University of Patras

Machine Learning Algorithms for Financial Fraud Detection

Konstantinos Benekos

«Applied Economics and Data Analysis»

Department of Economics

School of Economics and Business Administration

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science

08/2024

University of Patras, Department of Economics

Konstantinos Benekos

Dissertation Committee

Research Supervisor:	Manolis Tzagarakis	Associate Professor
Dissertation Committee Member:	George Filis	Assistant Professor
Dissertation Committee Member:	Victoria Daskalou	Specialized teaching staff

The present dissertation entitled

"Machine Learning Algorithms for Financial Fraud Detection"

was submitted by **Konstantinos Benekos, ID 1067754** , in partial fulfillment of the requirements for the degree of Master of Science in *"Applied Economics & Data Analysis"* at the University of Patras and was approved by the Dissertation Committee Members.

I would like to dedicate my dissertation to my late father, Polyzois Benekos, who never had the chance to watch me grow up.

Acknowledgments

Firstly I want to thank God almighty for his presence and guidance over me and all my academic achievements. Next, I ought to aknwoledge the guidance and expertise of Manolis Tzagarakis without whom the completion of this study would not have been possible. I would also like thank George Filis and Victoria Daskalou for sitting on my panel and taking the time to read my thesis. Last but not least I want to thank my mother, Anna Rau, my brother, Theofilos A. Benekos and all of my dear friends who supported my rigurosly throughout this process. Without any of you non of this would have been possible.

" If your time to you is worth saving then you better start swimming or you'll sink like a stone for the times they are A-changing" (Bob Dylan).

Abstract

This here study explores the effectiveness of machine learning algorithms in detecting credit card fraud. Specifically, the algorithms evaluated include random forest, logistic regression and support vector machines. The dataset utilized was obtained from Kaggle and includes data regarding 287,807 credit card transactions which were conducted by European card holders over a two day period in September of 2013.

After following standard data prepossessing procedures we continue by addressing the issue of imbalance in the data by proposing a resampling technique which entails under sampling the majority(non-fraud) class in order to create ten distinct balanced subsets of data, each of which captures a different aspect of the majority data. The models are then trained on each of the subsets separately and the final prediction made is determined by majority voting. For comparison, we also trained the models on the raw-unbalanced data in order for us to determine the effects our resampling technique had on the models overall performance. We evaluated the models both from a quantitative perspective leveraging metrics such as accuracy, recall, precision, F1 scores and most importantly ROC-AUC scores as well as in business context in the sense of how applicable/useful they would be in a real world scenario.

The results revealed that while all models trained on the balanced subsets performed fairly well in terms of accuracy, recall and ROC-AUC scores they were all severely lacking in precision, something which in the fraud detection context poses a major issue. For this, it was decided that the our resampling technique negatively affected the results of our models as it made them too sensitive in predicting the fraud class which proved to be a drawback when applied to real world data. Next, when looking at the models trained on the unbalanced data we observed that our best performer in term of ROC-AUC score, which since we are talking about unbalanced classification, is our most important metric, was logistic regression with a score of 99.06%. Furthermore, logistic regression brought to the table the added advantage of transparency, in the decision making process, as well as it not needing any prior assumption of the distribution of the data. For those reason we concluded that logistic regression arose as the most well suited model

for fraud prediction in this particular data set.

Keywords: Credit card fraud, fraud detection, machine learning, logistic regression, random forest, support vector machines, data imbalance

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

As technological advances continue to gallop at unimaginable speeds it seems that the metaphorical Pandora's box has been opened, since the growing dependence of the world, and especially the financial sector, on digitized tools has lead to cybercrimes, like credit card fraud, which already posed a substantial issue, to be exacerbated to unbeknownst heights. These phenomena has adverse effect on both individuals who fall victim to such schemes as well various private and public organizations who are left to pick up the pieces when the dust finally settles. While there are, as one would imagine, mechanisms acting as safeguards against credit card fraud in place their shortcomings have been made evident by not only the overwhelming volume of fraudulent credit card transactions committed but also by the ever evolving landscape of fraud with which they struggle to keep up. As a result researchers have turned to data driven approaches, such as machine learning, to provide new and innovative solutions to this problem. In this study the aim is to assess the effectiveness of three distinct supervised machine learning algorithms(Random Forest, Logistic Regression, Support Vector Machines) in detecting fraudulent credit card transactions occurring in a real-life data set. Additionally we examine a resampling technique, implemented with the

purpose of dealing with the issue of imbalanced in the data, something which is usually associated with credit card fraud data sets, and the effects it has on the results of these models.

The structure of the study is as follows: in the second chapter we delve deeper into the background of the problem as well as further elaborate on our motivation in conducting the research at hand, next in the literature review we look at some of the studies already done on the subject and what lessons their results yield for us. In the fourth section we go into detail about the specific methodology adopted, moving on to the empirical results of our analysis we compare the results not only between the different models themselves but also between the models trained on the data produced by our proposed resampling process and those trained on the raw-default data set. And finally in the last section we summarize our conclusions and outline the limitations of our study hoping to prompt further investigations in the areas where this here research is lacking.

Chapter 2

Background

2.1 Background

2.1.1 Background of the problem

Credit card fraud is on the rise, the effects of cybercrime which is enabled by the mounting digital dependence of the financial system have manifested themselves, partially, in an ever-growing trend of credit card frauds. Credit card fraud involves a number of different strategies such as identity theft and document fabrication. Though credit card fraud already posed a major problem for financial institutions, businesses and consumers alike the boom in e-commerce and online payments, experienced in the last couple of years has significantly exacerbated the issue. The shortcoming of traditional, rule-based, fraud prevention systems is made evident through the sheer number of transactions occurring at any given moment which overwhelm those systems and limit their efficiency significantly. Moreover, the multiple new "path-ways" through which credit card fraud can occur have lead to a constantly shifting fraud environment making it harder for traditional systems to adapt. Hence, it comes to no surprise, that recent research into fraud detection has identified the necessity of data-driven approaches in order to keep up with and conform to new fraudulent patterns in credit card transactions.

2.1.2 Statement of the problem and purpose of the study

To tackle the aforementioned issue of the immense increase in credit card transactions and the evolving fraud landscape, we examine the performance of three distinct machine learning techniques in identifying and correctly classifying credit card transactions as either fraudulent or legitimate. By doing so the aim is to provided a nuanced insight into the effectiveness, in terms of accuracy and fairness, each of this techniques provides as well as to highlight their distinct strengths and weaknesses.

2.1.3 Formulation of Research Questions

We address the following research questions:

- How effective are the proposed techniques at detecting fraudulent credit card transactions in real world data set?
- What are their strengths and limitations?
- What are the difficulties in terms of handling highly imbalanced fraud datasets, and how do we overcome them?

2.2 Significance of investigation

The study hold significance to a number of stakeholders, first and foremost financial institutions, on which the weight of developing fraud detection and prevention mechanisms lies. As the volume of fraud rises and the range of the problem increases financial institutions are forced to allocate more and more resources towards fraud prevention in an effort to protect themselves, their reputations and of course their clients. Hence the development and implementation of innovative and improved fraud detection techniques hold a potentially massive benefit for these organizations, as they may provide a more efficient, cost effective solution to a

substantial issue which is only showing signs of growing in the future. On top of that, fraud detection also provides customers with enhanced financial security and the feeling of safety in their credit card transactions. At last, this study provides a thorough insight into the application of different machine learning techniques on a real life fraud data set, something which may assist in the development of robust and reliable fraud detection mechanisms. Given the prevalence of credit card fraud, and the non-stationary nature of fraud in general the potential benefit from leveraging those kind of models becomes more than evident.

Chapter 3

Literature Review

In today's digital age, where the ever evolving technologies are fully incorporated into the financial system, to the degree where one could even say that they form the practical backbone of all financial institutions, more and more new and unbenounced opportunities for fraudulent individuals have arisen. This facilitates the necessity for innovative strategies for recognizing and preventing financial crimes. In the quest for such innovative solutions Machine Learning (ML) algorithms have arisen as a potential measure against credit card fraud. The literature review at hand provides an overview of existing research on the topic of credit card fraud as well as an in-depth look at certain, specific, ML techniques implemented in this study.

3.1 Credit Card Fraud

According to Azhan(2020) credit card fraud can be defined as: "when an unauthorized person uses a credit card for personal use without the approval or knowledge of the card owner and the card issuer doesn't have a clue of what the card is being used for".

While Bhatla, Prabhu, Dua (2003) provide a more extensive definition of credit card fraud as: "When an individual uses another individuals' credit card for personal reasons while the owner of the card and the card issuer are not aware of the

fact that the card is being used. Further, the individual using the card has no connection with the cardholder or issuer, and has no intention of either contacting the owner of the card or making repayments for the purchases made."

3.1.1 Significance of Credit Card Fraud

Credit card fraud is on the rise, with the Federal Trade Commission (FTC) reporting roughly 460,000 cases in 2023, which while being 5% lower than in 2022 is still a staggering 53% higher than in 2019 when the FTC reported a, comparably, mere 278,000 cases. Furthermore it is important to note that credit card fraud topped the charts in identity theft complaints in 2023 (Federal Trade Commission 2023).

The sharp rise in online payment and e-commerce services experienced over the last decade seems to be a pivotal factor, since it opens up a multitude of new path ways for fraudsters to attack credit card transaction. "An awfully very little quantity of data is needed by the assaulter for conducting any fallacious dealing in online transactions."(Mohari et al. 2021). While certain defence mechanisms such as encryption and tokenization are in place to protect credit card transactions, and whilst those methods do prove to be effective to a degree they still have shortcomings(Ileber, Sun, Wang 2022). Dai, Yan, Tang, Zhao, Guo (2016) outlined that due to the card being used as a virtual duplicate of itself during online transactions, new opportunities for fraudsters arise since: "An attacker only needs to obtain few important information of the card (e.g., card ID, secure code) to make a fraudulent transaction on the Internet while the genuine cardholder often does not notice that his card information has been leaked, which may cause a significant financial loss both to the cardholder and credit card company." To further make evident how common and severe an issue identity theft is in the digital environment we refer to Yu and He (2021), whose work on data leakage demonstrated the widespread risk to financial and personal information security exemplified through incident like Facebook's 2018 break which exposed 87 million

user's data and the fact that in 2014 about 50% of leaks occurred in the business sector, showcasing the vulnerability of commercial entities. The financial magnitude of the issue becomes more evident as we look at the statistics on credit card fraud within the euro area, where in 2021 alone the total volume of Card-not-present fraud[1] amounted to a value of 1.28 billion euros(European Central Bank 2021).

