International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 01 | January - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Detection of Fraud Insurance Claims in Vehicles

Rohith Reddy, Golla Prem Kumar, Goddati Jagadish, Susmitha Dudipalli, Adapa Venkata Prabhas, Kallam Pavan Teja Reddy

Insurance fraud has been a problem since the inception of insurance. However, the methods used for committing fraud and the frequency of these incidents have increased in recent years. Vehicle insurance fraud often involves making false or exaggerated claims for damages or injuries resulting from an accident. Examples of this type of fraud include staged accidents, the use of phantom passengers, and false personal injury claims. In this paper, we analyze data to understand the characteristics of fraudulent claims and use machine learning algorithms to detect this type of fraud.

Additional Key Words and Phrases: Random Forest, Decision Trees, Exploratory data analysis, Fraud detection

1 INTRODUCTION

Insurance fraud is a pervasive problem that has been affecting the insurance industry for many years. One of the most common types of insurance fraud is vehicle insurance fraud, which involves making false or exaggerated claims for damages or injuries resulting from a car accident. In recent years, the volume and frequency of vehicle insurance fraud incidents have increased significantly, leading to significant losses for insurance companies.

The purpose of this project is to create a model using machine learning algorithms to detect vehicle insurance fraud. One challenge in using machine learning for fraud detection is that fraud is much less common than legitimate insurance claims, which can make it difficult for the model to accurately identify fraudulent activity. In order to develop a successful model, it is important to balance the cost of false alerts with the potential savings from avoiding losses due to fraud. Insurance fraud can take many forms, including arranging accidents, misrepresenting the circumstances of an accident, and exaggerating the extent of damages or injuries. Machine learning can help improve the accuracy of fraud detection and allow insurance companies to more effectively identify and prevent fraudulent activity.

2 METHODOLOGY

The first step in our project was to collect a large dataset of past insurance claims, both fraudulent and legitimate. We obtained this dataset from Kaggle, which provided us with anonymized data on a variety of claims made over a period of several years. The dataset included information on the type of claim, the amount of the claim, the date of the claim, and other relevant details.

Once we had collected the dataset, we performed basic data analysis to understand the characteristics of fraudulent claims. This analysis allowed us to identify key features that are often associated with fraudulent claims, such as the amount of the claim, the type of claim, and the date of the claim. We also looked at other factors, such as the location of the accident and the number of people

involved, to see if they had any impact on the likelihood of fraud.

With this information in hand, we proceeded to train a machine-learning model to detect fraudulent claims. We used a variety of algorithms, including logistic regression, decision trees, and random forests, to develop the model. We trained the model on the dataset of past claims, using the identified features as inputs and the known fraudulent and legitimate labels as outputs.

Once the model was trained, we tested it on a separate dataset of claims to see how well it performed. We found that the model was able to accurately detect fraudulent claims with a high degree of accuracy

achieving an overall accuracy rate of over 94 *percent*.

Our work can be divided into four main components:

· Exploratory Data Analysis: This involves examining the data to understand its characteristics and identify any patterns or trends.

· Data Preprocessing: This involves cleaning and preparing the data for modeling, such as by handling missing values, transforming variables, and scaling the data.

· Data Modeling: This involves building and fitting statistical or machine learning models to the data to make predictions or classify data points.

· Model Evaluation: This involves assessing the performance of the model using metrics such as accuracy, precision, and recall, and making adjustments to improve the model as needed

2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial step in any data science project, including the project on detecting vehicle fraud insurance claims. EDA involves analyzing and summarizing the characteristics of the data, identifying any trends or patterns, and checking for inconsistencies or anomalies.

The goal of EDA is to gain a better understanding of the data and to identify any potential problems or opportunities that could affect the success of the project. This involves examining the distribution of the data, looking for correlations between variables, and visualizing the data using charts and plots.

Our first goal was to get familiar with the dataset, We found that the data has 33 columns including our dependant column *'FraudFound'*. Our data consists of a total of 9 numerical and 24 categorical columns with no missing values.

Some important plots and pairwise comparisons between our dependent and independent variables.

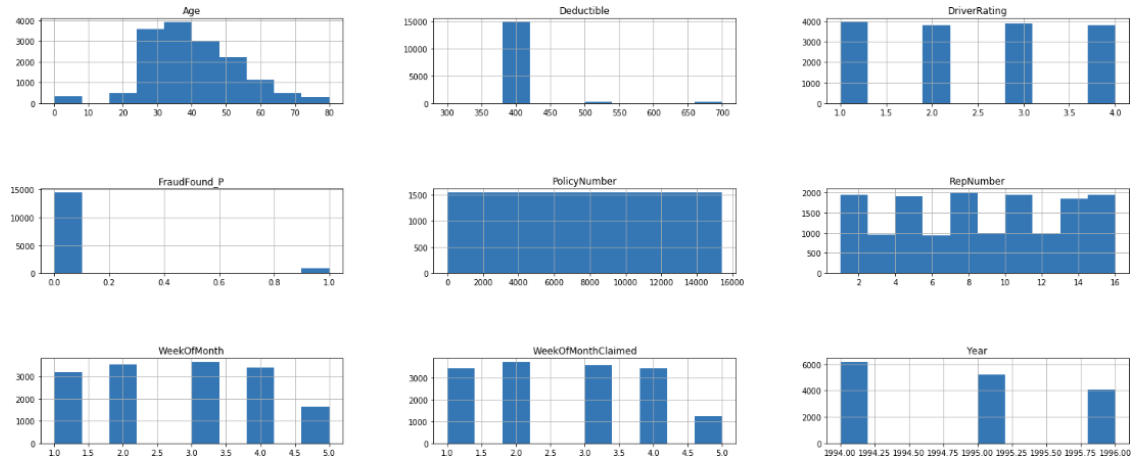


Fig. 1. Distribution of Variables

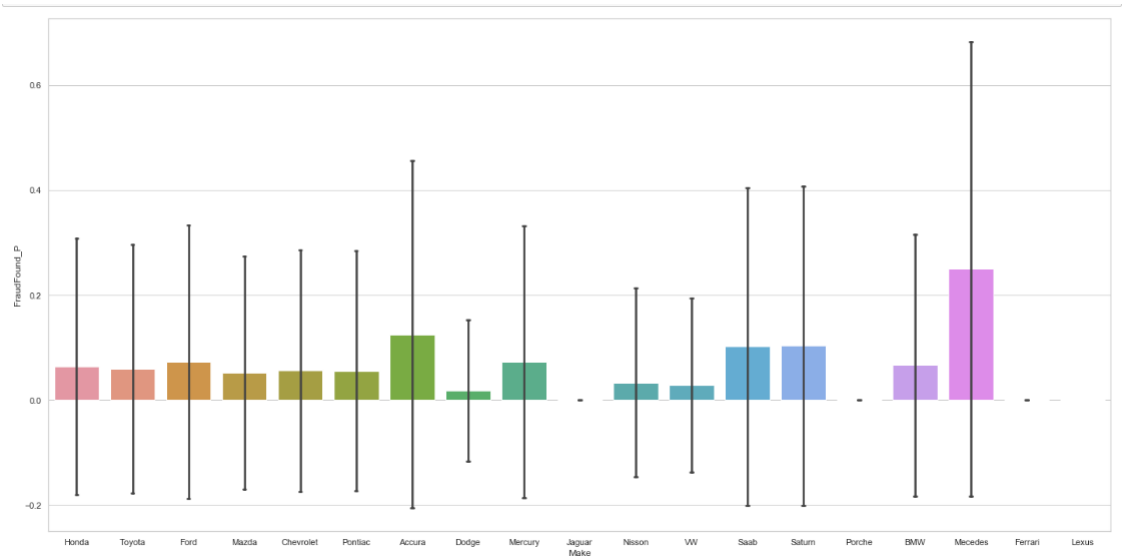


Fig. 2. FraudFound vs Make

Analysis: Mercedes and Accura have a higher probability of fraudulent transactions, most likely due to a higherreturn in these costlier cars

