Optimizing Fraud Detection in Auto Insurance Claims Using Machine Learning

Tova Thorén

EC Utbildning, DS23

Data Science Project

2024-08-19

Abstract

This report explores the development of a machine learning model for detecting fraudulent auto insurance claims, with a focus on addressing class imbalance through oversampling techniques like SMOTE. The study demonstrates that applying SMOTE improves recall, enhancing the model's ability to identify fraud, though it can also lead to a slight decrease in precision. Logistic Regression emerged as the best-performing model, balancing interpretability with strong performance metrics. Despite these improvements, the model's precision remains a challenge due to the introduction of synthetic samples and the small dataset size. The findings highlight the potential of machine learning in fraud detection while acknowledging the limitations of data quality and model generalization.

Table of Contents

1 Introduction

We are all exposed to various risks in life, such as traffic accidents, fires, theft, or prolonged illness. Without insurance, these events could lead to a financial devastation for individuals. Insurance means that the entire collective shares these risks, where each person pays a premium in exchange for the insurance company providing compensation in case of an accident. This system contributes to both the financial security of individuals and more effective risk management within society. Each year, Swedish insurance companies handle more than 3 million claims and pay out 70 billion SEK in compensation to policyholders (Svensk Försäkring & Larmtjänst, 2024).

However, the sustainability of this system is compromised when individuals or criminal networks exploit it for personal gain. To combat this, insurance companies rigorously investigate suspicious claims to detect and prevent fraud. Over the past five years, the number of fraudulent investigations in Sweden has risen by approximately 32%, particularly within home/villa/travel, and auto insurance sectors. Among these, auto insurance claims have the highest proportion of denied claims relative to the number of fraudulent incidents (Larmtjänst, 2024).

In 2023 alone, denied claims in Sweden amounted to a total of 682 million SEK, highlighting the financial impact of fraudulent activities on the industry. Additionally, it is estimated that every honest policyholder contributes approximately 500 SEK annually to cover losses from undetected fraudulent claims. As such, fraudulent activities not only strain insurance company resources but also financially burden honest policyholders through increased premiums (Svensk Försäkring & Larmtjänst, 2024).

Traditional methods of fraud detection often rely on manual reviews and rule-based systems, which can be inefficient, subjective, and prone to oversight. With the advancement of technology more companies and authorities engaged in fraud prevention are adopting machine learning techniques, resulting in improved detection rates (ACFE & SAS Institute, 2024). But a significant portion of fraud cases still remain undetected. A study on fraud prevention by the Association of Certified Fraud Examiners (ACFE) and the AI-company SAS Institute (2024) estimates that undetected fraud accounts for approximately 5-10 percent of all compensation paid out annually from Swedish insurance companies. Therefore, there is much room for improvement in the field of fraud detection by enhancing machine learning methods.

1.1 Problem Statement

The primary goal of this project is to develop a fraud detection model using machine learning, specifically targeting auto insurance claims. A common challenge in fraud detection is the imbalance

within the dataset, where the majority of cases are non-fraudulent. This project aims to address this issue as well. This leads to the following research questions:

1. How can machine learning techniques be utilized to effectively detect fraudulent auto insurance claims?
2. What impact does the application of oversampling techniques have on the performance of fraud detection models?
3. What are the most significant features that distinguish fraudulent claims from legitimate ones?

1.2 Definitions and limitations of study

The dataset used in this project is sourced from Kaggle and is limited to auto insurance claims. While this dataset provides a valuable foundation for exploration, its quality and generalization to real-world scenarios cannot be assured. Because of this, the results and conclusions from this study should be considered with these limitations in mind.

2 Theory

This chapter covers the theoretical foundations related to this project, including supervised machine learning, common algorithms for fraud detection, methods for addressing imbalanced datasets, and the evaluation metrics critical for assessing model performance.

2.1 Machine Learning in Fraud Detection

Machine learning, a subfield of Artificial Intelligence (AI), involves the use of algorithms to create autonomous or semi-autonomous systems. It enables computers to learn and make decisions without being explicitly programmed. Machine learning can be broadly categorized into supervised and unsupervised learning algorithms (Géron, 2019).