3.1.2 Who does credit card fraud affect?

Bhatla, Prabhu, Dua (2003) established that while banks/ credit institutions and card holders do experience some impact from credit card fraud most of the direct financial and reputational consequences are for the merchants to bear. Under most legistlations/bank policies consumers(card holders) face limited financial liability due to consumer protection laws which are in place as well as "Bank specific" standards that limit the amount for which a consumer can be held accountable in the event of fraud. Based on scheme rules defined by both Master card and Visa the issuer of the card does bear, in some cases, the burden of fraud, yet even in the scenario where the issuer does not bear any direct cost they still have to cover certain indirect costs like administrative and manpower expenses related to managing charge-backs. Furthermore, the issuer has the responsibility of preventing fraud, this goes hand in hand with making huge investments into advanced IT systems for detecting fraudulent transactions. Merchants on the other hand, are those who suffer the most on cases of fraud since, especially in card-not-present transactions, the full direct, along with the indirect costs associated with fraud (Administrative etc.) are laid upon them. Also very importantly merchants suffer a significant reputational loss in cases of fraud since high charge back rates can lead to penalties, loss of card acceptance agreements which in turn can serve as a deterrent for customers.

[1]**Card-not-present fraud: i.e. fraud conducted remotely in online and telephone payments, using card details obtained by scams such as phishing (Source: European Central Bank 2021)**

While consumers seem to be the least impacted party in cases of fraud it is important to take into consideration the psychological effect fraud has on those falling victim to it, this was further analyzed by Kemp, Perez (2023) who examined the effect of consumer fraud amongst an, age-specific group of individuals. Further, "Individuals may suffer significant emotional distress through fraud, as well as the burden of recovering their stolen identity and assets" (Spathis 2002).

3.1.3 Types of Credit Card Fraud

As one might expect most financial institutions already have fraud prevention mechanism in place that alert customers in cases of suspicious activities, yet this has pushed fraudster to advance their operations/techniques to overcome those ever evolving security barriers(Schaffer 2018).

In this section we will briefly discuss some of the most prevalent fraud schemes, based on the works of Ayorinde(2021), Barker, D'Amato, Sheridon (2008) and Bhatla, Prabhu, Dua (2003):

- **Stolen/Lost card:** This method is the most common. It is usually related with stealing someones physical card or copying the information on the card, to make purchases. In cases where the physical card is not missing, as long as the perpetrator, keeps his spending below a certain threshold the bank wont notify the card holder leaving more than enough room for the fraudster to impersonate the card holder in online transactions.

- **Synthetic Fraud (False application method):** This is when a fraudster applies for a new card using somebody else's information. Of course this entails the fraudster firstly acquiring the necessary information.

- **Data Breach:** As stated earlier due to the massive increase in online transactions, hackers get the opportunity to gain access not only to a users personal data, enabling them to commit synthetic fraud, but in some cases they might even be able to completely take over an individuals computer or smart

phone.

- **Mail interception:** Usually goes hand in hand with a fraudster obtaining an individuals address, after they apply for a new card, and intercepting the card before the actual owner get to it.

- **Skimming:** Card skimming involves stealing information of a card during a legitimate transaction usually with the use of a special device know as a "wedge" through which the card is put. This device stores all relevant information off the card and allows the fraudster, through the encrypted verification code, to "fake" confirm his validity as the card holder during transactions (Shannon 2008). The most common ways this is used is either to make counter-fit cards(ACFE, 2007, p. 1.104) or to charge a "fee" every-time the actual owner makes a transaction (Ayorinde 2021). Again this kind of fraud is exacerbated by e-commerce and online payments since if someone enters their card details into a fraudulent website the information provided can be easily copied and reproduced.

While the above mentioned fraud techniques could be categorized as consumer related frauds, we are now going to take a look at techniques classified as merchant related fraud schemes. Where the fraud is initiated by either the owners of the merchandise business or one on their employees(Bhatla, Prabhu, Dua 2003).

- **Merchant Collusion:** This occurs when owners or employees conspire to use cardholders information and/or personal accounts to either commit fraudulent transactions themselves or alternatively sell the information to potential fraudsters.

- **Triangulation:** This is a web-based type of scam, where the owner of the fraudulent website offers a good at a heavily discount rate to attract unsuspecting customers. Once the customer enter their card information into the fraudulent website the owner uses said information to make purchases them-selfs.

3.2 Traditional Methods for Fraud Detection

To combat the issue of credit card fraud many Credit Card Fraud Detection Systems (CCFDS) have been developed, yet traditional systems shortcomings, in terms of computational capabilities, are made evident due to the sheer amount of credit card transactions performed, "The growing number of users and payment transactions has brought heavy workloads to these systems. The speed of new transactions coming into the system can reach millions per second while the size of stored historical transactions can reach several PBs or even EBs. In this case, processing detection tasks and model training on so many incoming transactions with a low delay is very hard for most traditional systems."(Dai, Yan, Tang, Zhao, Guo 2016).

3.3 Practical applications of Big Data and ML techniques in the broader picture of digital finance

Machine learning can provide a solution in keeping up with the ever evolving landscape of fraud, since the way through which fraudulent behaviour is identified is using pattern recognition and anomalies detection of features usually associated with fraud. This establishes ML as a powerful tool in fraud detection since, at least in theory, various models can be calibrated on characteristics associated with different kinds of fraud and trained on real-life data sets to achieve high predictive accuracy in managing big data sets, something which traditional methods lack. "One of the biggest challenge for fraud detection systems is the tremendous growing amount of transactions. Current fraud detection systems need to be more effective and scalable in order to handle such large amount of incoming data. Hence, using Big Data technology is the best solution for this problem."(Dai, Yan, Tang,

Zhao, Guo 2016). The rising trend in research on the application of ML for fraud detection is also shown by Ashtiani and Raahemi (2022), and their comprehensive review of relevant literature on financial statement fraud detection. Additionally Žigiene, Rybakovas, and Alzbutas (2019); Amin et al. (2023) looked into the implementation of ML techniques for risk management in commercial processes and prediction of investors intentions respectively. What seems to be a reoccurring theme amongst most relevant literature is that the main challenges that ML models face are those of data quality and the evolution of how fraud is committed. This only makes sense since whenever new methods of fraud arise ML models need to be re-calibrated in order for them to be able to capture the new patterns that point to fraudulent behaviour. This makes it clear that in order to develop efficient methods of fraud detection one must have clearly defined picture of the forms fraud, and of course, the respective patterns associated them, as well as a comprehensive data set which includes all those patterns and where observations are already classified as fraudulent or not, in order to properly train/test the model.

The ability to leverage big data(high frequency: daily, hourly, etc.) have greatly benefited the banking sector for example, Nugroho and Hamsal (2021), who analyze research trends in digital innovations in banking, found that leveraging complex data analysis techniques not only boosts customer satisfaction and help maintain profitability but furthermore allows institutions to make more deeply informed strategic decisions. Since in contrast to just taking into consideration traditional analytical and theoretical approaches, the usage of big data, and the overall digital innovations that go hand in hand with using/analyzing those kind of data sets, allows for an inflow of information directly originating from patterns arising in the actual data and hence can provide a more nuanced and hands on approach to identifying the root causes behind certain phenomena.

Another study emphasizing the importance of big data is Melnychenko, Volosovych, Baraniuk (2020), who in their endeavour to identify and define dominant concept of financial technologies in digital banking, underscore that the ability to generate,

store and analyze large amounts of user-specific banking data gives organizations the edge so mush so in creating personalized banking services as well as a multitude of other areas such as customer behaviour analysis, monitoring of transaction costs, fraud management, risk assessment, investment decision making , process automation etc.. The authors demonstrate how the implantation of complex data mining technologies withing key areas of the banking system in Ukraine could enhance efficiency, reduce risk and uplift customer satisfaction while at the same time reducing costs.

Overall, the results current literature produces strongly encourages the further exploration and implementation of data driven techniques to not only further our knowledge of the "as is" systems and all their parameters already in place, but to furthermore find new and innovative ways to identify and cover for inefficiencies and malpractices.

3.4 Application of Big Data technology an ML techniques in Credit Card Fraud Detection

Seeing the various ways in which machine learning and big data are leveraged in the digital finance sector, in general, it should come to no surprise that there is a multitude of research on implementing those techniques on credit card fraud detection as well. Maniraj et al. (2019) exemplify a modeling approach for credit card fraud detection, the aim of the authors is to detect transactions that are 100% fraudulent, hence trying to annihilate possibility of incorrect fraud classification. The anatomized data set they used was a result of a PCA transformation, multiple anomaly detection algorithms were employed resulting in a staggering maximum 99.6% accuracy in fraud prediction, with a though somewhat disappointing 33% precision rating. They attribute this disparity to the highly unbalanced nature of the data.

Awoyemi et al. (2017) investigated the performance of naïve Bayes, k-nearest

Neighbor, and logistic regression classifiers on highly skewed credit card fraud data. The performance of the models was assessed based on accuracy, sensitivity, specificity, precision, Mathew's correlation coefficient and flat classification rate. After comparing the methods it was concluded that K-NN outperforms both naive Bayes and Logistic regression.

Mohari et al. (2021) established the advantage machine learning techniques have in tackling the issue of rising fraud cases, since they are compatible to evolving as fraud techniques evolve and since they give an edge in dealing with large amounts of data(transactions). During their study they examined the performance of ten such techniques leading them to conclude that Local Outlier Factor accuracy score is the highest compared to the rest.

In a similar manner Parmar et al. (2020) considered several techniques for fraud identification namely K-Nearest Neighbor, Support Vector Machine (SVM), Decision Trees, Logistic Regression, Random Forest, and XGBoost. The results are ranked based on accuracy, F1 score and the confusion matrix, the findings support that various rule-based methods can be applied for fraud detection resulting in very high accuracy hence ensuring that most flagged transactions are correctly identified as fraudulent.

Shirgave et al. (2019) also reviewed credit card fraud detection using machine learning, they compare the techniques implemented based on accuracy, precision and specificity. Moreover they propose a fraud detection system which is based on a supervised Random Forest classifier, their model yields an increased precision score while also addressing the issue of drift in fraud recognition.

Warghade et al. (2020) improved fraud detection by avoiding miss classification of legitimate transactions as fraudulent. They recommended using synthetic sampling methods such as SMOTE in combination with advanced boosting methods such as Local Outlier Factor(LOF), isolation forest and Support Vector Machines(SVM). These methods combined with the proposed sampling techniques proved to be fast and robust providing high accuracy in fraud detection.

More et al. (2021) showed a comparative analysis of three classifier namely, Decision Tree, Naive Bayes, and Random Forest. With a specially targeted focus on the Random Forest classifier, which outperformed the other models in terms of accuracy precision and recall(sensitivity), the study concluded that even with a highly unbalanced set the model worked well yielding an accuracy 0f 97.93% thus making it highly useful tool in real world fraud detection scenarios.

Sadineni (2020) following a similar approach compared Support Vector Machine(SVM), Artificial Neural Network (ANN), Decision Trees, Logistic Regression and Random Forest on credit card fraud detection. The models performance was once again judged based on accuracy, precision and false alert rate metrics(precision). The result show that ANN's achieved the highest accuracy of 99.92% with random forest following closely behind at at 99.21% accuracy rate. Logistic regression and SVM on the other hand achieved accuracy scored of 95.55% and 95.16% respectively.

