage of as many observations as possible.

Then, we fine-tune lightGBM with grid search and undersampling. In the grid search, we tested to find the optimal values for the parameters: number of estimators, learning rate, max depth, and number of leaves to minimize the mean squared error. Additionally, undersampling is a technique that keeps all the instances belonging to the category with fewer observations, in our case, loan defaults, and randomly selects instances from the category, not default, with the most cases. Undersampling decreased the imbalance in the categories, and it helped lightGBM, in this case, make better decisions.

Afterward, the model is used to make predictions on the unlabelled data, the rejected part of the population. We make these predictions using the predict function. We store these predictions on whether the rejected loan applicant would default as pseudo labels and create a new variable: fake labels. This variable contained the actual labels for the accepted data, data used for the previous model, and the pseudo labels. Then, we used this variable to train the second model, a logistic regression.

4.2.2 Setting all to a Category:

For the other models whose pseudo-labels were not a result of lightGBM, we had to assign the rejected into one of two categories: 1) set all to default or 2) set all to not default. We did both scenarios as we will investigate further if there is an impact on the performance depending on whether we assign all to default or not default.

We created the pseudo labels by allocating all rejected to a category and then joining it with the actual labels for the rest of the data. As in the lightGBM case, we will also use the pseudo labels to train a logistic regression.

4.2.3 Missing at Random

The missing not-at-random case results from the prior model decisions, which we found in the data set; in the missing not-at-random case, we know that the distributions between those rejected and accepted differ; however, in a missing-at-random case, the distributions are expected to remain similar. As seen in the literature review section, noisy labels, whether the corruption of the labels is the result of randomness or not, impact performance. This impact created a need to investigate this scenario. As such, we had to create a missing at-random sce-

nario.

We created the missing at-random scenario by randomly allocating n % of the observations in the data set to the rejected category. Since the prior model rejects 2 % of the loan applicants and the expected proportion of the population is the proportion of the sample if no evidence points otherwise, we will allocate 2 % of observations to the rejected category. For repeatability purposes, we used a seed.

We will study the impact of random decisions for the three models: lightGBM, logistic set all to default, and logistic set all to not default.

4.2.4 Logistic Regression:

After these first steps, meaning we train a lightGBM to make predictions for the rejected group or allocate the rejected to either default or not default, the pseudo labels are joined with the actual labels for the accepted part of the sample to train a logistic regression model. Afterward, we evaluated the logistic regression models against the actual labels.

4.2.5 Rejection Rates:

We used percentages to help guide banks' decisions and make the models comparable by selecting the same number of observations to reject versus accept. Using rejection rates instead of a cut-off threshold was a necessary decision. If we used the basic score rule over 0.5, all the models would have different score predictions, likely resulting in significant differences in the number of observations each model would set to default/non-default. In essence, we selected the top percentage of observations with the highest scores for default prediction. The remaining data were predicted not to default. The percentages used were as follows: 1, 5, 10, 20, and 25 %.

4.3 Model Evalutation

4.3.1 Distribution and Metrics:

The final data on which the hypothesis will be formulated results from the six models. A distribution is needed to make this possible, creating several results for each model. As such, although K-folds are generally used for hyperparameter tuning, we used them to create a distribution of the results. We used five-folds. The five-folds means that each model has five outputs for each rejection rate, resulting in twenty-five outputs per model. The output consists of the

steps	lightGBM MAR	lightGBM MNAR	log MNAR 1	log MNAR 0	log MAR 1	log MAR 0
decision-making	random allocation of 2% of dataset into rejected	rejected as a result of the prior's model decision	rejected as a result of the prior's model decision	rejected as a result of the prior's model decision	random allocation of 2% of dataset into rejected	random allocation of 2% of dataset into rejected
pseudo labels for the rejected	lightGBM trained with accepted and prior model's data in order to predict the pseudo label of the rejected observations	lightGBM trained with accepted and prior model's data in order to predict the pseudo label of the rejected observations	all rejected allocated to the defaults category	all rejected allocated to the not defaults category	all rejected allocated to the defaults category	all rejected allocated to the not defaults category
final model	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)	logistic regression trained with full data set (pseudo labels for the rejected and real labels for the accepted)
evaluation	model evaluated against real labels	model evaluated against real labels	model evaluated against real labels	model evaluated against real labels	model evaluated against real labels	model evaluated against real labels

Figure 5: Steps to create the models

Note: The colors are used to identify the different options in the steps. Two options are given for the decision-making: red (random allocation) and black (not random allocation). For pseudo-labels, there are three options given by the colors: black (set to 0), red (lightGBM), and green (set to 1). While for the final model and evaluation, only one option is given by the color black.

following metrics: accuracy, F1 score, recall, precision, lift score, ROC curve, and revenues, which we will explain in the next paragraph. In addition, there was also a graph showing the scores distribution.