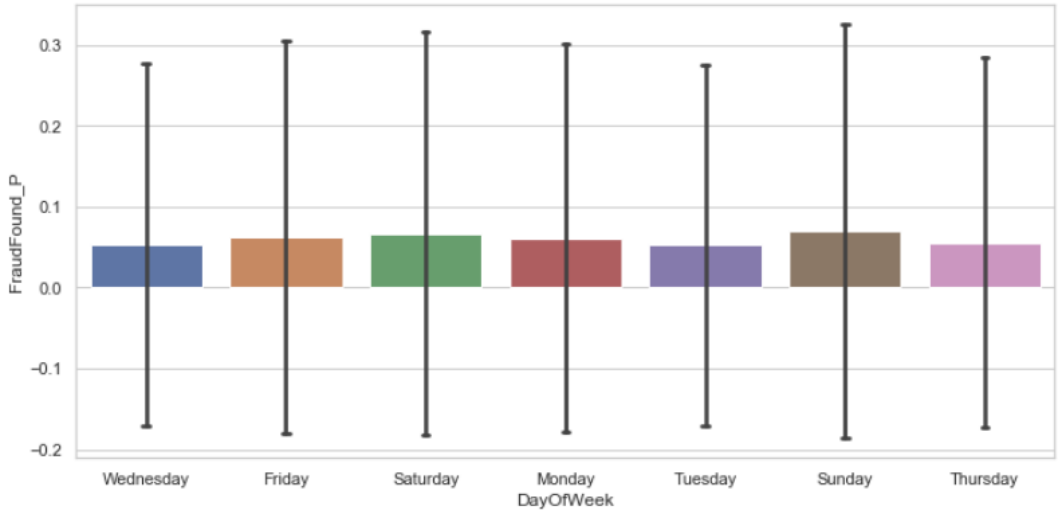


Fig. 3. FraudFound vs DayOf Week

Analysis: Fraudulent claims are higher nearer to the Weekends!

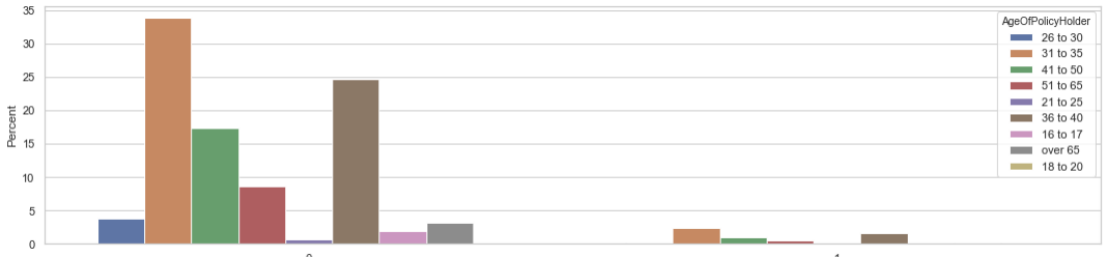


Fig. 4. FraudFound vs AgeOfPolicyHolder

Analysis: Fraudulent claims are generally made from persons ranging from the age group 30-40

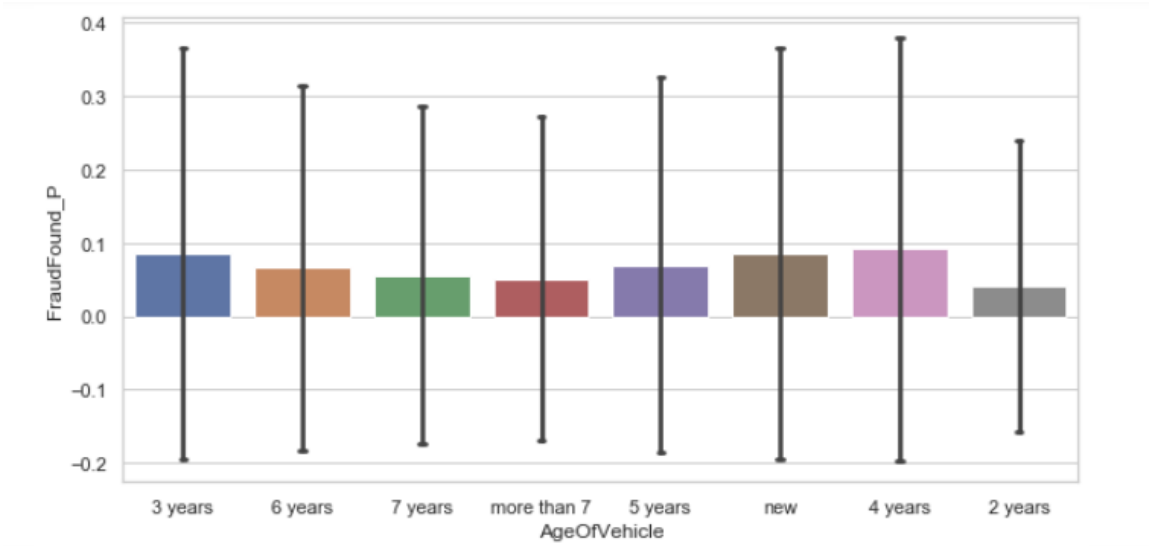


Fig. 5. FraudFound vs AgeOf Vehicle

Analysis: Newer Vehicles and Ages of vehicles between 2-4 years have encountered many Fraudulent claims