3.5 Challenges in ML fraud detection

"Machine learning (ML) application in fraud detection presents various challenges due to factors such as the dynamic nature of fraud patterns, data quality and variability, model interpretability, and dataset balance."(Uwaoma 2024).

3.5.1 Data requirements

As outlined by Dal Pozzolo, Caelen, et al. (2014), a persistent issue in credit card fraud detection is the availability, or rather the unavailability, of relevant real life data sets to train and test realistic models. This is due to the confidentiality issues surrounding the information usually associated with credit card transactions. Furthermore the study outlines the issue of maintaining such a data set, since fraud patterns are non-stationary and constantly evolve, one must either periodically rerun the algorithms on a new data set(including all new relevant data patterns)

or have it continuously update through online learning. Since, "The performance of ML models dramatically depends on the availability of varied and high quality training data." one could perceive the acquisition and maintenance of a adequate dataset as the cornerstone for any analysis on the the topic.

3.5.2 Model Quality and Availability

While ML are proven to be of particular high accuracy they come short in terms of transparency and explain-ability of their decision making process, this phenomena is referred to as the "black-box" problem. This issue has deterred users from accepting the use of ML models in high stakes domains such as fraud detection. On the other hand logistic regression is cited as a model that offers high transparency and clear explanations in it's decision making process, yet there seem to be concerns around whether or not logistic regression can match ML models prediction accuracy (Yang, Wu 2021).

3.5.3 Imbalanced data sets

Another common issue facing fraud detection datasets, mentioned throughout most of the literature is that of the high skew typically associated with those kinds of data. The imbalance between the fraud and non-fraud classes, typically highly skewed toward the non-fraudulent observations, can lead to bias towards the majority class (Abdallah, Maarof, and Zainal 2016). Dealing with this issue involves either data-level techniques like oversampling minority and under-sampling majority classes using methods such as SMOTE, or alternatively making use of algorithm level techniques that adjust the models sensitivity to the minority class. Additionaly ensemble techniques like EasyEnsemble, that combine balanced subsets to improve accuracy, are mentioned as a solution to this hurdle (Dal Pozzolo, Caelen, et al. 2014).

3.5.4 Datasets

The datasets leveraged by most other relevant literature vary from either publicly available sources like Kaggle (Maniraj et al. 2019;Awoyemi et al. 2017;Randhawa et al. 2018) or from private sources like banks or other financial institutions(Dal Pozzolo, Boracchi, et al.2018). While some other authors mention the use of synthetic data sets it is established that real-world data offers far greater robustness and accuracy since with synthetic data often the issue arises that they are not able to capture certain patterns associated with fraud on a realistic basis as they are more or less calibrated on just certain types of fraud, making the models trained on these sets less reliable when applied to real life scenarios(Randhawa et al. 2018).

3.6 Machine learning models for fraud detection

As is made evident by the previous section there has been extensive, to say the least, research into credit card fraud detection with the use of machine learning algorithms. Both supervised and unsupervised techniques have been proposed, unsupervised methods, which consist of outlier/anomaly detection, have the advantage of better being able to unbiasedly handle unbalanced data sets while furthermore being able to handle unclassified transactions. Supervised method, on the other hand, are by far the more popular in current literature, they take as input labeled transactions, and either train the classifier to directly be able to distinguish between fraudulent and non-fraudulent transactions, based on the feature vector of transactions provided, or alternatively they compute a probability score for each transaction indicating how likely it is to fall into each category (Dal Pozzolo, Boracchi, et al. 2018).

In this subsection I will briefly go over the three supervised methods chosen for my analysis(Logistic regression, Random Forest and Support Vector Machines) and discuss their strengths and weaknesses.

3.6.1 Logistic regression

Admittedly, the easiest and one of the most commonly used supervised learning techniques used in relevant literature. Logistic regression (Logit), makes use of a sigmoid function to describe the relationship between the dependent and independent variables, the output it produces is either binomial or multinomial. It is typically used, in the credit card fraud detection context, to classify a transaction as fraudulent or not (Sadineni 2020). It's strengths lie in the fact that, as stated earlier, logit offer the advantage of going around the "black-box" problem by offering transparency in its decision making process through the fact that the coefficients(weights) assigned to each independent feature give a clear picture of how much they affected the probability of classification (Yang, Wu 2021) while furthermore making no assumptions about the scattering of classes in the feature space. It's weakness are that it assumes a linear relationship between the dependent and independent variables, and moreover there are concerns of over-fitting in high dimensional data-sets.

3.6.2 Random Forest Classifier

Random Forest(RF), use a bagging approach to create a big number of separate decision tress, each tree makes a class prediction, the class with the maximum votes is then considered as the ultimate prediction. It's strengths are that feature selection is unnecessary and the model quickly trains and balances errors. However, RF has been proven to be sensitive to data with diverse values and attributes containing more than one value(Sadineni 2020).

3.6.3 Support Vector Machines

Support Vector Machines (SVM), aims to find the best boundary(hyperplane) that separates the two classes(fraud/non-fraud). The hyperplane is chosen with goal of maximizing the distance between the nearest data points of each class and the

hyperplane itself. The data points closest to the hyperplane, which determine it's position, are called support vectors. The SVM model is trained using historical data and finds the hyperplane which separates the support vectors of each class with the maximum possible margin. When a new transaction is processed the model determines in which side of the hyperplane the new data point lies, and thus, whether it is fraudulent or not. SVM models advantages lie in the fact that they are good in dealing with structured and semi structured as well as high-dimensionality data while maintaining a low risk of over-fitting, yet due to their high computational expense they take up significantly more time with larger data sets(Sadineni 2020).

3.7 Comparison of models performance

As for the comparison of the models it was decided to compare them in two different contexts, first we look at the quantities assessment of the algorithms, based on metrics usually associated with machine learning algorithms. For this I consulted the guidelines outlined by Nami, Shajar (2018);Sadineni (2020);Warghade et al. (2020), amongst others, who evaluated the models based on the results of their adjacent confusion matricides. More specifically they used the computed counts of True Positive(TP), True Negative(TN), False Positive(FP) and False Negative(FN) to calculate the models accuracy as a proxy of the overall correctness of the models classifications, recall as a measure of how many actual fraud vases the model was able to correctly identify and precision which showcases the proportion of a actual fraud cases predicted as fraudulent to the total number of fraud prediction the models made. Furthermore the authors included the F1 score, which is the harmonized mean of precision and recall, as it is particularly useful and showing the trade-off, which is often observed, between those two metrics. Finally taking in to consideration the observations of Ayorinde(2021) and Dal Pozzolo, Caelen, et al. (2014) it was decided to also include the ROC-AUC

score (Receiver Operating Characteristics Area Under Curve) since the authors highlighted this performance metrics as arguably the most important subsequent to the fact that confusion matrix accuracy is a lot of times not meaningful in unbalanced classification.

Another plane on which we aught to compare the models is of course the business value they bring to the table and how applicable each of them is in real world context. Dal Pozzolo, Caelen, et al. (2014) highlighted the significance of timeliness of the decision making process, "a card should be blocked as soon as it is found victim of fraud, quick reaction to the appearance of the first can prevent other frauds", hence it becomes evident that, since faster reaction time can prevent more fraud from occurring, the timeliness factor is directly embedded within the "cost" of the fraud detection model. Moreover, as we already assessed earlier, transparency in the decision making process also plays a tremendous role since in such high stakes domains as credit card fraud the decision making behind should be very clear as the consequences of each decision are significant for all parties involved (Yang, Wu 2021).

3.8 Results

Overall throughout the literature there doesn't seem to arise one model which could be deemed as universally optimal. For example Sadineni (2020) results exhibited RF as superior to Logit and SVM while on the other hand in the work of Parmar et al. (2020) SVM outperformed RF. This can be attributed to a number of factors, Sisodia, Reddy, and Bhandari (2017) demonstrated, for one, that model performance varies across different data sets and different kinds of fraud. Furthermore the way in which different studies address the issue of class imbalance also plays quite the significant role since as exemplified by Awoyemi, Adetunmbi, and Oluwadare (2017) as well as Ileberi, Sun, and Wang (2021) addressing class imbalance through resampling boosts results, yet the resampling technique chosen

may influence the performances of the models implemented.

Chapter 4

Methodology

4.1 Methodology

In this section of the study I will go over the methodology adopted in order for us to distinguish and correctly classify the fraudulent transactions from the non-fraudulent ones. Figure 4.1 showcases a generalized outline of the steps taken during our endeavour. Before we delve deeper into the specifics behind each step taken I first want to discuss our main "ingredient", the dataset.

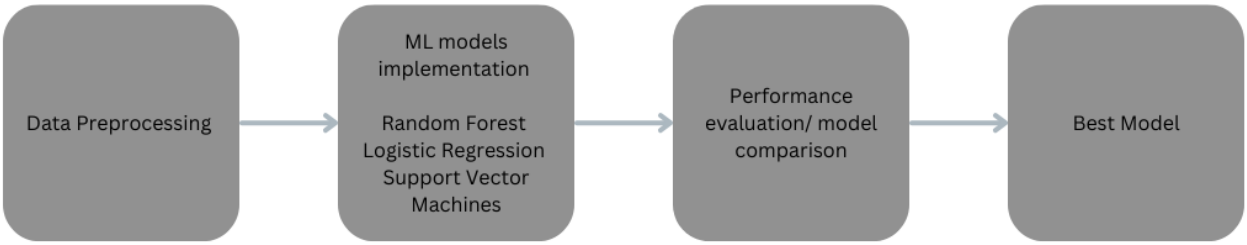

Figure 4.1: Outline of steps taken. Figure created using Canva

4.1.1 Dataset

The dataset used was obtained from Kaggle[1], it presents relevant data linked to transactions made over a time period of two days during September 2013 by European cardholders. The total number of transactions included amount to 287,807 out of which a mere 492 were fraudulent, hence the highly unbalanced nature of the data becomes evident with the positive class(fraud) accounting for just 0.172% of total transactions. The dataset was collected and analyzed during a research collaboration of the Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles)[2] on big data mining and fraud detection.

The data set contains a total of thirty feature all of which are strictly numerical, most features are the result of a PCA transformation (features V1-V28), in order to keep in line with confidentiality concerns, no more information for those features is provided. The only features that are as is are "Time" and "Amount" the former representing seconds between a transaction and the first transaction recorded, while the latter representing transaction amount. Finally, feature "Class" is a binary variables indicating whether or not a transaction is fraudulent. Class=0 indicates that the transaction is legitimate while Class=1 indicates that it is fraudulent.