The goal of a bank is not to maximize lift, F1 score, or accuracy. Like most corporations, the bank's main objective is to maximize profit. The necessity to assess performance using profit resulted in a need to create business Key Performance Indicators to show the impact on profit. It is important to note that banks maximize profit by maximizing the amount of credit that the bank grants as loans that are re-paid and minimizing the losses from credits that are not re-paid. We created the following variables to conduct these analyses:

1- Revenue Gained: This variable results from the sum of the amount of credit variable for all those the model predicted would not default and did not default. The reason why we chose to use the amount of credit and not compute the actual value that the bank will earn in interest was that 1) We do not have access to that data; 2)the interests applied to each loan depend on the bank, the economic state, characteristics of the loan applicants and many other factors that are not the focus of this thesis. Even so, the amount of credit given is a proxy for profit;

2- Revenue Lost: This results from the sum of the amount of credit variable for all those that the model predicted would not default but do, in fact, default. Revenue lost represents the bank's loss;

3- Revenue Potential: This consists of the sum of the amount of credit variable for all those the model predicted would default but did not default. This variable allows us to investigate how much better each model could potentially be in terms of the amount of credit;

4- Profit: This variable results from Revenue Gained minus Revenue Lost. We created this variable to obtain a better way to evaluate the best rejection rate. The need for it came as the rejection rates increased, and the bank granted fewer loans when compared to higher rejection rates. Therefore, we expect that the Revenue gained will decrease. However, we expect a decrease in revenue loss as the model would hopefully make fewer mistakes. As such, there was a need to create this variable to allow us to investigate whether the decrease in Revenue lost would be higher than the decrease in Revenue gained.

Our target variable, the variable that indicates whether the loan applicant defaults or not, is highly imbalanced. Only 8 % of observations default. As such, we will not use the metric accuracy to evaluate the final models' performance.

4.4 Approach Evaluation

4.4.1 Hypothesis tests:

We used Kruskall Wallis H test. It is a non-parametric test that uses ranks instead of means, sometimes known as a one-way ANOVA on ranks. The reason why we opted for a non-parametric test is that there are not enough observations per rejection rate and model to ensure the Central Limits Theorem.

As the quantity of hypothesis tests is too high to show the results of all the tests, we will only acknowledge when there is a significant difference or if none of the metrics resulted in rejecting the null hypothesis.

5 Results

The main goal is to investigate the impact of using lightGBM versus setting all rejected to a given category. To make this analysis, we used six metrics: F1 score, recall, precision, lift, revenue potential, and profit. Additionally, we used the Kruskal-Wallis test. We used the 0.05 p-value as a reference and will only present the tests where the null hypothesis was rejected.

We will start by assessing the quality of the pseudo-labels and then we will see different rejection rates, not to improve the model's performance but to help guide the banks' future decision-making. Next we will go into the main question of this thesis: lightGBM will outperform setting all rejected to a category. Then, we will check if setting all to default versus not default will impact the final model's performance. Afterward, we will look at a factor that can condition the performance, missing not at random. Lastly, we will show the score distribution and ROC curves.

We created the tables shown in this section by computing the mean and median values in our code and then inserted those values into the tables.

5.1 Quality of Pseudo Labels:

We will now analyze the pseudo-labels for the 4451 rejected observations. For the missing at-random scenario, there will be a rounding error as we used 2 % of all the data set.