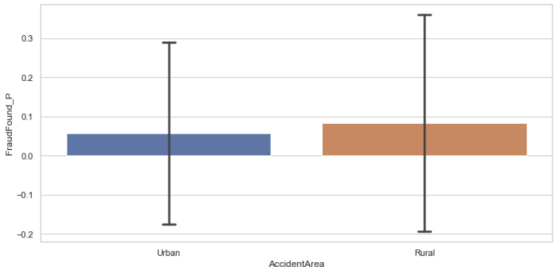


Fig. 6. FraudFound vs AccidentArea

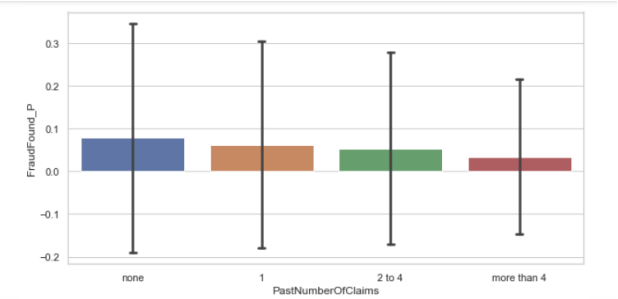


Fig. 7. FraudFound vs PastClaims

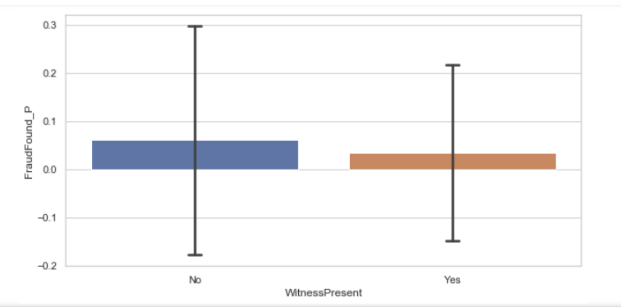


Fig. 8. FraudFound vs WitnessPresent

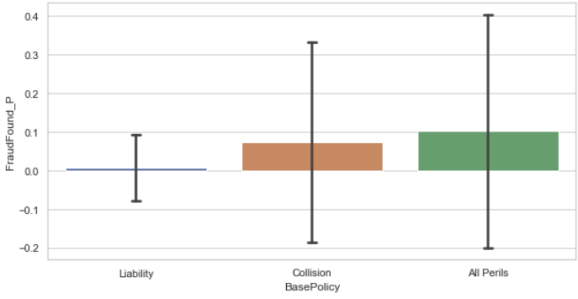


Fig. 9. FraudFound vs TypeOfPolicy

These are the other interesting analysis we found from *EDA*

3 DATA PREPROCESSING

3.1 Preprocessing

After having checked that no missing values nor duplicates were present in the dataset we started with the preprocessing. We first removed two columns – *PolicyNumber* and *Year* – as they would be useless in the modeling part: *PolicyNumber*, in fact, is just an ID column; on the other hand, Year has just three values (1994, 1995, 1996) so it will not be significant in a model. Moreover, we did not want our model to be *historical based*, as – theoretically – it would serve for new observations without considering the year of them. Afterwards, we divided the data in train and validation (75% and 25%) and fitted a preprocessing pipeline that encodes the categorical features and standardizes the numerical ones.

3.2 Variables Importance

Before starting with the actual modeling and prediction part, we decided to train two models to infer which variables might be more important, in particular a Classification Tree and a Random Forest.

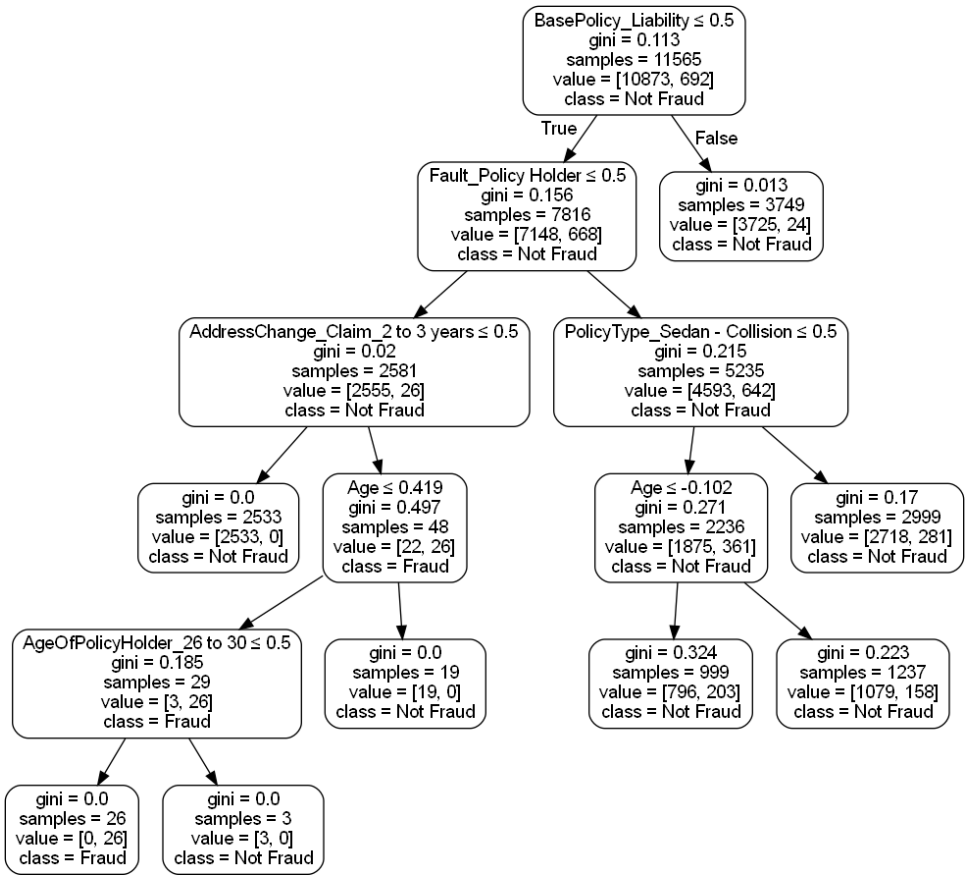


Fig. 10. Regression Tree

From the Regression Tree, which was trained with pruning in order to avoid some complexity and make it more interpretable and readable, the most important features seem to be BasePolicy, Fault, *PolicyType, AddressChange_Claim,Age and AgeOfPolicyHolder*.

On the other hand, using Random Forest we calculated two features' importance, one impurity-based and one usingpermutation importance based on the decrease in accuracy.

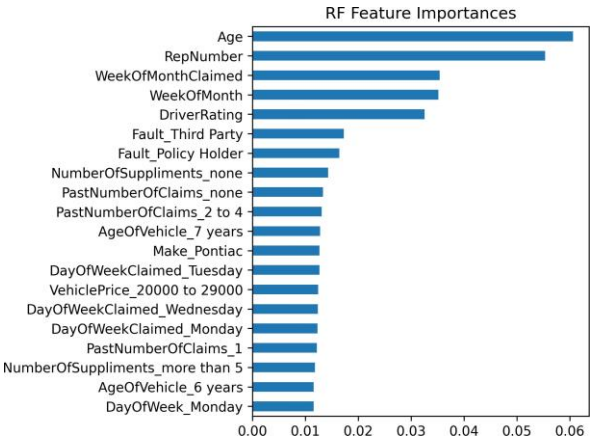


Fig. 11. Random Forest Feature Importance

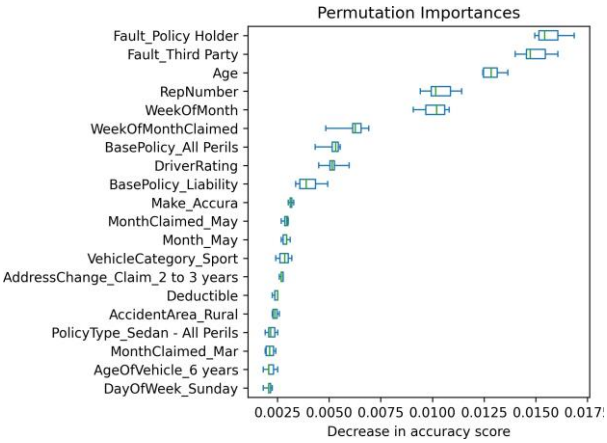


Fig. 12. Permutation Importance

The results obtained differ between them but have some similarities: we can deduce that the features with higher importance are *Age, Fault, RepNumber, WeekOfMonth, WeekOfMonthClaimed, DriverRating, and BasePolicy* ; some ofthem are the same identified in the Regression Tree before.