Statistic	Non-Fraudulent (Class 0)	Fraudulent (Class 1)
Count	283,253	473
Mean	88.41	123.87
Std Dev	250.38	260.21
Min	0.00	0.00
25%	5.67	1.00
Median	22.00	9.82
75%	77.46	105.89
Max	25,691.16	2,125.87

Table 4.1. *Descriptive statistics for the Amount attribute by Class*

Table 4.1 displays some descriptive statistics for the amount attribute by class

[1]The data set can be found here:link
[2]link: http://mlg.ulb.ac.be

(fraudulent, non-fraudulent), we can see that while the mean of 123.87€ for the fraudulent class is higher than the mean of 88.41€ of the non-fraudulent class, the median of the two classes tells a different story since for the non fraud class it amounts to 22€ while for the fraud class it is equal to 9.82€. When furthermore taking into consideration the slightly higher standard deviation of the fraudulent class (260.21 > 250.38) and the significant difference in maximum value for each class we begin to suspect that overall fraudulent transactions tend to be of a smaller amount than non fraudulent ones. To further elaborate on this point we continue by presenting the box-plot and the density plot for the amount feature again by class.

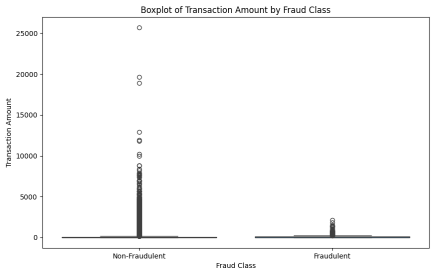

Figure 4.2: Box-plot of Amount feature by class. Created using matplotlib in python.

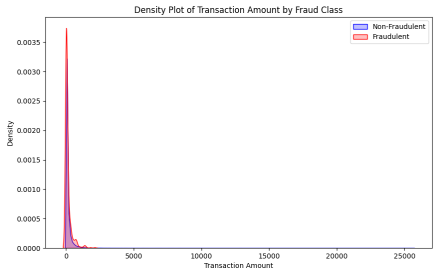

Figure 4.3: Density of Amount feature by class. Created using matplotlib in python.

Our suspicions are confirmed when looking at the box-plot(figure 4.2) as it is made evident that non-fraudulent transactions tend be of a higher amount while fraudulent ones move within a lower range. When looking at the density plot(figure 4.3), on the other hand, one can observe that the distributions of both classes exhibit similar patterns and tend to involve smaller amount far more often than not.

4.2 Data prepossessing

Data prepossessing is an essential step before implementing any machine learning algorithm, it covers all activities related to assemble the final data set used for modeling. Different models may require different specifications for their predic-

tions, and the way data is trained/treated significantly affects the outcome. In this project the prepossessing phase entails detection and deletion of missing values & outliers, splitting the data set into training and testing subsets, handling class imbalances using a resampling technique, which involves under-sampling across multiple subsets of the original data, and finally scaling/standardizing features. To achieve this I made use of data manipulation library pandas and machine learning library sci-kit learn in python. The steps taken are illustrated in figure 4.4.

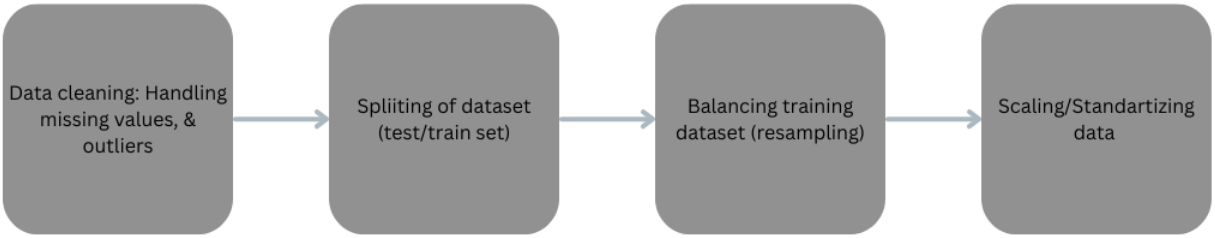

Figure 4.4: Outline of steps taken. Figure created using Canva

4.2.1 Data cleaning

During the data cleaning process two actions were performed, 1) detection/deletion of missing(NA)/null values and 2) handling of outliers. There were neither null nor missing values in our data set, hence we directly move forward to handling outliers.

Outliers can be defined as observations that are numerically distant from the rest of the data. We identified outliers using the Inter Quantile Range(IQR)[3] technique, in this technique any data points above the upper-bound of Q3 + 1.5*IQR or below the lower-bound of Q1-1.5*IQR are considered to be an outlier and are eliminated with purpose of making the machine learning algorithms more

[3]Where IQR= Q3-Q1

accurate and robust.

Interestingly, upon eliminating all outliers across all independent variables, we found that we were left without any fraudulent cases in the dataset. This encouraged a deeper investigation which lead us to print a summary of outliers for each independent variable by class[4]. The results revealed a high correlation amongst outliers in the PCA transformed attributes (V1...V28) and fraudulent observations. Hence, by removing outliers in these attributes we effectively eliminated all fraudulent cases in the data set.

Given the lack of information about the nature of these PCA transformed attributes it was decided to keep the outliers found in these features. Instead, only the outliers found in the "Amount"[5] feature were removed, this approach aims to preserve the integrity of the fraud class while still addressing significant outliers in the data set.

As a reference we provide the box-plot and density plot of the "Amount" attribute, by class, after the removal of outliers:

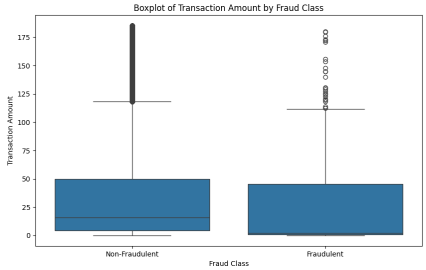

Figure 4.5: Box-plot of Amount feature by class, without outliers. Created using matplotlib in python.

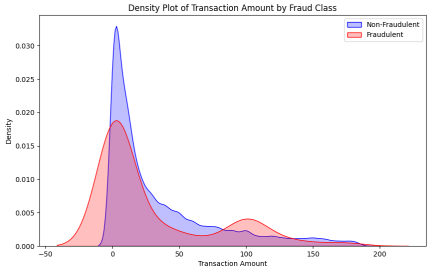

Figure 4.6: Density of Amount feature by class, without outliers. Created using matplotlib in python.

After the cleaning of the data we are left with a total of 252,041 observations out which 386 are classified as fraudulent and 251,655 as legitimate.

[4]The full summary can be found in the Appendix
[5]They have been visualized in Figure 4.2

4.3 Splitting of data set

In this analysis the dataset is split into training and testing sets using stratified sampling to ensure that the class distribution amongst fraudulent and non-fraudulent cases remains the same as is in the original data, in both sub-samples. Specifically, 80% of the data is allocated toward training while the remaining 20% is used for testing. The stratified sampling is achieved through the use of the "train_test_split" function from the Scikit-learn library in python, with the "stratify" parameter set to the target variable (Class) to ensure the the proportion of fraudulent and non-fraudulent remains the same in both subsets. Furthermore, a fixed random state is used to ensure the reproducibility of the results.

Class Distribution	Class 0 (%)	Class 1 (%)
Original Data	99.841441	0.158559
Training Data	99.841342	0.158658
Test Data	99.841838	0.158162

Table 4.2. *Class Distribution in Original, Training, and Test Data*

4.3.1 Balancing dataset

As mentioned earlier in this study our data sets is highly unbalanced, when a model is trained using those kind of data sets it can lead to bias towards the majority class when predicting the class of a new observation. One common way of dealing with this issue is through under-sampling of the majority class in order to achieve an analogy of 1:1, yet as highlighted by Liu, Wu and Zhou(2009), a major drawback of this strategy is that it potentially "throws away" useful information that could be found in the majority class.

To overcome this issue, and inspired by the EasyEnsemble methodology, we started by creating ten balanced subsets from the original data, each subset is constructed by sampling a portion of the majority class so that it matches the size of he minority class, ensuring even distribution amongst classes within each

subset. By schuffeling the indices of the of the majority class in each iteration we ensure that the selected indices for the majority class are different for each subset, thus creating balanced subsets for which, the sample of the majority class contained, each captures a unique aspect of the majority class data. After creating the subsets we train separate models on each one, and then we aggregate the prediction produced by each model through majority voting, in other words for each instance the final prediction is determined by the most common prediction amongst all models. This methodology allows us to leverage insights from all subsets, hence ensuring diverse representation of the data and leading to a more robust and comprehensive evaluation of performance.

The pseudo-code for this sampling methodology is shown in **Algorithm 1**.

4.3.2 Feature scaling

The next stage in our prepossessing methodology is to normalize the range of independent variables within the dataset. The adopted scaling technique centres the data at mean of zero with a standard deviation of one. When having widely varying values it can cause certain machine learning algorithms(like Logistic regression and SVM) to perform poorly due to bias created towards features with larger ranges. To prevent this from happening we perform feature scaling using StandartScaler. Scaling entails subtracting the mean of each feature and dividing by the standard deviation, it ensures that all features contribute equally to the model and prevents the dominance of certain features due to their scale. **Algorithm 2**, showcases our feature scaling process where each subset of data is scaled independently, and then the test data is scaled based on the parameters derived from the scalation of the training data.

Algorithm 1 Balanced Subset Creation and Aggregated Prediction

Require: X: Features of the dataset
Require: y: Target variable
Require: $n\_subsets$: Number of subsets to create
Require: $random\_state$: Seed for reproducibility (optional)
Ensure: $final\_predictions$: Aggregated predictions on the test set

Separate the minority and majority classes:
$X\_minority \leftarrow X[y == 1]$
$X\_majority \leftarrow X[y == 0]$
$y\_minority \leftarrow y[y == 1]$
$y\_majority \leftarrow y[y == 0]$
$subsets \leftarrow []$
$majority\_indices \leftarrow y\_majority.index.tolist()$
for $i = 1$ to $n\_subsets$ **do**
  if $random\_state$ is not None **then**
    $np.random.seed(random\_state + i)$
  end if
  $shuffled\_indices \leftarrow np.random.permutation(majority\_indices)$
  $selected\_indices \leftarrow shuffled\_indices[: len(X\_minority)]$
  $X\_majority\_sample \leftarrow X\_majority.loc[selected\_indices]$
  $y\_majority\_sample \leftarrow y.loc[selected\_indices]$
  $X\_subset \leftarrow pd.concat([X\_minority, X\_majority\_sample])$
  $y\_subset \leftarrow pd.concat([y\_minority, y\_majority\_sample])$
  $subsets.append((X\_subset, y\_subset))$
end for
$all\_predictions \leftarrow np.zeros((len(X\_test), n\_subsets))$
for $i = 1$ to $n\_subsets$ **do**
  $X\_subset, y\_subset \leftarrow subsets[i]$
  $model \leftarrow TrainModel(X\_subset, y\_subset)$
  $all\_predictions[:, i] \leftarrow model.predict(X\_test)$
end for
$final\_predictions \leftarrow$
  $[np.bincount(predictions.astype(int)).argmax()$
   for $predictions$ in $all\_predictions]$
Evaluate the aggregated performance:
Compute precision, recall, F1 score, accuracy, confusion matrix, and classification report based on $final\_predictions$ and y_test =0

Algorithm 2 Feature Scaling using StandardScaler

Require: Subsets of data: X_subsets, y_subsets; Test data: X_test
Ensure: Scaled subsets: X_subsets, y_subsets; Scaled test data: X_test_scaled
Initialize the scaler: scaler ← StandardScaler()
for $i = 1$ **to** len(X_subsets) **do**
    Extract the i-th subset: X_subset ← X_subsets[i]
    Scale the subset: X_subset_scaled ← scaler.fit_transform(X_subset)
    Update the i-th subset with scaled data:
    X_subsets[i] ← pd.DataFrame(X_subset_scaled, columns=X.columns)
    Preserve the corresponding labels:
    y_subsets[i] ← y_subsets[i]
end for
Scale the test data:
X_test_scaled ← scaler.transform(X_test)
Return scaled subsets and test data:
X_subsets, y_subsets, X_test_scaled
=0

4.4 Machine Learning Models

In this study, I employed three different supervised machine learning algorithms, namely Logistic regression(Logit), Random Forest(RF) and Support Vector Machines(SVM). In this subsection we discuss the creation and the rationale behind each model.