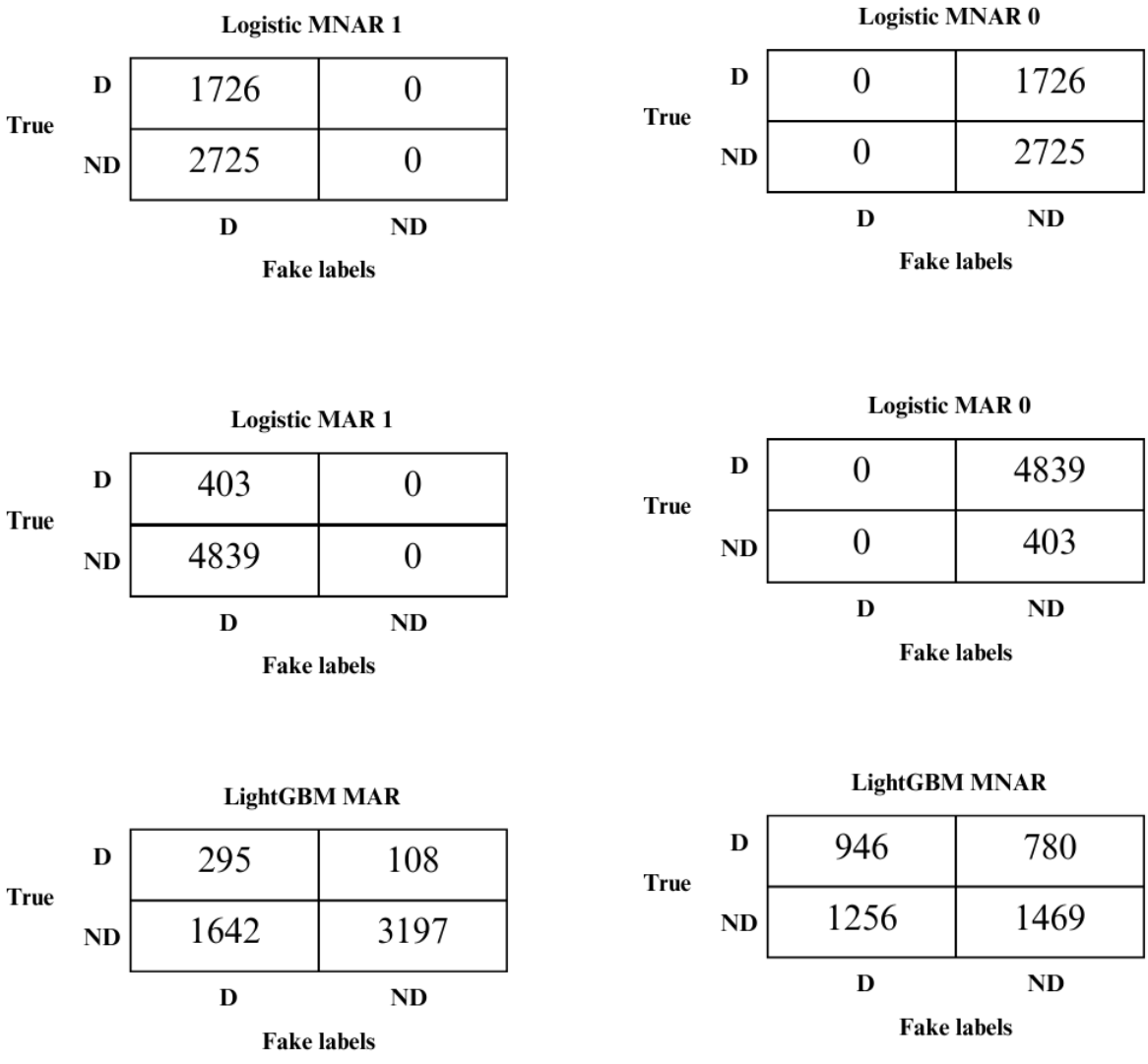


Figure 6: quality of pseudo-labels

Note: These represent the pseudo labels against the real labels. NOT the final model's performance. D stands for defaults, and ND for not defaults.

The figure above shows the confusion matrix for the three options for creating pseudo-labels. These confusion matrices do **not** result from the final model's decisions. Instead, they are comparing the pseudo-labels created by lightGBM, setting all to default and setting all to not default against the actual labels. We will then train a logistic regression based on these pseudo-labels.

5.2 Rejection Rates Analysis

We opted not to use the 0.5 cut-off threshold. Instead, we are selecting the top n % of observations with the highest scores we predict to default, as it is important to study if there is a

rejection rate that would lead the models to have better performance. To evaluate the different possibilities for n=[1, 5, 10, 20, 25], we used three metrics: F1 score, lift, and profit. We created the variable profit to be a business KPI and to investigate whether the decrease in revenue lost would compensate for the decrease in revenue gained as the percentage of observations rejected would rise. We used the F1 score and lift to assess the performance using non-business KPIs.

We used medians as an alternative to means because we are not assuming the data to be normally distributed. Hence, the means might not represent the distribution of the variable. Additionally, we used five percentages (1, 5, 10, 20, and 25 %), and each will have five observations for each of the six models. Resulting in 25 observations per model.

%	LGBM MNAR	LGBM MAR	Log MNAR 1	Log MNAR 0	Log MAR 1	Log MAR 0
1	0.1	0.06	0.1	0.09	0.1	0.09
5	0.22	0.23	0.24	0.24	0.24	0.24
10	0.26	0.28	0.28	0.28	0.28	0.28
20	0.26	0.28	0.28	0.28	0.28	0.28
25	0.26	0.27	0.27	0.27	0.27	0.27

Table 1: median F1 score

%	LGBM MNAR	LGBM MAR	Log MNAR 1	Log MNAR 0	Log MAR 1	Log MAR 0
1	5.54	2.90	5.42	5.38	5.40	5.38
5	3.71	3.73	3.96	3.93	3.95	3.94
10	3.04	3.20	3.23	3.22	3.25	3.24
20	2.40	2.52	2.53	2.53	2.53	2.53
25	2.19	2.30	2.30	2.31	2.31	2.31