In fraud detection, two primary approaches are used: anomaly detection and classification. Anomaly detection approaches the problem from an unsupervised learning perspective by identifying unusual patterns that may indicate fraud. In contrast, classification uses supervised learning to distinguish between fraudulent and non-fraudulent cases based on labelled data (Bambo, 2022).

2.1.1 Classifiers

Several machine learning models are commonly employed in fraud detection, including:

- **Logistic Regression:** A basic yet powerful machine learning algorithm for predicting binary outcomes (fraud or non-fraud). It is popular for its speed, low computational cost and ability to handle imbalanced data (Bambo, 2022).
- **K-Nearest Neighbours (KNN):** A simple algorithm that stores all available cases and classifies any new cases by taking a majority vote of its k best neighbours. To do this, it makes use of a distance function like the Euclidean distance (Bambo, 2022).
- **Decision Trees:** Another popular algorithm that learns rules to split or classify data. Popular for its interpretative nature. The decision trees algorithm provides a clear understanding of the decision process and can help interpret how fraud was committed (Bambo, B., 2022).
- **Random Forests:** An ensemble method that builds on multiple decision trees to provide classifications that are more accurate. It does this by averaging the results of individual decision trees, hence its predictive power is superior. It is however less explainable than decision trees. Random forests work well with very large training datasets that have a large number of input variables (Bambo, 2022).
- **Gradient Boosting:** An advanced ensemble technique that builds models sequentially to correct errors made by previous models (Zheng et al., 2023).

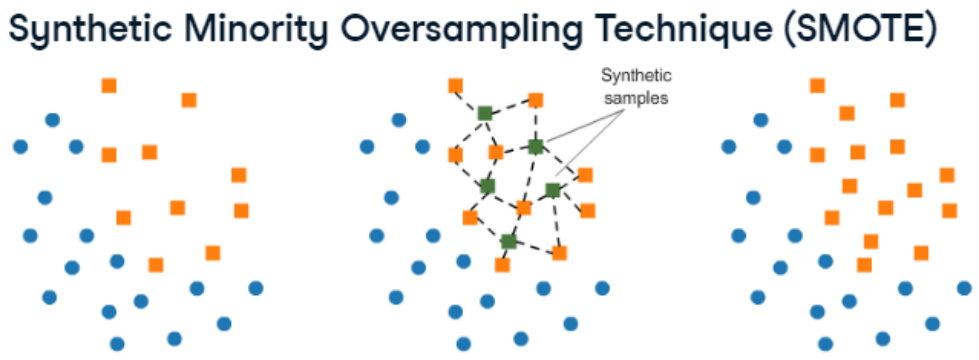
- **Support Vector Machines (SVM):** Effective in high-dimensional spaces and useful for cases where the number of dimensions exceeds the number of samples (Géron, 2019).

2.2 Class Imbalance

Class imbalance is a significant challenge in fraud detection due to the rarity of fraudulent cases compared to genuine ones. Machine learning algorithms can become biased toward the majority class, reducing their ability to accurately detect fraud (Galli, 2023). Since obtaining enough fraudulent cases to balance the dataset is often unrealistic, different sampling methods can be employed to address the issues related to class imbalance. There are generally two different approaches that can be used: oversampling or undersampling.

- **Oversampling** involves increasing the number of instances in the minority class (fraudulent cases) by duplicating existing ones or generating new synthetic samples (Galli, 2013).
- **Undersampling** reduces the number of instances in the majority class (non-fraudulent cases) by randomly removing some of these samples. While this can help balance the dataset, it can also lead to the loss of important information since it reduces the amount of data available for training (Galli, 2013).

Due to the potential loss of information associated with undersampling, the oversampling approach is usually preferred. Techniques like SMOTE (Synthetic Minority Over-sampling Technique) create new synthetic instances that are combinations of existing ones. This helps to balance the dataset without simply duplicating samples, thus avoiding overfitting (Galli, 2013). Additionally, ensemble methods like Random Forest and XGBoost can also be effective in handling imbalanced datasets.



[1] https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets

Figure 1. A visualization of how SMOTE is operated on the data. Retrieved from DataCamp, but originally from Kaggle.

2.2.1 Imbalanced-learn

Imbalanced-learn (Imblearn) is an open-source Python package designed to tackle imbalanced datasets. It provides tools and algorithms for processing imbalanced datasets and improving model performance, including methods like SMOTE.