4.4.1 Random Forest

Arguably one of the most popular machine learning algorithms in fraud detection, it is a vast pool separate decision trees that form the so called "forest". This technique can be used both for solving classification problems as well as regression problems. The way it works is that each different tree makes a class prediction, any class that that has the maximum votes is then considered for the ultimate prediction of the model. As mentioned earlier the strengths of this technique lie in the fact that no feature selection is needed and that it quickly trains and balances errors. Yet, it key weakness is that the model is sensitive to diverse values in the data and attributes containing more than one value. The "forest" the

model builds is also known as a decision tree ensemble, which is trained using a technique called bagging. Bagging, and hence also Random Forest, combine multiple models into one making it extremely effective in different types of predictive modeling. "It is one of the best algorithms used in the banking system for fraud detection"(Ayorinde 2021). "When building trees Random Forest always attaches randomness making it essential to find the topmost feature amongst all features for modeling, especially during the splitting of the node"(Donges 2021). Moreover, random forest combats the reoccurring issue of over fitting, in machine learning modeling, due to it creating many distinct decision trees and hence limiting the probability of the classifier over-fitting. An overview of the pseudo code behind how random forest is utilized in this analysis is provided in **Algorithm 3**, it is important to note that our random forest classifier was created using the RandomForestClassifier of the Scikit-learn library in python, while the number of decision trees is set to default as 100.

4.4.2 Logistic Regression

Logistic regression is a very straightforward technique, again, used in both classification as well as regression problems. It makes use of a sigmoid function to describe the relationship between dependent and independent variables, estimating the probability of an output being either binomial or multinomial. In supervised learning it deals with assigning input into one of the predefined categories. In our case the output is binary(binomial) and hence logistic regression predict a response variable as either 0 or 1, using the sigmoid function to map input values to probabilities between 0 and 1. The main advantage logistic classification brings to the table is that it avoids the so called "black-box" problem[6] while simultaneously making no a assumptions about the classes in the feature space, it's main disadvantage lies in the fact that it assumes a linear relationship between the input and output variables.

[6]Look at literature review section 3.6.1 for further information

Algorithm 3 Train and Aggregate Predictions from Random Forest Models

Require: Subsets of data: X_subsets, y_subsets; Test data: X_test, y_test
Ensure: Aggregated performance metrics and evaluation results
Initialize all_predictions ← np.zeros((len(X_test), len(X_subsets)))
for $i = 1$ **to** len(X_subsets) **do**
    Extract the i-th subset: X_subset ← X_subsets[i]
    Extract the i-th labels: y_subset ← y_subsets[i]
    Initialize Random Forest classifier: rf_classifier ← RandomForestClassifier(random_state=42)
    Train the classifier: rf_classifier.fit(X_subset, y_subset)
    Predict on the test set: predictions ← rf_classifier.predict(X_test)
    Store predictions: all_predictions[:, i] ← predictions
end for
Aggregate predictions:
final_predictions ← [np.bincount(predictions.astype(int)).argmax()
    for predictions in all_predictions]
Compute performance metrics:
precision ← precision_score(y_test, final_predictions)
recall ← recall_score(y_test, final_predictions)
f1 ← f1_score(y_test, final_predictions)
accuracy ← accuracy_score(y_test, final_predictions)
Print "Aggregated Performance Metrics:"
Print precision
Print recall
Print f1
Print accuracy
Print "Confusion Matrix:"
Print confusion_matrix(y_test, final_predictions)
Print "Classification Report:"
Print classification_report(y_test, final_predictions)
=0

A generalized equational form of logistic classification as well as a description of its components is provided as follows:

$$\hat{y} = \sigma(\mathbf{w}^T\mathbf{x} + b) \tag{4.1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.2}$$

$$\hat{y} = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{x}+b)}} \tag{4.3}$$

- $\hat{y}$: The predicted probability that the output belongs to class 1.
- $\sigma(z)$: The logistic sigmoid function, which maps any real-valued number into the (0, 1) interval.
- $\mathbf{w}$: The vector of weights (parameters) associated with the features.
- $\mathbf{x}$: The vector of input features.
- b: The bias term (intercept).
- $\mathbf{w}^T\mathbf{x}$: The dot product of the weight vector and the input feature vector.
- e: The base of the natural logarithm.

We build our logistic classification model in python by making use of the LogisticRegression package of the Scitkit-learn library, additionally we computed the average coefficients of the models created (one for each different subset)[7]. **Algorithm 4**, provides an overview of the process implemented.

4.4.3 Support Vector Machines

Support Vector Machines(SVM), aims to find the optimal boundary(hyperplane) that separates the data into the two predefined classes (fraud/non-fraud). The

[7]The coefficients can be found in the Appendix

Algorithm 4 Aggregate Logistic Regression Models

Require: Subsets of data: `subsets` (list of tuples), Test data: `X_test`, Test labels: `y_test`

Ensure: Average coefficients of the logistic regression models

Initialize `all_predictions` as a zero matrix of shape `(len(X_test), len(subsets))`
Initialize `all_coefficients` as a zero matrix of shape `(len(subsets), X_test.shape[1])`
for $i = 1$ **to** `len(subsets)` **do**
    Extract the i-th subset: X_subset ← `subsets`$[i][0]$
    Extract the i-th labels: y_subset ← `subsets`$[i][1]$
    Initialize logistic regression model: log_reg ← `LogisticRegression(max_iter=1000, random_state=42)`
    Fit the model: log_reg.fit(X_subset, y_subset)
    Store coefficients: all_coefficients$[i, :]$ ← log_reg.coef_[0]
    Predict on test data: all_predictions$[:, i]$ ← log_reg.predict(X_test)
end for
Aggregate predictions:
final_predictions ← [np.bincount(pred.astype(int)).argmax()
    for pred in all_predictions]
Calculate performance metrics:
precision ← precision_score(y_test, final_predictions)
recall ← recall_score(y_test, final_predictions)
f1 ← f1_score(y_test, final_predictions)
accuracy ← accuracy_score(y_test, final_predictions)
Print "Aggregated Performance Metrics:"
Print precision
Print recall
Print f1
Print accuracy
Print "Confusion Matrix:"
Print confusion_matrix(y_test, final_predictions)
Print "Classification Report:"
Print classification_report(y_test, final_predictions)
Calculate average coefficients:
avg_coefficients ← np.mean(all_coefficients, axis=0)
Return avg_coefficients
=0

hyperplane is chosen with the goal of maximizing the distance between the nearest data point of each class and the hyperplane itself. The data points closest to the hyperplane, which determine it's position, are called support vectors. The SVM model is trained using historical data and finds the position of the optimal hyperplane which separates the support vectors of each class with the maximum possible margin. When a new transaction is processed the model determines on which side of the hyperplane the new data point lies, and thus, what class it belongs to. SVM models advantages lay in the fact that they are good in dealing with structured and semi-structured as well as high dimensionality data while still maintaining a relatively small risk of over fitting, ye due to it's high computational expense it take significantly more time with larger data sets (Sadineni 2020).

To build our SVM model we made use of the SVC and CalibratedClassifierCV packages from the Scikit-learn library in python. The pseudo-code for the model can be seen in **Algorithm 5**.

SVC stands for Support Vector Classification, it is an implementation of SVM specifically designed for classification tasks. We initializing SVC we set the kernel parameter to linear, this means that the algorithm will seek to find a straight line hyperplane to separate the two classes. Additionally setting the parameter probabilities as "true" allows the SVM to output probability estimates for each for each class. Finally the random state parameter ensures that the model produced during training is reproducible making the results consistent across different runs.

4.4.4 Evaluation

There are two aspects of model evaluation, first we at the quality metrics described in the literature review[8], namely accuracy, precision, recall, F1 score their adjacent confusion matrices and finally the ROC-AUC scores, on the test dataset in order to validate predictive capabilities. Second we consider the relevance of the models in a business context, to do so we consider finding by

[8]Chapter 3, section 3.7

similar studies and reviews by domain experts. Bases on those criteria, we rank the models established on how easy to interpret they are, how accurate their predictions are and last but not least how applicable they are in real world business scenarios.

Algorithm 5 Train and Aggregate SVM Models

Require: Subsets of data $(X_{\text{subsets}}, y_{\text{subsets}})$, Scaled test data X_{test}, Test target variable y_{test}

Ensure: Aggregated predictions and performance metrics

Initialize $all\_predictions \leftarrow \text{zeros}((\text{len}(X_{\text{test}}), \text{len}(X_{\text{subsets}})))$
Initialize $svm\_classifier \leftarrow \text{SVC}(\text{kernel} = \text{"linear"}, \text{probability} = \text{True}, \text{random\_state} = 42)$
Initialize $calibrated\_svm \leftarrow \text{CalibratedClassifierCV}(svm\_classifier)$
for $i = 0$ **to** $\text{len}(X_{\text{subsets}}) - 1$ **do**
    $X_{\text{subset}} \leftarrow X_{\text{subsets}}[i]$
    $y_{\text{subset}} \leftarrow y_{\text{subsets}}[i]$
    $calibrated\_svm.\text{fit}(X_{\text{subset}}, y_{\text{subset}})$
    $all\_predictions[:, i] \leftarrow calibrated\_svm.\text{predict}(X_{\text{test}})$
end for
$final\_predictions \leftarrow$ [np.bincount($pred$.astype(int)).argmax() for pred in all_predictions]
$precision \leftarrow \text{precision\_score}(y_{\text{test}}, final\_predictions)$
$recall \leftarrow \text{recall\_score}(y_{\text{test}}, final\_predictions)$
$f1 \leftarrow \text{f1\_score}(y_{\text{test}}, final\_predictions)$
$accuracy \leftarrow \text{accuracy\_score}(y_{\text{test}}, final\_predictions)$
Print "Aggregated Performance Metrics:"
Print "Precision: ", precision
Print "Recall: ", recall
Print "F1 Score: ", f1
Print "Accuracy: ", accuracy
Print "Confusion Matrix:"
Print $\text{confusion\_matrix}(\text{y}_{\text{test}}, final\_predictions)$
Print "Classification Report:"
Print `classification_report`$(\text{y}_{\text{test}}, final\_predictions)$
=0

Chapter 5

Empirical Results

5.1 Empirical Results

In this chapter we shall present and compare the results and findings of the machine learning algorithms implemented, as discussed in the previous chapter, we will begin by comparing the aforementioned quantitative metrics to evaluate the performance of our model. Next we will compare the results to a performance evaluation of our models without the implementation of our resampling technique to handle the imbalance in dataset, in order for us to establish the effect of our balancing technique. And finally we will discuss the results and compare the models in a business context.