Table 2: median lift

%	LGBM MNAR	LGBM MAR	Log MNAR 1	Log MNAR 0	Log MAR 1	Log MAR 0
1	44496.0	44174.0	44513.0	44523.0	44519.0	44517.0
5	43438.0	43486.0	43564.0	43552.0	43556.0	43560.0
10	41772.0	41922.0	41948.0	41947.0	41964.0	41949.0
20	37908.0	38131.0	38135.0	38147.0	38119.0	38135.0
25	35853.0	36082.0	36076.0	36090.0	36092.0	36088.0

Table 3: median profit

By running a Kruskal Wallis H test, we found significant differences in all six models for all three variables. However, this test did not tell us which had higher KPIs, only that there was a significant difference. By analyzing the medians, we encountered the following: 1) Table 1 shows that the F1 score is highest in the 10 and 20 % and then lowers for the 25 % and is always lowest in the one %. 2) Table 3 evidences that the profit decreases with the percentage increase, consistently highest in the one %. We expected this decrease as we reject more applicants and will grant fewer loans and less credit. However, we were hoping to see if the decrease in revenue lost would make up for that fact. It did not. 3) In Table 2, it is observable that lift decreases with the percentage increase. Being highest in the one % case. Except for the LightGBM missing at random case, it is highest in the five % case.

5.3 LightGBM vs. Setting all to a Category:

The means and medians in the following tables were computed using all 25 observations(5-folds and 5 rejection rates). We used the results of all five rejection rates because we wish a model to outperform another regardless of the rejection rate.

We ran hypothesis tests for the six metrics: F1 score, lift, recall, precision, profit, and revenue potential.

- LightGBM MNAR vs. Logistic MNAR 1:

There is only a significant difference in terms of F1 score. By analyzing Table 4, we can see that the median F1 score is higher in the case of logistic MNAR 1.

model	F1 mean	F1 median
LGBM MNAR	0.22	0.26
Log MNAR 1	0.23	0.27

Table 4: f1 score comparison between the two models

- LightGBM MNAR vs. Logistic MNAR 0:

There is only a significant difference in terms of F1 score, with it being highest in the case of setting all rejected to not default based on table 5.

model	F1 mean	F1 median
LGBM MNAR	0.22	0.26
Log MNAR 0	0.23	0.27

Table 5: f1 score comparison between the two models

5.4 Impact of Setting to the two different categories:

We wanted to see if there was a difference in performance depending on the choice of setting all to default vs. not default.

- Logistic MNAR 1 vs. Logistic MNAR 0:

No significant differences exist in any of the metrics, as all tests returned 'fail to reject the null hypothesis.'

- Logistic MAR 1 vs. Logistic MAR 0:

There are no significant differences in any of the metrics.

5.5 MNAR vs. MAR:

Another goal was to see if there would be an impact on the performance of the models depending on whether the decision to reject had been made at random or not. We ran hypothesis tests with the same six metrics used above.

- LightGBM MNAR vs. LightGBM MAR:

There is only a significant difference in terms of F1 score. By analyzing the medians in Table 6, we can see that the F1 score is higher in the random missing case.

model	F1 mean	F1 median
LGBM MNAR	0.22	0.26
LightGBM MAR	0.24	0.27

Table 6: f1 score comparison between the two models

- Logistic MNAR 1 vs. Logistic MAR 1:

There are no significant differences in any of the metrics.

- Logistic MNAR 0 vs. Logistic MAR 0:

There are no significant differences in any of the metrics.

5.6 Analysis of Revenues:

Despite not having found significant differences in the hypothesis tests regarding the variable profit and revenue potential, since the primary goal of a bank is to maximize this variable and not to achieve significant results, we will still explore the potential profit and revenue variables for the six models. Revenue potential represents those the model predicted would default, but the actual outcome is not. Consequently, the bank could make more money, but not because of the model's decisions. The revenue potential represents a variable we wish to minimize.

model	mean profit	mean revenue potential
Log MNAR 1	40824.28	5077.56
Log MNAR 0	40816.76	5081.32
Log MAR 1	40944.36	5077.52
Log MAR 0	40823.16	5078.12
LGBM MNAR	40689.56	5144.92
LGBM MAR	39155.8	5111.8

Table 7: Mean Profit and Revenue Potential for all models

Table 7 shows that the highest profit mean and lowest revenue potential come from the same model, logistic missing at random setting all to default.