To conclude this part, we repeated the preprocessing using just a part of the features *(Age, Fault, Make, PolicyType, BasePolicy, AddressChange_Claim, MonthClaimed, RepNumber, AgeOfPolicyHolder, Days_Policy_Claim, NumberOfCars,DriverRating, Deductible, WeekOfMonth, WeekOfMonthClaimed)*, in the modeling section we will use both sets to see ifany improvement – or not – could be achieved using a subset of most important features.

4 MODELING

For the modeling, we chose four different models – Logistic Regression, SVM, RandomForest, and XGBoost – tuned and trained on the full set and on the one with selected features. The tuning of hyperparameters – for SVM, RandomForest and XGBoost – was performed using a Bayesian Optimizer, which speeds up the tuning process with respect to a GridSearch and it is not solely based on the randomness of parameters' choice as RandomSearch; a classical Bayesian Optimizer, in fact, combines randomness and posterior probability distribution in searching the optimal parameters by approximating the target function through gaussian process – random samples are drawn iteratively and the function outputs between the samples are approximated by a confidence region; new samples are then drawn from the parameter space at the high mean and variance over the confidence region. Afterward, the models were tested on the validation set, calculating different performance measures (accuracy, recall, AUC,

etc.). Moreover, we conclude the modeling partwith some threshold optimization. Usually, the threshold used for classification tasks is 0.5; in this case, as the class label was very unbalanced towards *Not fraud* observation, we tried to optimize the threshold following two different metrics, F1-score and Youden's Index. The idea behind this was to sacrifice a bit of overall accuracy to correctly predict more fraud claims, as, in a real case scenario, it would be more important to identify these types of observations

4.1 Logistic Regression

For the logistic regression with full dataset, we obtained the following confusion matrix and ROC curve, with an accuracy of 0.938, recall 0.013, F1 0.025, and Youden's Index 0.012. The result present were obtained with the default threshold of 0.5. In order to deduce the validity of the model, it is better to consider the AUC and not the accuracy, as itmight be misleading due to the unbalancedness of the class label.

· *Full Dataset*

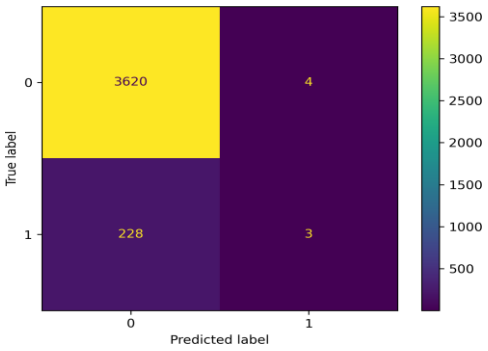


Fig. 13. Confusion Matrix for Full dataset

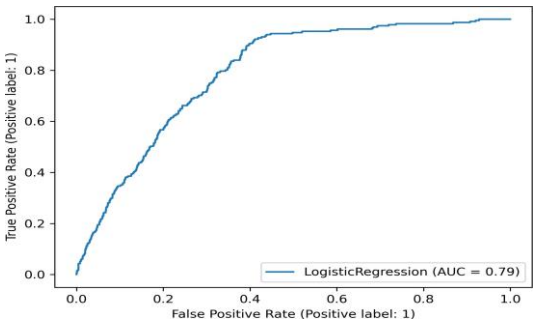


Fig. 14. ROC Curve for Full Dataset

In the threshold optimization part, we first optimize with respect to the Youden's Index and then F1-score. For all models, we will first present the confusion matrix corresponding to Youden's Index optimization and then the one withrespect to F1-score.

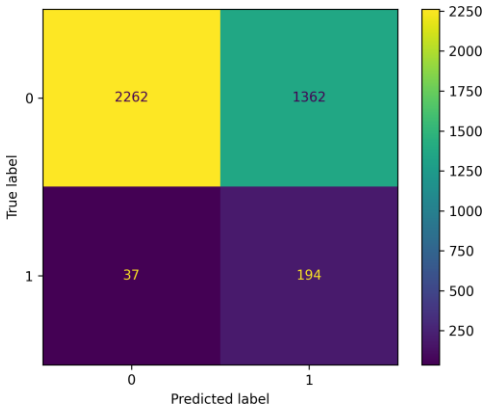


Fig. 15. Youden Optimized

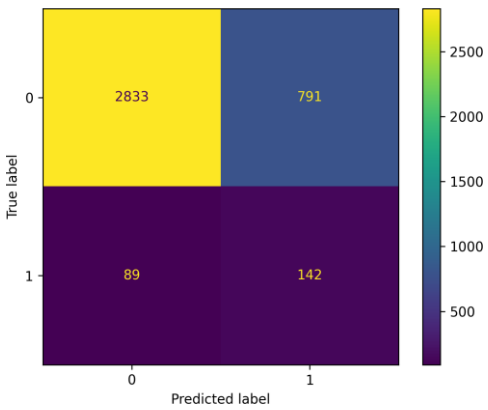


Fig. 16. F1 score optimized

In Fig.15, with 0.05 as the threshold, we obtained a Youden's Index of 0.464, an accuracy of 0.637, recall of 0.839, and F1-score 0.217. On the other hand in Fig.16, with a threshold of 0.1, we got Youden's Index 0.396, Accuracy 0.772, recall0.615, F1-score 0.244.

· *Selected Dataset*

With selected features, the result obtained is very similar, with both models reaching 0.79 of AUC and similar values inthe other metrics.