5.2 Evaluation Metrics

The evaluation metrics implemented are based on the definitions given by Nami and Shajar (2018);Sadineni 2020;Warghade et al. 2020. Model performance is assesed based on the values of True Positive(TP), True Negative(TN), False Positive(FP) and False Negative(FN) which are defined as follows:

- **TP:** count of actually fraudulent transactions, classified as fraudulent by the model.

- **TN:** count of actually non-fraudulent transactions classified by the model as non-fraudulent.
- **FP:** count of non-fraudulent transactions incorrectly classified as fraudulent by the mode.
- FN: count of fraudulent transactions incorrectly classified as non-fraudulent by the model.

5.2.1 Confusion Matrix

The confusion matrix displays a complete quantitative breakdown of the models performance based on the aforementioned metrics i.e. count of correct classifications and Type error I, type error II. An illustration of the confusion matrix is displayed in **Table 5,1**:

		Predicted	
		Negative	**Positive**
Actual	**Negative**	TN (Correct)	FP (Type I Error)
Actual	**Positive**	FN (Type II Error)	TP (Correct)

Table 5.1. *Confusion Matrix, general outline.*

5.2.2 Accuracy

Accuracy is defined as the the fraction of correctly classified transactions to the total number of classifications made. While accuracy is good measure of a classification models performance when taken into consideration on it own it can give a false sense of high precision.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TN + TP}{TP + FP + FN + TN} \tag{5.1}$$

5.2.3 Recall

Recall is the ratio of correctly predicted positive observations divided by the total number of observations that should have been predicted as positive. In other words it gives us a sense of how many positive observations(fraudulent transactions) the model didn't predict. In fraud detection this measure is of significant importance since it shows us what percentage of fraudulent transactions managed to cheat the model and be falsely classified as non-fraudulent.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

5.2.4 Precision

Precision, on the other had, is the ratio of correctly predicted positives to the total number of predicted positives. It gives us an idea of how many non-fraudulent cases were falsely classified as fraudulent.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.3}$$

5.2.5 F1 Score

The F1 score can be calculated as the weighted average of precision and recall, and is a measure of accuracy. Since it takes into consideration both the falsely classified fraudulent as well as the falsely classified non-fraudulent transactions it gives a complete picture of how precise and how strong the model is. If for example a model has low recall but high precision it may be incredible accurate in the predictions it made as positive but it may also be weak in the sense that it missed a lot of true positive predictions, which were falsely classified as negative. Overall, the higher the F1 score the better the model performed.

$$\text{F1 Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{5.4}$$

5.2.6 ROC-AUC Score

The final quantitative measure we use to examine the performance of our models is the ROC-AUC score, ROC standing for Receiver Operating Characteristics and AUC standing for Area Under Curve. ROC is a probability curve, it is plotted with the TPR(True Positive Rate) on the y-axis against the FPR(False Positive Rate) on the x-axis. The AUC evaluates class separability, i.e.how good the model is at labeling 0 classes as 0 and 1 classes as 1, across all possible thresholds. A higher AUC indicates that the model is better, than some other benchmark, at distinguishing between fraudulent and non-fraudulent transactions.

$$\text{TPR} = \text{Recall} = \text{Sensitivity} = \frac{TP}{TP + FN} \tag{5.5}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5.6}$$

$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP} \tag{5.7}$$

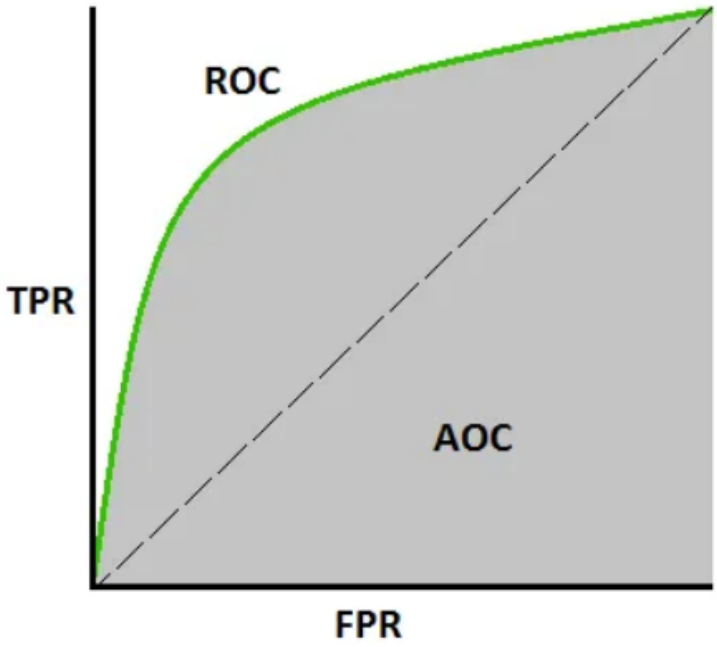

Figure 5.1: AUC-ROC curve, image source: link

5.3 Quantitative Evaluation of models trained on balanced subsets

In this section we will perform a comparative analysis of the aforementioned metrics across our three distinct models. The results presented in this section will be focused on the the models trained using the balanced subsets discussed previously, this will serve as a benchmark for us to compare against those trained on the entire set, hence giving us an overview of how our balancing process affected the results.

	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest	0.9883	0.1135	0.9375	0.2024	0.9955
Logistic Regression	0.986	0.0966	0.9375	0.1752	0.9903
Support Vector Machines	0.9921	0.1567	0.9125	0.2674	0.9905

Table 5.2. *Performance Metrics for Different Models(Balanced subsets)*

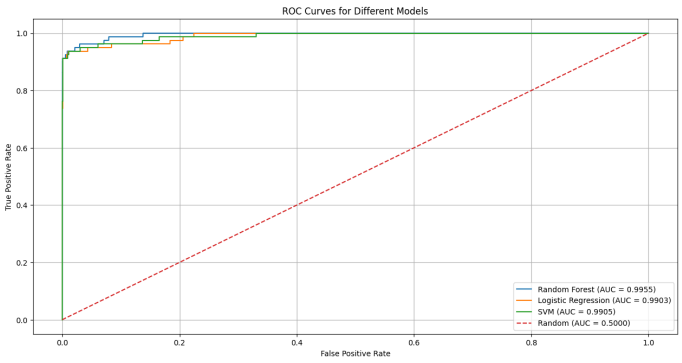

Figure 5.2: AUC-ROC curves for diferent models(balanced subsets)

Table 5.2 displays the performance metrics of our models, while **Figure 5.2** shows their adjacent plots of the ROC curve. Overall, all models seemed to have performed relatively well in the context of accuracy, recall and ROC-AUC score yet all of them seem to be lacking in precision and hence also F1 scores.

5.3.1 Random Forest

The random forest model demonstrates a decent fraud detection ability, having a substantial, confusion matrix based, accuracy of 98.83% and the highest ROC-AUC score amongst the models at 99.55%. Furthermore, the model showcases a significantly high recall score of 93.75% something which implies that the model manages to correctly classify most of the actual fraud cases, even compared to other relevant literature on the subject, those results are quite high. However precision is low in our model standing at a mere 11.35% showing that most positive fraud prediction made by the model are false. This is a big issue since, as already assessed, in the fraud detection domain falsely classifying legitimate transactions as fraudulent comes with significant consequences both financial(rework etc.) and reputations as genuine customers are wrongly accused of fraud. To further investigate this issue we take a look at **Table 5.3** which showcases the confusion matrix for the random forest model. We can clearly see that there is a significantly high number of false positives which is directly associated with this issue. Overall, random forest, while achieving significant success in accuracy, AUC and recall seems to be overly sensitive to predicting the positive class, when being implemented on the unbalanced testing set.

	Predicted Negative	**Predicted Positive**
Actual Negative	49,915	586
Actual Positive	5	75

Table 5.3. *Confusion Matrix for the Random Forest model(balanced subsets).*

5.3.2 Logistic Regression

In similar fashion the logistic regression model displays reasonably good, yet slightly lower when compared to random forest, metrics of accuracy(98.6%) and AUC(99.03%) while also maintaining a high recall rate of 93.75%. Overall, one could say that the model performed well in terms of these metrics yet there seem to

be the same pattern arising, as previously mentioned in the random forest model evaluation, where precision and hence also F1 score are unsatisfactory low. This once again indicated that the model is overly sensitive to making positive class predictions in a real-life unbalanced dataset, something which is problematic in terms of fairness. To further elaborate on that point we present the confusion matrix of the logistic regression in **Table 5.4** where once again we observe a significantly high number of false positive predictions, trumping even those of the random forest model and hence explaining the further fall in precision and F1 score.

	Predicted Negative	**Predicted Positive**
Actual Negative	49,800	701
Actual Positive	5	75

Table 5.4. *Confusion Matrix for the Logistic Regression model(balanced subsets).*

5.3.3 Support Vector Machines

The SVM model, on the other hand, achieved the highest accuracy score out of all the models (99.21%) while also boasting a satisfactory high AUC score of 99.05% and a slightly lower than the rest bust still significant recall score of 91.25%. In terms of precision the SVM models seems to outperform both the random forest as well as the logistic regression. But still the persisting pattern of relatively low precision and F1 score steaming from high number of false positive predictions is concerning. Since this issue seems to be persistent amongst all models it gives us reason to suspect that the main driver behind this is the fact hat the model is trained on balanced subsets, and when implemented on real-world unbalanced sets it tends to be far too sensitive toward predicting the fraud class.

	Predicted Negative	**Predicted Positive**
Actual Negative	50108	393
Actual Positive	7	73

Table 5.5. *Confusion Matrix for the Support Vector Machines model (balanced subsets).*

5.3.4 Comparative quantitative evaluation of models trained on unbalanced data

To further investigate the issue of the reoccurring pattern of high false positive prediction, and the low precision and F1 score these entail, in this section we will present the same quantitative evaluation metrics as before but this time the models from whom these metrics are attained were trained on the unbalanced, raw, data set. This, also, gives us an opportunity to compare the effects of our sampling technique on the performance of each model.