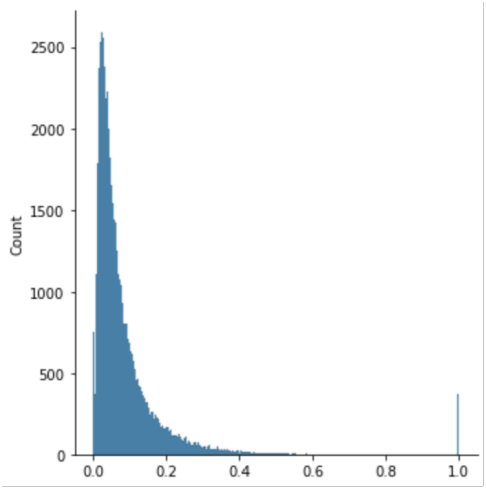
model	median profit	median revenue potential
Log MNAR 1	41948.0	3953.0
Log MNAR 0	41947.0	3955.0
Log MAR 1	41964.0	3940.0
Log MAR 0	41949.0	3952.0
LGBM MNAR	41772.0	4035.0
LGBM MAR	41825.0	3969.0

Table 8: Median Profit and Revenue Potential for all models

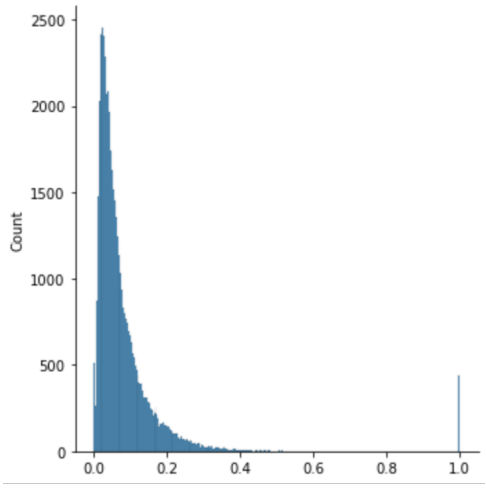
Additionally, Table 8, which differs from Table 7 by having the medians instead of the means, shows the same conclusion as Table 7. The model with the highest median profit and lowest median revenue potential is still logistic missing at random, setting all to default. For the missing not-at-random case, which is the scenario the bank will find, the highest median profit and lowest revenue potential come from the model logistic missing not-at-random, setting all to default.

5.7 Analysis of Scores Distribution

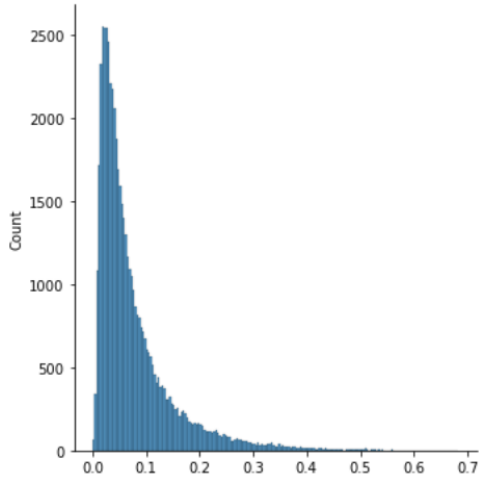
In the cases where the first model is lightGBM, the predicted scores of those rejected by the model are close to one, which is not the case in the models where the first model is a decision.



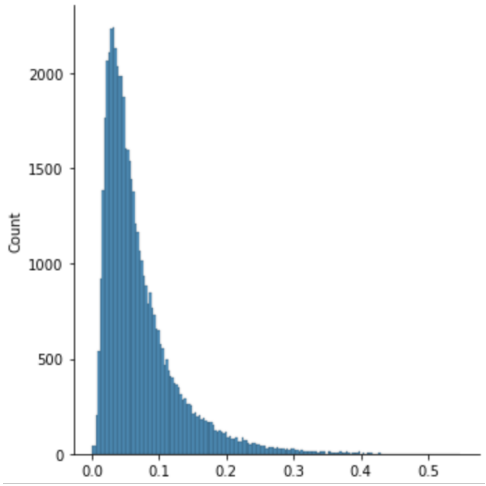
(a) The Distribution of scores in LGBM MAR



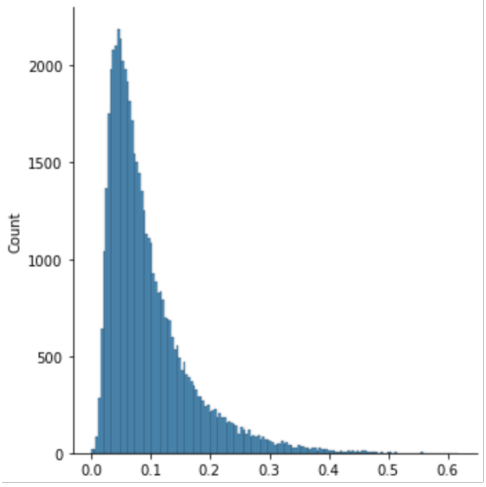
(b) The Distribution of scores in LGBM MNAR