Accuracy: 0.939
Recall: 0.008
F1: 0.016
Youden's Index: 0.007

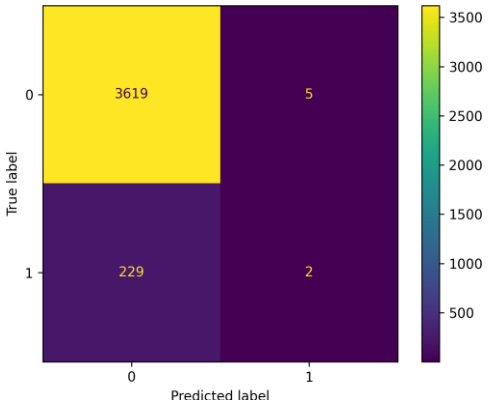


Fig. 17. Confusion Matrix for Selected Dataset

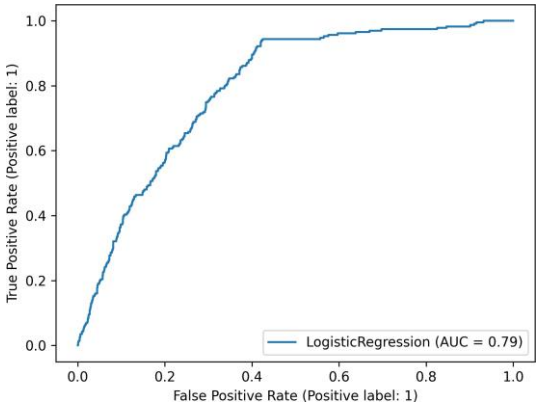
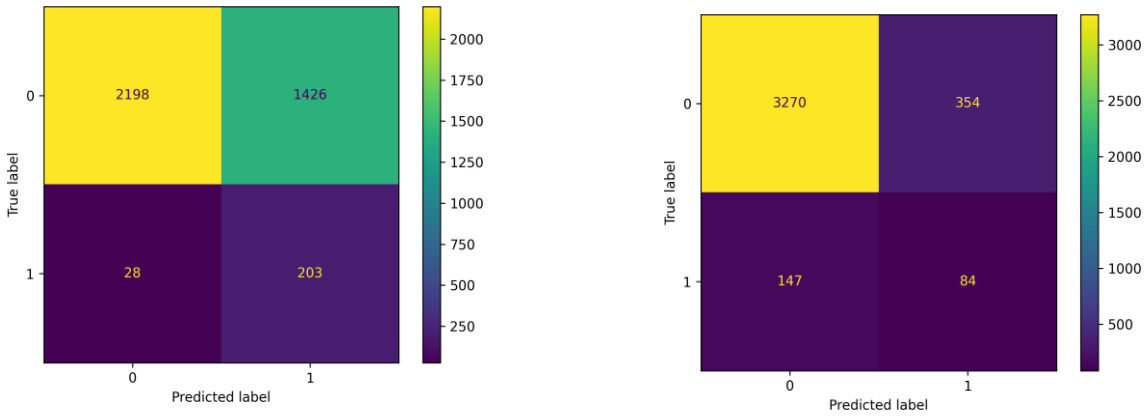


Fig. 18. ROC for Selected Dataset

For threshold optimization, a slightly higher Youden's Index and F1 score was reached.



Threshold: 0.05
Youden's Index: 0.485
Accuracy: 0.622
Recall: 0.879
F1: 0.218

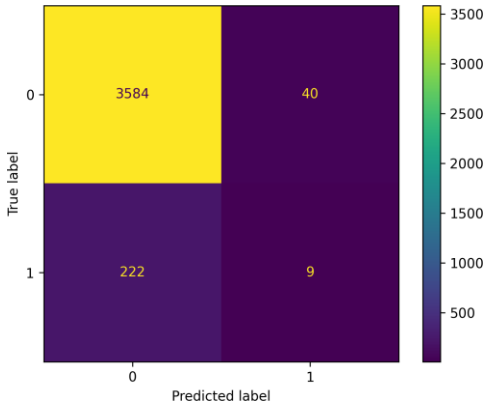
Fig. 19. Youden Optimized for Selected Dataset

Threshold: 0.15
Youden's Index: 0.266
Accuracy: 0.870
Recall: 0.364
F1: 0.251

Fig. 20. F1 score optimized for Selected Dataset

4.2 SVM

After tuning and training the model (the parameters of it and all other models can be found in our Python code), these were the result obtained (with default 0.5 threshold in the confusion matrix) for SVM.



Accuracy: 0.932
Recall: 0.038
F1: 0.064
Youden's Index: 0.027

Fig. 21. Confusion Matrix for Full Dataset

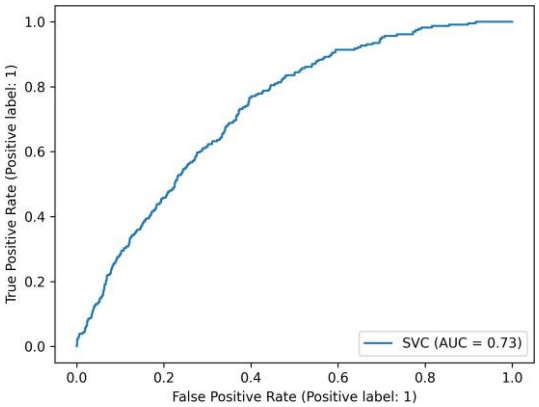
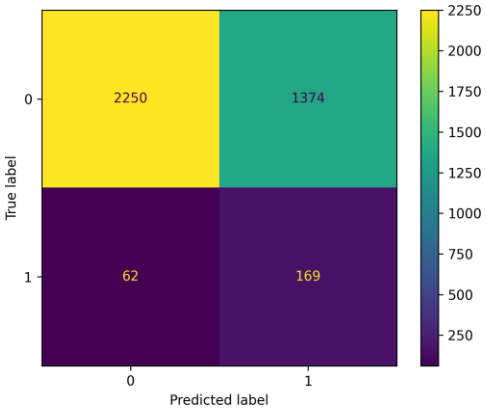
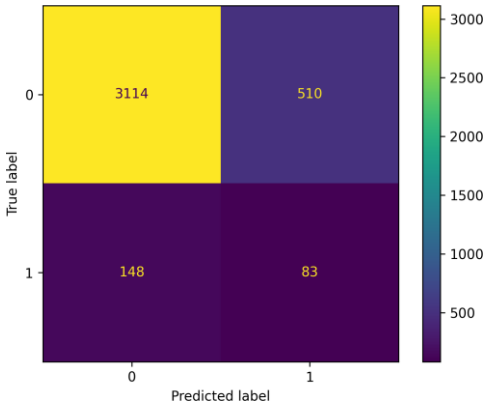


Fig. 22. ROC for Full Dataset





Threshold: 0.05
Youden's Index: 0.352
Accuracy: 0.627
Recall: 0.731
F1: 0.191

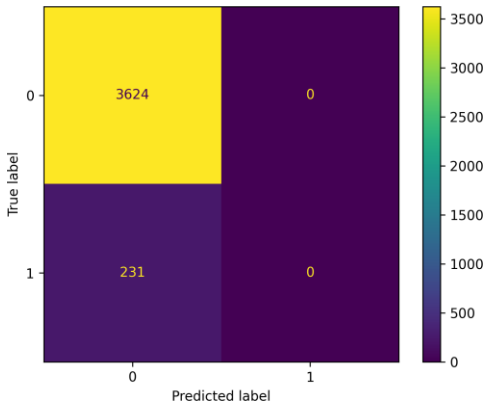
Fig. 23. Youden Optimized for Full Dataset

Threshold: 0.1
Youden's Index: 0.219
Accuracy: 0.829
Recall: 0.359
F1: 0.201

Fig. 24. F1 score optimized for Full Dataset

. Selected Dataset

With selected features only, this time we obtained a substantial increase in AUC – from 0.73 to 0.79 – thus increasing the overall performance of our model regardless the threshold used. Nevertheless, with default threshold, the model classified every observation as *Not fraud*.