	Accuracy	Precision	Recall	F1 Score	AUC
Random Forest	0.9996	0.9306	0.8375	0.8816	0.9679
Logistic Regression	0.9906	0.8689	0.6625	0.7518	0.9906
Support Vector Machines	0.999	0.9677	0.375	0.5405	0.9135

Table 5.6. *Performance Metrics for Different Models(Unbalanced dataset)*

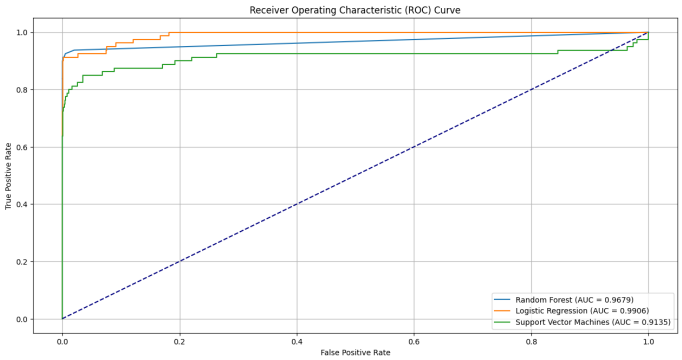

Figure 5.3: AUC-ROC curves for different models(Unbalanced training data)

Random Forest

	Predicted Negative	**Predicted Positive**
Actual Negative	50496	5
Actual Positive	13	67

(a) Confusion Matrix for Random Forest

Logistic Regression

	Predicted Negative	**Predicted Positive**
Actual Negative	50493	8
Actual Positive	27	53

(b) Confusion Matrix for Logistic Regression

Support Vector Machines

	Predicted Negative	**Predicted Positive**
Actual Negative	50500	1
Actual Positive	50	30

(c) Confusion Matrix for Support Vector Machines

Figure 5.4: Confusion Matrices for Random Forest, Logistic Regression, and Support Vector Machines Models (Unbalanced Training Data)

Table 5.3 displays the performance metrics across all different models, **Figure 5.3** shows their adjacent plots of the ROC curve and finally, **Figure 5.4** exhibits every models confusion matrix.

We observe that when the models are trained on the unbalanced data set the, confusion matrix based accuracy increases across the board for all models. Random Forest and SVM[1] seem to exhibit the most significant improvements in accuracy with Logistic regression also increasing yet, to a lesser extend than the other models. In terms of precision we see a dramatic rise in all models, with SVM taking the lead with precision score of 96.77%, closely followed by random which is standing at 93.06% and finally logistic regression with 86.89%. This remarkable improvement in precision confirms our suspicions that the shortcomings, in precision, exhibited by the models trained on the balanced subsets seems to stem from

[1]Since SVM is highly time consuming and computationally expensive we have limited the sample size to 10,000 observations in the interest of saving time, the results are still good and comparable to the others models.

the fact that during the training process they become too sensitive in predicting the fraud class, and when the there are implemented o a real-life data set which is highly unbalanced, i.e. there are significantly less fraudulent cases at hand than there is in the training data, they tend to be overly prawn to predict fraud. However there seems to be a trade-off between precision and recall, since shifts in recall are experienced by all models. Specifically random forest, while showing a slight decline, still maintains a decent recall score of 83.75% while logistic regression and SVM show more severe drop to 66.25% and 37.5% respectively. This drop indicates that while now the models are less sensitive to predicting false positives they have also lost, to an extend, their edge in predicting fraudulent classes overall leading them to miss a larger proportion of fraudulent cases. This trade-off between recall and precision is further reflected in the models respective F1 scores who, since the rise in precision is far greater than the decrease in recall, have risen quite substantially with random forest achieving the overall highest score of 88.16%.

Finally, when looking at our most important metric, AUC, we observe that with the exception of logistic regression, the scores drop. With random forest AUC falling, slightly, to 96.79% and SVM's to 91.35%, logistic regression's AUC score on the other hand remains consistent at a staggering 99.06% rate. These results suggest that when trained on unbalanced data SVM and random forest lose some of their discriminatory ability making them less effective in separating between fraudulent and non-fraudulent transactions.

Overall, this comparative analysis verifies that the models performance is highly dependent on whether or not balanced or unbalanced data is used in the training phase. Whilst when trained on the balanced subsets the models achieve great results in terms of accuracy, recall and AUC scores there are severe shortcomings in precision which pose a severe threat in the context of fraud detection. This issue is resolved when using unbalanced training data since precision increases significantly across the board, yet there seems to be a trade-off in recall

which slightly decrease. Despite this though, the recall rate remains satisfactory and when further taking into consideration the vast leap in precision one could conclude than training the ML models on unbalanced data yield better, all encompassing, results. Moreover, it is important to note that while the AUC score, seems to drop in the case of unbalanced training data, the rates are still pretty good for all models, with logistic regression managing to maintain the same AUC as it had when trained on balanced data.

5.4 Business context & Discussion

In this subsection we will compare the models in a business context, meaning in terms of how applicable/useful they are in a real world business scenario. Furthermore to solidify our conclusions we compare our results to the findings of similar studies, which in some cases, even made use of the same dataset.

We have already established that models trained on unbalanced data produce better results, subsequent to the fact that when trained on balanced subsets of the data the models become too sensitive in predicting the positive class leading to inadequately low precision. In real world scenarios this poses a dramatic issue as misclassifying transactions by actual customers as fraudulent comes at a significant financial cost in terms of the rework that goes hand in hand with correcting the false flagging of the transaction as fraudulent as well as a reputational cost that is consequent to the emotional distress those legitimate customers are inevitably put through(Spathis 2002;Ayorinde 2021;Bhatla, Prabhu and Dua 2003). For this reason we will only consider the models trained on the raw, unbalanced data whose results are far more promising in this context.

Similarly to Sadineni (2020), amongst others, our result highlight random forest as the highest achiever in terms of confusion matrix accuracy, yet when taking into consideration the works of Ayorinde(2021) and Dal Pozzolo, Caelen, et al. (2014) who established that confusion matrix accuracy can be quite a meaningless

metric when assessing the performance of unbalanced classification, we next turn out sight towards the ROC-AUC scores the models achieved. Here, we see that in accordance to Mohari et al. (2021) and Uwaoma(2024), logistic regression arises as the most effective model in terms of separability of the two classes. While, random forest still is the best performer in terms of precision and recall, and hence also F1 score, logistic regression also yields satisfactory results in terms of those metrics. Furthermore logistic regression brings another advantage too the table that both other models lack, namely transparency. While the other models decision making process is not clear when utilizing logit one can see clearly which factor and to what extend influenced the classification processes since the coefficients/weights assigned to each feature give a clear picture of which the main drivers behind the probability of classification are. As already mentioned earlier, in high stakes domains such as fraud detection, transparency is key since at the end of day regardless of the efficiency of any model one must be able to clearly consummate the rationale behind the flagging of a transaction as fraudulent in order to not only be able to correctly identify possible discrepancies in the systems but to furthermore be able to understand and obviate possible discriminatory biases against certain minority groups based on the characteristics associated with them(Yang, Wu 2021;Uwaoma 2024).

Finally, I want to briefly address the performance of the Support Vector Machines (SVM) model. While SVM did not produce the best results in any metric, when trained on unbalanced data, we ought to cut it some slack since we only used a subset of 10,000 observations for it's training and testing, something which is contrary to the other two models where the entire data set was used. This was done due to the high computational expense going hand in hand with the use of SVM and hence also the sheer tremendous amount of time it takes for the model to be trained, tested and subsequently, in a real world scenario, applied. This immense time consumption required for the implementation of this algorithm is the reason why even if the model were to produce better results than any competitors

it would be discredited and rendered useless in any real world application. According to Dal Pozzolo, Caelen, et al.(2014), the timeliness of the decision making process is of great significance in this domain, due to the fact that the slower the reaction time of the model the higher the cost associated with fraud as more and more fraud is made possible since the credit card used to commit those fraudulent transactions is slow to be blocked. Hence it becomes evident that SVM, due to its latency in classifications, may add more cost to the process than it actually would save.

Chapter 6

Conclusions

6.1 Conclusions

The changes of the financial landscape brought forth by the rapid advancements in technology, experienced over the last decades have forever changed not only the way we financial institutions and systems operate, but also the way simple day to day financial acts, such as payment transactions and procurement processes, function. These changes, whilst having brought a great gain in efficiency and overall financial growth, have not come without negative side effects. One of those is the rising trend in credit card fraud, as the exacerbation of credit card payments, as well as the growing trend in digitization experienced due to online usage of credit cards resulting from the boom in e-commerce, have lead to a plethora of new opportunities for fraudulent individuals to exploit the weakened security systems in place, the gaps of which have been made evident not only by overwhelming number of credit card transactions but also by the ever evolving landscape of fraud itself. This has lead to an growing trend in research seeking improved and innovative ways to deal with this issue. The purpose of this study is to examine the performance of three supervised machine learning models namely, random forest, logistic regression and support vector machines, in order to asses their effectiveness in correctly classifying a credit card transaction as fraudulent.

The data set leveraged for the purpose of this is a real life anonymized data set containing information about credit card transactions that occurred over the course of three days, by European card holders, in September of 2013. We trained the algorithms on two different training sets, one was made up of ten distinct balanced subsets of the original data, and the other was an 80% cut off from the original unbalanced data. The results exhibited that both the models trained on the balanced subsets as well as those trained on the raw unbalanced data were satisfactory in terms of accuracy and ROC-AUC scores, the former had a severe lacking in precision something which is of great concern in the fraud detection domain. For this it was concluded that our resampling method, while effective to the degree of it balancing the data and still yielding satisfactory predictive accuracy, negatively impacted the predictive ability of the models when applied to real life data as they were way too sensitive to predicting the positive-fraud class. Moving on, when examining the results of the models trained o the unbalanced training set we can observe that while random forest seems to be the best performer in certain metrics, it is out-shined by logistic regression in our most important metric, the ROC-AUC score, whilst also maintaining satisfactory levels across all other metrics. Due to this and more importantly the fact that logistic regression also brings to the table an unmatched level of transparency as well as it not needing any assumptions about the prior distribution of the data, something particularly useful in dealing with unbalanced data, which is more often than not the case in fraud detection, we conclude that logistic regression seems to be the most fitting model for credit card fraud prediction, in this data set, when also taking into consideration the real-life business context of things.

6.2 Limitations and Future research

One of the main issue we, along with most other studies on the subject, faced was the public unavailability of data. While our data set is a real life data set stemming

from actual credit card transactions the main features are the results of a PCA transformation and no further information is provided on any of them. This lack of precise information limits our ability to interpret/understand the exact factors that are usually associated with fraudulent credit card transactions. Moreover our resampling approach, with the purpose of balancing the training data, proved to be inefficient since the results yielding from the model trained on this data were lacking in precision. Hence future studies are encouraged to experiment with different and more complex resampling techniques such as SMOTE in order to overcome the issue of imbalance in the data and subsequently produce better more reliable results.

References

1. Abdallah, Aisha, Mohd Aizaini Maarof, and Anazida Zainal (June 2016). "Fraud detection system: A survey". en. In: Journal of Network and Computer Applications 68, pp. 90– 113. issn: 10848045. doi: 10.1016/j.jnca.2016.04.007.