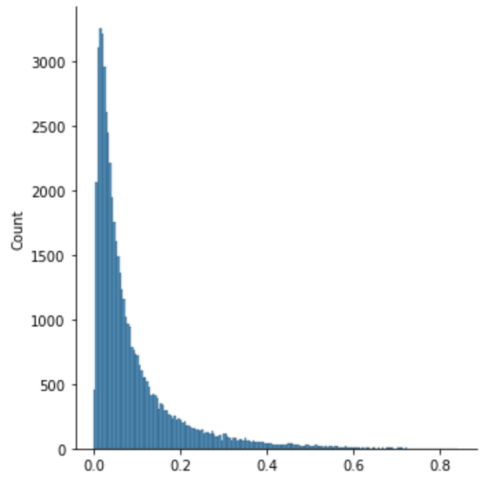
(c) The Distribution of scores in Log MAR 0



(d) The Distribution of scores in Log MNAR 0



(e) The Distribution of Scores in Log MAR 1



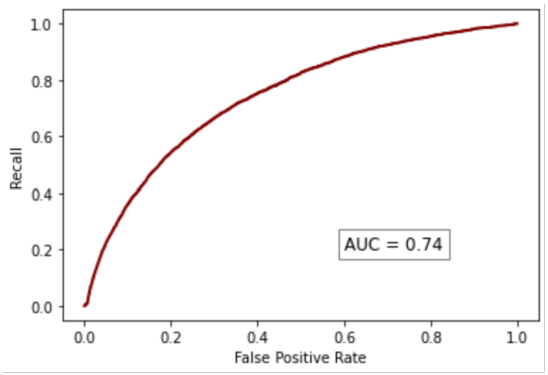
(f) The Distribution of scores in Log MNAR 1

Note: These represent the distributions of the scores of the final models.

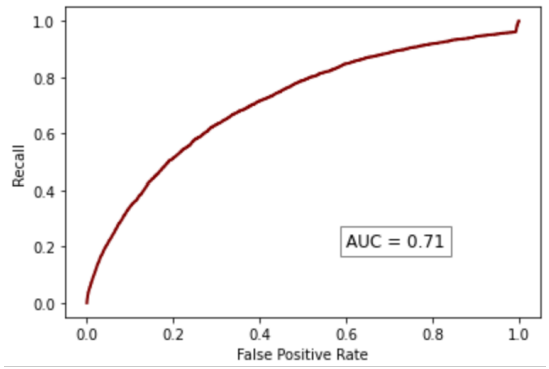
Figure 7: Scores distribution

5.8 Analysis of Roc Curves:

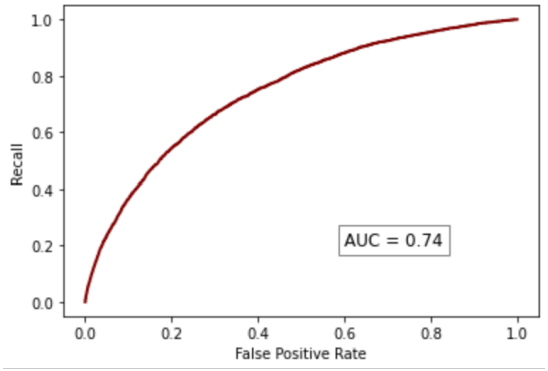
The ROC curve is similar and has the same AUC value for all the models that result from setting all to a specific category and then applying logistic regression. In LightGBM, in the missing not-at-random case, the AUC is slightly lower at 0.71 vs. the 0.74 of all other models.