Accuracy: 0.940
Recall: 0.0
F1: 0.0
Youden's Index: 0.0

Fig. 25. Confusion Matrix for Selected Dataset

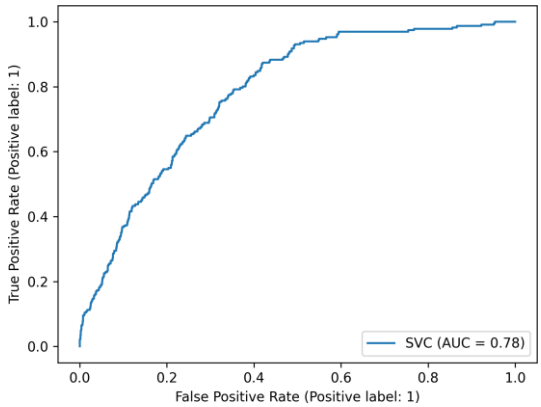
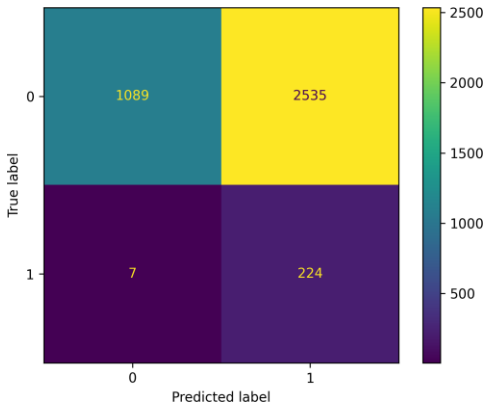
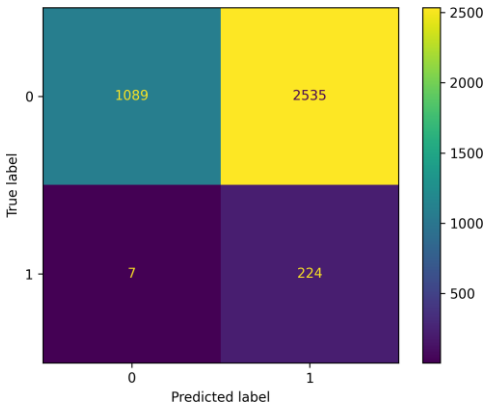


Fig. 26. ROC for Selected Dataset





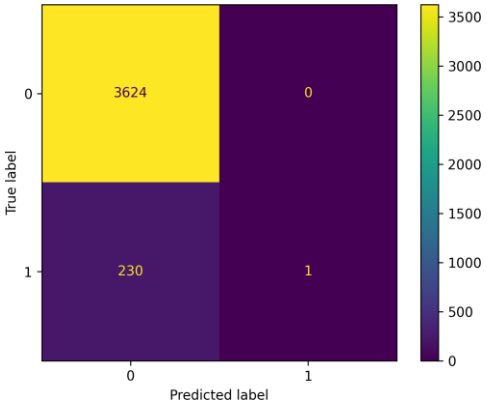
Threshold: 0.05
Youden's Index: 0.270
Accuracy: 0.341
Recall: 0.970
F1: 0.150

Fig. 27. Youden Optimized for Selected Dataset

Threshold: 0.05
Youden's Index: 0.270
Accuracy: 0.341
Recall: 0.970
F1: 0.150

Fig. 28. F1 score optimized for Selected Dataset

4.3 Random Forest



Accuracy: 0.940
Recall: 0.004
F1: 0.008
Youden's Index: 0.004

Fig. 29. Confusion Matrix for Full Dataset

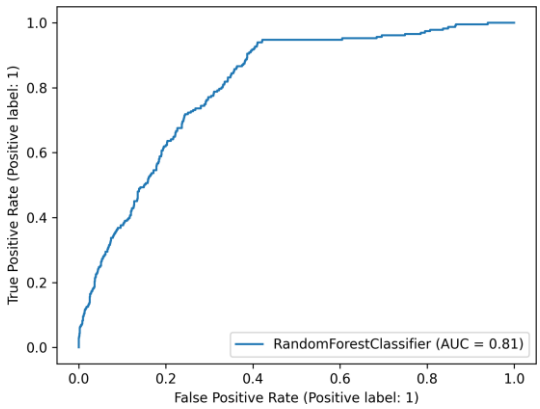
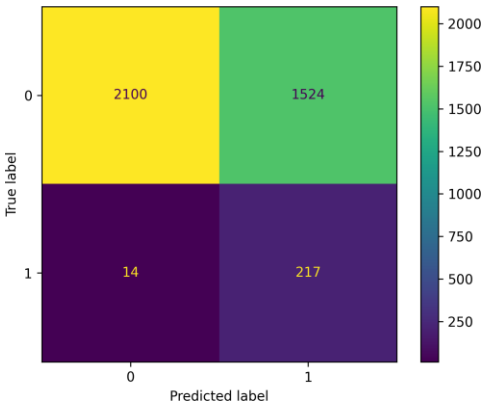
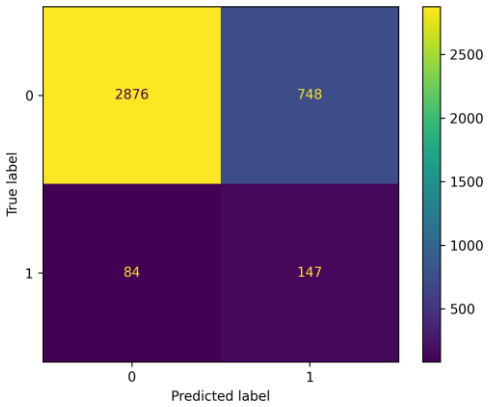


Fig. 30. ROC for Full Dataset





Threshold: 0.05
Youden's Index: 0.519
Accuracy: 0.601
Recall: 0.939
F1: 0.220

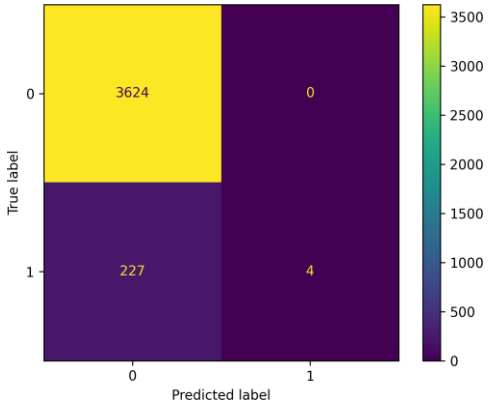
Fig. 31. Youden Optimized for Full Dataset

Threshold: 0.1
Youden's Index: 0.430
Accuracy: 0.784
Recall: 0.636
F1: 0.261

Fig. 32. F1 score optimized for Full Dataset

· Selected Dataset

With selected features only, the results obtained were very similar to the model with full dataset.