2. ACFE (2007), The 2007 Fraud Examiners Manual, Association of Certified Fraud Examiners – ACFE, Austin, TX

3. Ashtiani, M.N. and Raahemi, B., 2021. Intelligent fraud detection in financial statements using machine learning and data mining: a systematic literature review. Ieee Access, 10, pp.72504-72525.

4. Amin, Ahmad et al. (2023). "Tree-based Machine Learning and Deep Learning in Predicting Investor Intention to Public Private Partnership". en. In: International Journal of Advanced Computer Science and Applications 14.1. doi: 10.14569/ijacsa.2023.0140121.

5. Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015.

6. Ayorinde, K., 2021. A methodology for detecting credit card fraud. Minnesota State University, Mankato.

7. Awoyemi, J.O., Adetunmbi, A.O. and Oluwadare, S.A., 2017, October. Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 international conference on computing networking and informatics (ICCNI) (pp. 1-9). IEEE.

8. Azhan, M. and Meraj, S., 2020, December. Credit card fraud detection using machine learning and deep learning techniques. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (pp. 514-518). IEEE.

9. Barker, K.J., D'amato, J. and Sheridon, P., 2008. Credit card fraud: awareness and prevention. Journal of financial crime, 15(4), pp.398-410.

10. Bertrand Lebichot, Yann-Aël Le Borgne, Liyun He, Frederic Oblé, Gianluca Bontempi Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection, INNSBDDL 2019: Recent Advances in Big Data and Deep Learning, pp 78-88, 2019

11. Bertrand Lebichot, Gianmarco Paldino, Wissam Siblini, Liyun He, Frederic Oblé, Gianluca Bontempi Incremental learning strategies for credit cards fraud detection, IInternational Journal of Data Science and Analytics.

12. Bhatla, T.P., Prabhu, V. and Dua, A., 2003. Understanding credit card frauds. Cards business review, 1(6), pp.1-15.

13. Carcillo, Fabrizio; Dal Pozzolo, Andrea; Le Borgne, Yann-Aël; Caelen, Olivier; Mazzer, Yannis; Bontempi, Gianluca. Scarff: a scalable framework for streaming credit card fraud detection with Spark, Information fusion,41, 182-194,2018,Elsevier

14. Carcillo, Fabrizio; Le Borgne, Yann-Aël; Caelen, Olivier; Bontempi, Gianluca. Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization, International Journal of Data Science and Analytics, 5,4,285-300,2018,Springer International Publishing

15. Dai, Y., Yan, J., Tang, X., Zhao, H. and Guo, M., 2016, August. Online credit card fraud detection: a hybrid framework with big data technologies. In 2016 IEEE Trustcom/BigDataSE/ISPA (pp. 1644-1651). IEEE.

16. Dal Pozzolo, Andrea Adaptive Machine learning for credit card fraud detection ULB MLG PhD thesis (supervised by G. Bontempi)

17. Dal Pozzolo, Andrea; Boracchi, Giacomo; Caelen, Olivier; Alippi, Cesare; Bontempi, Gianluca. Credit card fraud detection: a realistic modeling and

a novel learning strategy, IEEE transactions on neural networks and learning systems,29,8,3784-3797,2018,IEEE

18. Dal Pozzolo, Andrea; Caelen, Olivier; Le Borgne, Yann-Ael; Waterschoot, Serge; Bontempi, Gianluca. Learned lessons in credit card fraud detection from a practitioner perspective, Expert systems with applications,41,10,4915-4928,2014, Pergamon

19. European Central Bank. (2021). Annual report 2021. link

20. Federal Trade Commission. (2023). Consumer Sentinel Network data book 2023. link

21. Fabrizio Carcillo, Yann-Aël Le Borgne, Olivier Caelen, Frederic Oblé, Gianluca Bontempi Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection Information Sciences, 2019

22. Ileberi, E., Sun, Y. and Wang, Z., 2022. A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(1), p.24.

23. Kemp, Steven and Nieves Erades P´erez (2023). "Consumer Fraud against Older Adults in Digital Society: Examining Victimization and Its Impact". en. In: International Journal of Environmental Research and Public Health 20.7. doi: 10.3390/ijerph20075404.

24. Liu, X.Y., Wu, J. and Zhou, Z.H., 2008. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2), pp.539-550.

25. Maniraj, S.P., Saini, A., Ahmed, S. and Sarkar, S., 2019. Credit card fraud detection using machine learning and data science. International Journal of Engineering Research, 8(9), pp.110-115.

26. Mohari, A., Dowerah, J., Das, K., Koucher, F. and Bora, D.J., 2020. A comparative study on classification algorithms for credit card fraud detection. International Journal of Modern Research in Engineering and Technology, 2, p.12.

27. More, R., Awati, C., Shirgave, S., Deshmukh, R. and Patil, S., 2021. Credit card fraud detection using supervised learning approach. International journal of scientific & technology research, 9(10), pp.216-219.

28. Melnychenko, S., Volosovych, S., Baraniuk, Y., 2020. Dominant ideas of financial technologies in digital banking. Balt. J. Econ. Stud. 6 (1), 92–99.

29. S. Nami and M. Shajari, "Cost-sensitive payment card fraud detection based on dynamic random forest and k -nearest neighbors," Expert Systems with Applications, vol. 110, pp. 381–392, doi: 10.1016/j.eswa.2018.06.011, Nov. 2018.

30. Niklas Donges. (2021). A complete guide to the Random Forest algorithm. https://builtin.com/data-science/random-forest-algorithm

31. Nugroho, Antonius Sony Eko and Mohammad Hamsal (Oct. 2021). "Research Trend of Digital Innovation in Banking: A Bibliometric Analysis". en. In: Journal of Governance Risk Management Compliance and Sustainability 1.2. Number: 2, pp. 61–73. issn: 2776-9658. doi: 10.31098/jgrcs.v1i2.720.

32. Parmar, J., Patel, A.C. and Savsani, M., 2020. Credit card fraud detection framework-a machine learning perspective. International Journal of Scientific Research in Science and Technology, 7(6), pp.431-435.

33. Randhawa, K., Loo, C.K., Seera, M., Lim, C.P. and Nandi, A.K., 2018. Credit card fraud detection using AdaBoost and majority voting. IEEE access, 6, pp.14277-14284.

34. Sadineni, P.K., 2020, October. Detection of fraudulent transactions in credit card using machine learning algorithms. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 659-660). IEEE.

35. Sarang Narkhede. (2018). Understanding AUC – ROC Curve. link

36. Schaffer, P., 2018. Reducing the Impact of New Account and Credit Card Fraud on Financial Institutions. CPO Magazine, 1 June. Available at: link [Accessed 25 July 2024].

37. Sisodia, Dilip Singh, Nerella Keerthana Reddy, and Shivangi Bhandari (Sept. 2017). "Performance evaluation of class balancing techniques for credit card fraud detection". In: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 2747–2752. doi: 10.1109/ICPCSI.2017.8392219.

38. Shannon, E. (2000), "A new credit-card scam", Time Europe, July 10, p. 42.

39. Shirgave, S., Awati, C., More, R. and Patil, S., 2019. A review on credit card fraud detection using machine learning. International Journal of Scientific & technology research, 8(10), pp.1217-1220.

40. Spathis, Charalambos T. (2002). "Detecting false financial statements using published data: some evidence from Greece". In: Managerial Auditing Journal 17.4. Publisher: MCB UP Ltd, pp. 179–191.

41. Uwaoma, C., 2024. Detecting Bank Account Opening Fraud Using Machine Learning (Doctoral dissertation, Dublin Business School).

42. Warghade, S., Desai, S. and Patil, V., 2020. Credit card fraud detection from imbalanced dataset using machine learning algorithm. International Journal of Computer Trends and Technology, 68(3), pp.22-28.

43. Yann-Aël Le Borgne, Gianluca Bontempi Reproducible machine Learning for Credit Card Fraud Detection - Practical Handbook

44. Y. Yang and M. Wu, "Explainable Machine Learning for Improving Logistic Regression Models," 2021 IEEE 19th International Conference on Industrial Informatics (INDIN), Palma de Mallorca, Spain, 2021, pp. 1-6, doi: 10.1109/INDIN45523.2021.9557392.

45. Yu, Haifei and Xinyu He (2021). "Corporate Data Sharing, Leakage, and Supervision Mechanism Research". en. In: Sustainability 13.2. doi: 10.3390/su13020931.

46. Žigien˙e, Gerda, Egidijus Rybakovas, and Robertas Alzbutas (2019). "Artificial Intelligence Based Commercial Risk Management Framework for SMEs". en. In: Sustainability 11.16. doi: 10.3390/su11164501.

Appendix A

Appendix content

Table A.1. *Outlier Summary*

Feature	Total Outliers	Fraudulent Outliers	Non-Fraudulent Outliers
V1	5281	135	5146
V2	8288	205	8083
V3	2462	252	2210
V4	10025	263	9762
V5	10099	155	9944
V6	20202	119	20083
V7	5141	247	4894
V8	21042	190	20852
V9	8232	190	8042
V10	8434	318	8116
V11	656	241	415
V12	14356	324	14032
V13	2953	8	2945
V14	13068	344	12724
V15	2614	13	2601
V16	6957	279	6678
V17	6809	312	6497
V18	7078	190	6888
V19	9320	112	9208
V20	17990	159	17831
V21	11263	192	11071
V22	904	19	885
V23	10816	106	10710
V24	3942	2	3940
V25	4237	50	4187
V26	6118	8	6110
V27	33790	275	33515
V28	26188	217	25971

Table A.2. *Feature Weights Comparison, of logistic regresion, Between Subset-Aggregated and Raw Data Models*

Feature	Weight (Subsets)	Weight (Raw Data)
V4	1.922160	0.806524
V11	0.938827	0.053893
V22	0.514619	0.525896
V5	0.381178	0.050761
V2	0.288545	0.008057
V28	0.234604	-0.103935
V21	0.214213	0.197132
V27	0.170264	-0.260086
V25	0.165268	0.072415
V24	0.057832	0.077358
V16	0.049012	-0.198870
V19	-0.010628	-0.030424
Amount	-0.054422	-0.155431
V26	-0.066064	0.143168
V18	-0.066860	-0.027790
V1	-0.071288	0.092881
V17	-0.154022	0.033789
V13	-0.161936	-0.233601
V6	-0.189766	0.022204
Time	-0.277668	-0.005488
V23	-0.288680	-0.086811
V15	-0.331363	-0.191074
V9	-0.361824	-0.249339
V3	-0.430206	0.025494
V7	-0.452622	-0.226635
V20	-0.769014	-0.217462
V10	-0.846437	-0.393232
V8	-1.211123	-0.215473
V12	-1.339773	0.020582
V14	-2.827750	-0.713165

Note: The coefficients for the subsets are the average coefficients produced by the distinct models trained on each subset.