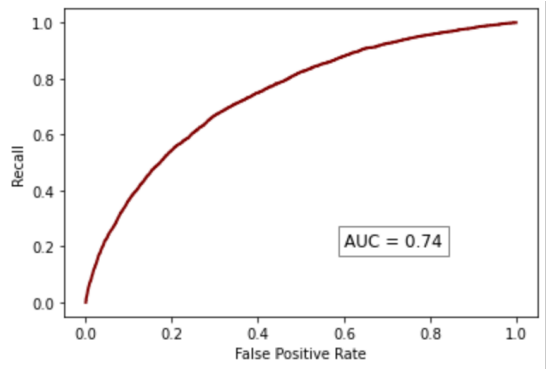
(a) ROC curve LGBM MAR



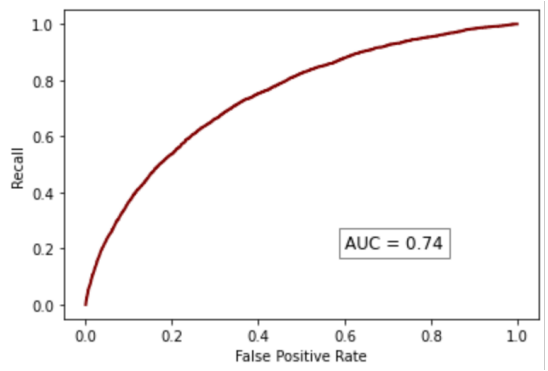
(b) ROC curve LGBM MNAR



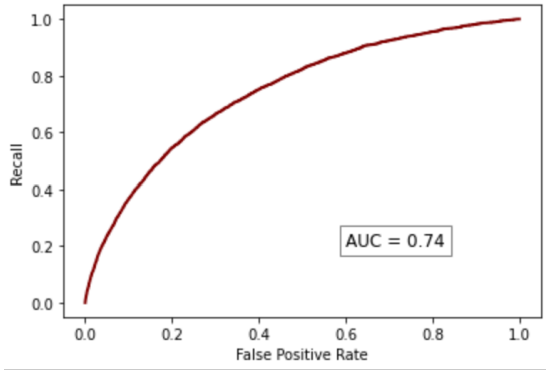
(c) ROC curve Log MAR 0



(d) ROC curve Log MNAR 0



(e) ROC curve Log MAR 1



(f) ROC curve Log MNAR 1

Note: These are the ROC curves for the performance of the six final models. Notice how AUC is equal for all except lightGBM MNAR.

Figure 8: ROC curves of Models

6 Discussion

In the results section, we showed how, despite having significant differences in all three key performance indicators, lift, profit, and F1 score, no rejection rate leads to better results than the others. There is a trade-off between lift and profit versus F1 score. As for the F1 score increases with higher rejection rates, with 20 % resulting in the highest F1 score results. While for-profit and lift it is higher in the case of 1 %. It makes sense for lift to decrease with the increase in rejection rate as it is sensitive to false positives, while the F1 score also takes into consideration the false negatives and, as such, is not as sensitive. Since profit is the bank's primary goal, it will yield better results to choose a rejection rate of 1 %, meaning a lower rejection rate. However, banks also have monetary constraints, which we did not consider as they will depend on the bank and the economic state.

Our main goal was to investigate whether it was worth doing a model like lightGBM instead of simply setting all rejected to a category. There is a significant difference in the final models' performance in terms of F1 score, with it being highest in the case of setting all to default and not default. As such, we did not add value to the performance by estimating the label for those rejected. Furthermore, it is essential to note that training a model would need resources, such as a data scientist team, which would have labor costs and computing resources. If it has a lower F1 score and costs more than setting all to default/not default, then creating a model could not be worth it.

Despite not having significant results in the business key performance indicators, we still wanted to investigate which model resulted in the highest profits and lowest revenue potential. This model was logistic regression, setting all to default in the missing not-at-random. The results are coherent with previous conclusions that indicated that simply setting all the rejected applicants into the default or not default outperforms using an algorithm to predict the pseudo labels.

By analyzing the distribution of the scores, we can see a significant gap in lightGBM between those it finds very likely to default and the rest of the observations. Unlike in the setting all to a given category where there is no gap in the distribution of scores.

The ROC curves are also similar and have the same value for AUC, 0.74. Except for lightGBM MNAR, where the AUC is 0.71. Together with a worse F1 score, it indicates that lightGBM makes more inaccurate predictions than the other models.

Furthermore, there was no significant difference between setting all the missing labels as default vs. not default. The lack of significant results was unexpected. However, it can be

justified by the fact that the rejected only accounts for around 2 % of the entire data set, and therefore, it might not be sufficiently big to result in significant differences in performance. Additionally, in section 5.1. we can see that there is an higher number of true positives, 1726, for the setting all to default in the MNAR case and therefore will have an higher recall. Together with the fact that it has a higher profit, it is better to assign all to default.

Next, we wish to compare the case of the missing labels due to random decision-making to investigate if it affects the models' performance. We found no significant differences for the models whose rejected we allocated to a given category. However, for the lightGBM model, there was a significant difference between having the labels be missing at random or not at random for the F1 score, with the mean and median being highest in the case of missing at random. This result was unexpected, as we expected that lightGBM would not suffer in performance between the two cases as it does not make assumptions regarding the data distribution. However, lightGBM learns from the accepted data to make predictions for the rejected data, and since the data is missing at random, the two will be similar. In contrast, in the missing not at random, the distributions differ. It is easier for lightGBM to generalize and find more patterns in the first case.

7 Limitations

One of the most significant limitations we encountered during this thesis was that we only had access to one data set. More data sets would have ensured higher reliability. Another limitation was how the data set came from a Kaggle source. Using a data set from a real-world scenario would have been optimal, but it would have been challenging to ensure data privacy, and there was the potential risk of not receiving the data set promptly.

In addition, there were constraints and situations that banks would face that we did not consider while doing this thesis. There are requirements that banks need to comply with that depend on the country and legislation, and to ensure more generalized results, we opted not to consider them. These requirements include the effort rate, minimum age, and maximum loan-to-value. Very recently, we saw how these requirements change with the economic state as the effort stress test was updated in 2023.