Accuracy: 0.941
Recall: 0.017
F1: 0.034
Youden's Index: 0.017

Fig. 33. Confusion Matrix for Selected Dataset

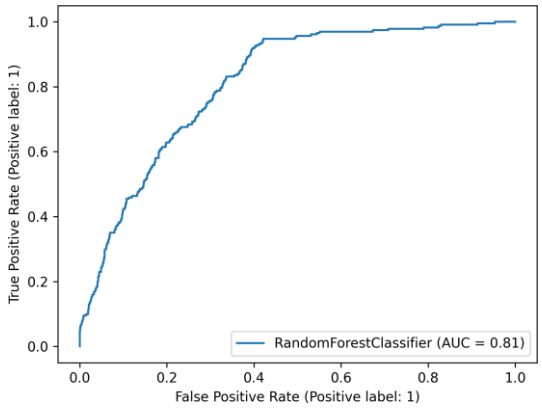
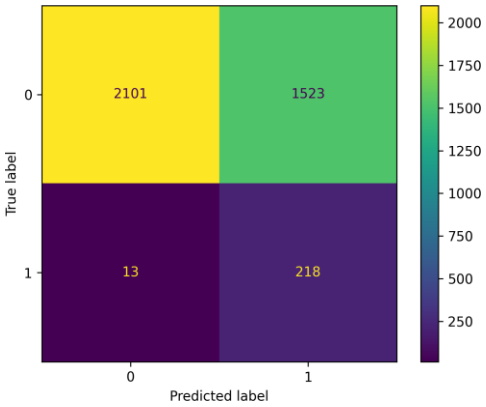
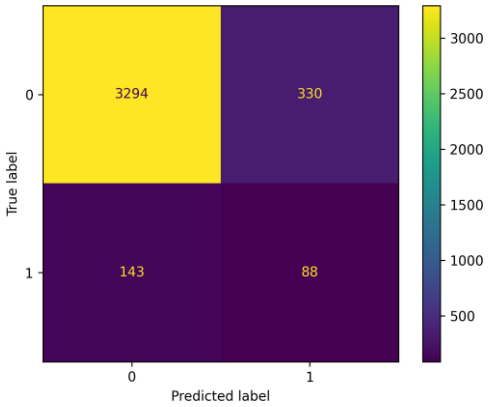


Fig. 34. ROC for Selected Dataset

Almost the same also for thresholds, with 0.05 for Youden's optimum, but 0.15 for F1-score one..





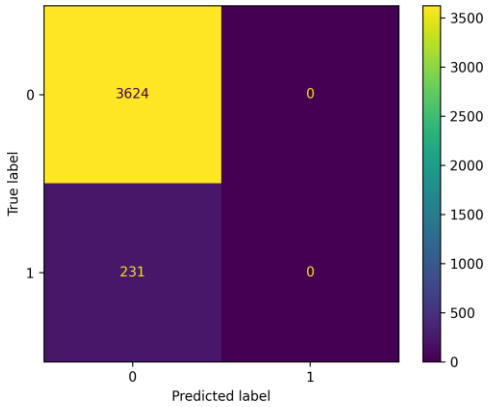
Threshold: 0.05
Youden's Index: 0.523
Accuracy: 0.601
Recall: 0.944
F1: 0.221

Fig. 35. Youden Optimized for Selected Dataset

Threshold: 0.15
Youden's Index: 0.290
Accuracy: 0.877
Recall: 0.381
F1: 0.271

Fig. 36. F1 score optimized for Selected Dataset

4.4 XGBoost



Accuracy: 0.940
Recall: 0.0
F1: 0.0
Youden's Index: 0.0

Fig. 37. Confusion Matrix for Full Dataset

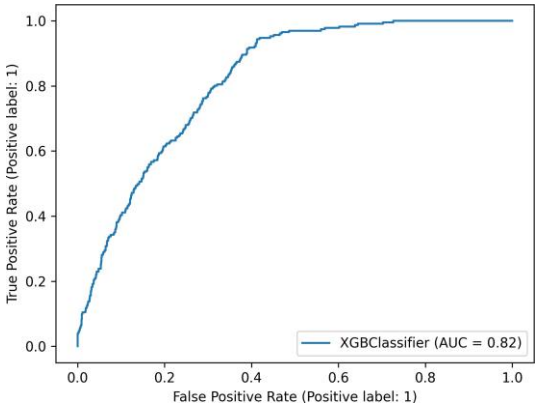
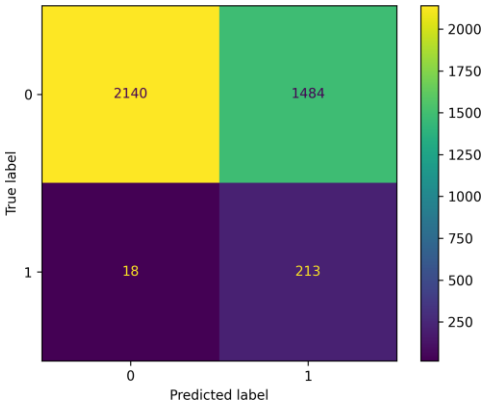
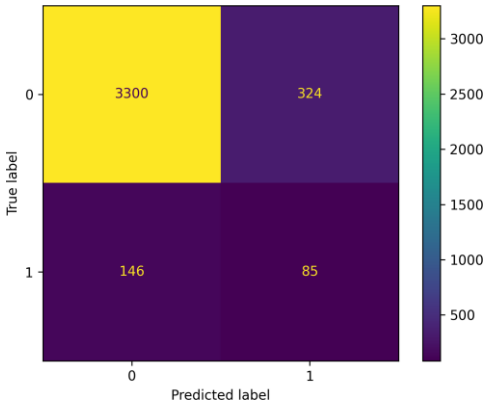


Fig. 38. ROC for Full Dataset

Even if the confusion matrix might be misleading – all observation classified as *Not fraud* again – the AUC of 0.82 obtained by XGBoost was the highest we were able to get.

For threshold optimization, we got similar result to Random Forest, with 0.05 and 0.15 as thresholds.





Threshold: 0.05
Youden's Index: 0.513
Accuracy: 0.610
Recall: 0.922
F1: 0.221

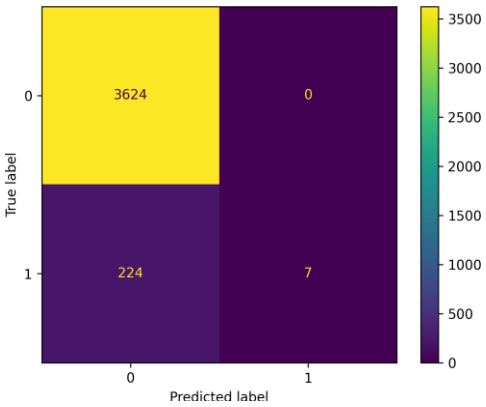
Fig. 39. Youden Optimized for Full Dataset

Threshold: 0.15
Youden's Index: 0.279
Accuracy: 0.878
Recall: 0.368
F1: 0.266

Fig. 40. F1 score optimized for Full Dataset

. Selected Dataset

Again, with selected features, the results are similar or slightly better in some metrics, without – unfortunately – anysignificant increase in AUC.