Additionally, another limitation is that there is a missing variable. Some loan applicants might get accepted and decide not to proceed. They might have gotten a better offer from a different bank. However, that did not happen in our data set, and that decision is different from the rejected/accepted decision. The bank decides who to accept or reject, while the applicant decides if they should still proceed with that loan. Loan applicants do not make this decision at

random. It is made based on how competitive a bank is, among other factors; for example, the applicant could decide to wait to buy the house for personal or economic reasons.

Although this thesis can be used for personal and corporate loans, we expect the results to be different as we also expect the characteristics of applicants for those types of loans to differ from those of applicants for home loans. For example, a home loan applicant's characteristics differ from those of a corporate loan applicant.

Lastly, we used the amount of credit as a proxy for the bank's profit. However, a bank's profits are calculated by summing the interests earned on a given loan.

8 Conclusions

Since the primary goal of this thesis was to investigate the impact of having lightGBM predict the missing labels compared to the baseline, setting all to a category (default or not default), we explored this topic by creating three models (lightGBM + logistic, setting all to default + logistic, setting all to not default + logistic). Additionally, we created an experiment to see if the missing labels resulting from random decision-making had an impact and tested five rejection rates to help guide the bank's decisions.

Our results showed that setting all rejected to a given category outperforms lightGBM regarding the F1 score. By analyzing the business KPIs, we found that setting all to default has a higher profit and lower revenue potential (revenue the bank could be making if not for incorrect labeling by the model of not defaults as defaults). In addition, it has a higher recall in the rejected population than assigning all to not default. Given the higher profit and lower revenue potential, we recommend that the banks set all rejected to the category default.

Furthermore, the rejection resulting from random decision-making only significantly impacted lightGBM, which is related to the model's ability to make better generalizations in the missing at-random as the rejected will be similar to the accepted population.

Moreover, we also investigated five rejection rates to help guide banks' decisions. We found that as the rejection rate increased, the F1 score increased while lift and profit decreased. As such, if the bank's ultimate goal is to generate profits, we recommend using a lower rejection rate, such as one % of the highest predicted scores.

We believe future work is needed in the following areas:

1. As one of the limitations was using a Kaggle data set, we suggest applying the same methodology to a real-world data set.

2. Additionally, in this thesis for lightGBM, we used the predict function to create the final predictions of the model. However, using a different cut-off threshold could yield better results. An experiment to check the improvement of lightGBM using different thresholds is needed.

3. Furthermore, an experiment comparing setting all rejected to the scenario where all data is labeled, meaning 100 % approval rates, is also needed.

This thesis contributes to the field by showing that banks should assign all rejected to default when attempting to do reject inference.

References

Addo, P. M., Guegan, D., and Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2020). Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Banasik, J. and Crook, J. (2007). Reject inference, augmentation, and sample selection. *European Journal of Operational Research*, 183(3):1582–1594.

Cascante-Bonilla, P., Tan, F., Qi, Y., and Ordonez, V. (2021). Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6912–6920.

Crook, J. and Banasik, J. (2004). Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance*, 28(4):857–874.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385.

Ehrhardt, A., Biernacki, C., Vandewalle, V., Heinrich, P., and Beben, S. (2021). Reject inference methods in credit scoring. *Journal of Applied Statistics*, 48(13-15):2734–2754.

Gareth James, Daniela Witten, T. H. R. T. J. T. (2013). *An Introduction to Statistical Learning: with Applications in Python*. Springer.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Khandani, A. E., Kim, A. J., and Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787.

Kim, M., Kim, J., Bento, J., and Song, G. (2023). Revisiting self-training with regularized pseudo-labeling for tabular data. *arXiv preprint arXiv:2302.14013*.

Maldonado, S. and Paredes, G. (2010). A semi-supervised approach for reject inference in credit scoring using svms. In *Advances in Data Mining. Applications and Theoretical Aspects: 10th Industrial Conference, ICDM 2010, Berlin, Germany, July 12-14, 2010. Proceedings 10*, pages 558–571. Springer.

Mancisidor, R. A., Kampffmeyer, M., Aas, K., and Jenssen, R. (2020). Deep generative models for reject inference in credit scoring. *Knowledge-Based Systems*, 196:105758.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, 26.

Nick, T. G. and Campbell, K. M. (2007). Logistic regression. *Topics in biostatistics*, pages 273–301.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. (2021). Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568.

Rizve, M. N., Duarte, K., Rawat, Y. S., and Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Y., Dobriban, E., and Davidson, S. (2020). Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning*, pages 10355–10366. PMLR.

Yang, X., Song, Z., King, I., and Xu, Z. (2022). A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhu, X. J. (2005). Semi-supervised learning literature survey.