Accuracy: 0.942
Recall: 0.030
F1: 0.058
Youden's Index: 0.030

Fig. 41. Confusion Matrix for Selected Dataset

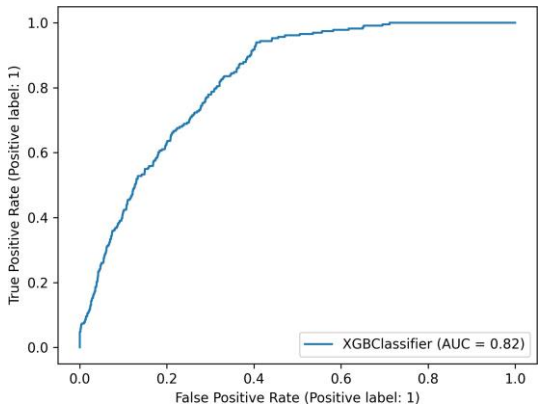
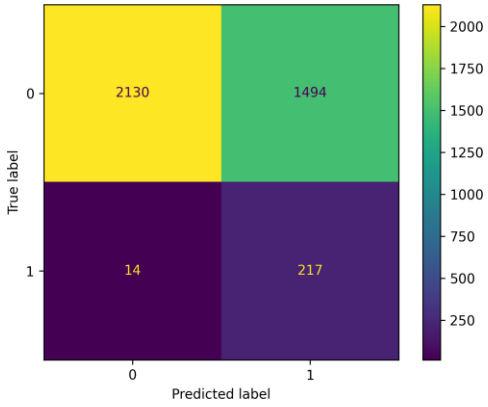
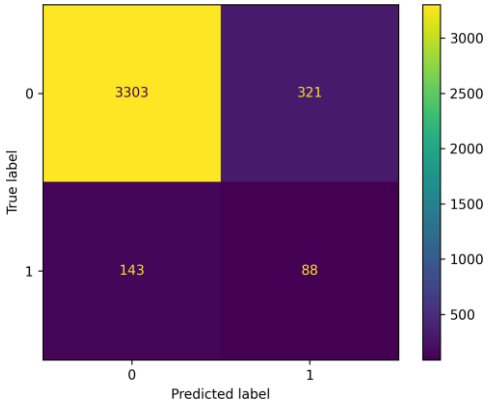


Fig. 42. ROC for Selected Dataset

Also in threshold optimization, the same 0.05 and 0.15 thresholds were chosen.





Threshold: 0.05
Youden's Index: 0.527
Accuracy: 0.609
Recall: 0.939
F1: 0.223

Fig. 43. Youden Optimized for Selected Dataset

Threshold: 0.15
Youden's Index: 0.292
Accuracy: 0.879
Recall: 0.381
F1: 0.275

Fig. 44. F1 score optimized for Selected Dataset

5 CONCLUSION AND FUTURE SCOPE

After reporting all the results and evaluations, we can see that all models – despite a good accuracy score over 0.9 – always struggles to reach a good recall, unless very low classification thresholds are chosen. Nevertheless, this has a drawback, with huge decreases in overall accuracy, especially using SVM with selected features, with the number falsepositives that exceed true negatives' one. As our final model, we would choose the XGBoost using selected features, probably using the threshold the maximize the F1-score; in fact, with even lower thresholds, the model classified too many false positives, obtaining a low accuracy of 0.609. On the other hand, with 0.05 as threshold, the model reaches a recall of 0.939, identifying almost all fraud correctly. At the end, the threshold choice would depend on a real case scenario, where – for example – we might want to get a high recall regardless the overall accuracy.

Apart from the Machine learning models we used in our project, Another approach is the use of deep learning techniques, like convolutional neural networks (CNN), which are effective to use image recognition tasks. For example, some researchers have proposed using CNNs to analyze images of vehicle identification numbers (VINs) to detect fraud by identifying tampered or fake VINs. There are also works on using Graph-based algorithms, which are effective in identifying the relationship between entities. Some researchers proposed using graph-based algorithms to analyze the relationships between vehicles, buyers, and sellers to detect fraudulent networks or patterns of activity.

The research in this area is still ongoing, but initial results have shown that machine learning can be an effective tool for detecting vehicle fraud. However, to have a robust system, it also needs to be combined with other techniques and approaches such as data visualization and domain-specific expertise

In conclusion, our project shows that machine learning algorithms have the potential to play a significant role in the fight against insurance fraud. By providing insurance companies with a powerful tool for detecting fraudulent claims, we can help them reduce their losses and improve the overall efficiency of the insurance industry.

REFERENCES

[1] Faheem Aslam, Ahmed Imran Hunjra, Zied Ftiti, Wael Louhichi, and Tahira Shams. 2022. Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance* 62 (2022), 101744.

[2] Botond Benedek, Cristina Ciumas, and Bálint Zsolt Nagy. 2022. Automobile insurance fraud detection in the age of big data–a systematic and comprehensive literature review. *Journal of Financial Regulation and Compliance* (2022).

[3] Andrea Dal Pozzolo. 2015. Adaptive machine learning for credit card fraud detection. (2015).

[4] Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud. 2019. Extreme gradient boosting machine learning algorithm for safe auto insurance operations. In *2019 IEEE international conference on vehicular electronics and safety (ICVES)*. IEEE, 1–5.

[5] MOHAMED Hanafy and Ruixing Ming. 2021. Using Machine Learning Models to Compare Various Resampling Methods in Predicting Insurance Fraud. *Journal of Theoretical and Applied Information Technology* 99, 12 (2021).

[6] Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng Yao. 2021. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[7] Jesús M Pérez, Javier Muguerza, Olatz Arbelaitz, Ibai Gurrutxaga, and José I Martín. 2005. Consolidated tree classifier learning in a car insurance fraud detection domain with class imbalance. In *International Conference on Pattern Recognition and Image Analysis*. Springer, 381–389.

[8] P Sai Pranavi, HD Sheethal, Sharanya S Kumar, Sonika Kariappa, and BH Swathi. 2020. Analysis of Vehicle Insurance Data to Detect Fraud using Machine Learning. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* 8, 7 (2020), 2033–2038.

[9] Riya Roy and K Thomas George. 2017. Detecting insurance claims fraud using machine learning techniques. In *2017 international conference on circuit, power and computing technologies (ICCPCT)*. IEEE, 1–6.

[10] Yibo Wang and Wei Xu. 2018. Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decision Support Systems* 105 (2018), 87–95.

[11] Meryem Yankol-Schalck. 2022. The value of cross-data set analysis for automobile insurance fraud detection. *Research in International Business and Finance* 63 (2022), 101769.