2.3 Evaluation metrics

Choosing the right performance metrics is essential when working with imbalanced data. Common metrics like accuracy are misleading since the class imbalance is not accounted for, and as such the proportion of correctly identified instances out of the total is not interesting. For fraud detection, which is a classification problem, according to Brownlee (2021) suitable metrics include:

- **Precision:** The ratio of true positive predictions to the total predicted positives, indicating how many predicted fraud cases are actual fraud.
- **Recall:** The ratio of true positive predictions to the actual positives, indicating how many actual fraud cases are correctly identified.
- **F1 Score:** The harmonic mean of precision and recall, providing a balance between the two.
- **AUC-ROC:** A metric that evaluates the model's ability to distinguish between classes across all thresholds, emphasizing the trade-off between true positives and false positives.
- **F-beta Score:** A generalized version of the F1 Score that allows for a different balance between precision and recall. This is a suitable metric for highly imbalanced datasets as it allows for customization based on the importance assigned to recall.
- **Confusion Matrix:** A table used to describe the performance of a classification model by displaying the true positives, true negatives, false positives, and false negatives. It provides detailed insight into not just the errors being made by a classifier but more importantly the types of errors being made.

Metrics that provide better insights into performance on imbalanced data, such as ROC-AUC or F-beta score are particularly valuable (Brownlee, 2021).

2.3.1 Cross-validation

When evaluating the model's performance on the training data, cross-validation is employed to ensure that the test set is reserved for the final evaluation of the best performing model among those trained. During cross-validation, the dataset is divided into k smaller parts, called "folds". The model is then trained and evaluated k times, where each fold is used as the validation set once, and the remaining folds are used as the training set. This approach provides a more robust evaluation of

the model's performance because each observation could be included in both the training and validation sets during different iterations (Géron, 2019).

Handling imbalanced datasets during cross-validation, particularly with techniques like SMOTE, requires careful consideration to prevent data leakage and ensure accurate evaluation. Instead of applying SMOTE to the entire dataset before cross-validation, it's crucial to integrate the sampling technique within the pipeline. This ensures that SMOTE is applied only to the training folds during each iteration of cross-validation. Otherwise, the same instances may be used for both training and validation, compromising the model's ability to generalize effectively (Martin, 2019).

3 Method

This chapter outlines the methodological approaches and decisions made throughout the project, covering data exploration, preprocessing, modeling and the evaluation employed to measure performance and interpret the results.

3.1 Project Plan

To maintain flexibility throughout the project, an agile methodology was implemented. Initially, a project plan was formulated and structured into 8 distinct steps. During the process these steps were regularly reviewed and adjusted iteratively based on new findings.

Original Project Plan
Step 1. Project Overview ➢ Formulate the problem statement ➢ Decide on metrics for evaluation
Step 2. Load data and initial exploration ➢ Conduct an initial overview of the dataset ➢ Explore data structure, types, and overall quality ➢ Set aside data for evaluation (test set)
Step 3. Data Cleaning ➢ Handle missing values ➢ Transform to correct data types ➢ Address any inconsistencies in the data
Step 4. Exploratory Data Analysis (EDA) ➢ Visualize feature distributions and patterns ➢ Identify outliers ➢ Examine correlations and relationships between features
Step 5. Prepare Data ➢ Transform/encode categorical features ➢ Preprocessing pipeline ➢ Feature Engineering if applicable
Step 6. Feature Selection ➢ Calculate MI scores to investigate feature importance
Step 7. Modeling and Testing ➢ Overview of metrics ➢ Assess feature importance ➢ Conduct testing, evaluation and detailed modeling analysis
Step 8. Results and Conclusions ➢ Explore insights gained from the analysis ➢ Draw conclusions and propose actionable solutions

3.2 Dataset

The data used in this project was sourced from Kaggle[1] and consists of 1,000 records of auto insurance claim cases with 40 different attributes. Besides the class feature (fraud or non-fraud) the dataset includes customer information (e.g., age, sex, address), incident details (e.g., severity, location), car specifics (e.g., make, model), and claim amounts.

3.2.1 Class imbalance

The dataset shows significant class imbalance, with non-fraudulent cases representing approximately 75% of all claims. To ensure proportional class representation, the dataset was divided into training (80%) and test (20%) sets using a stratified sampling method.

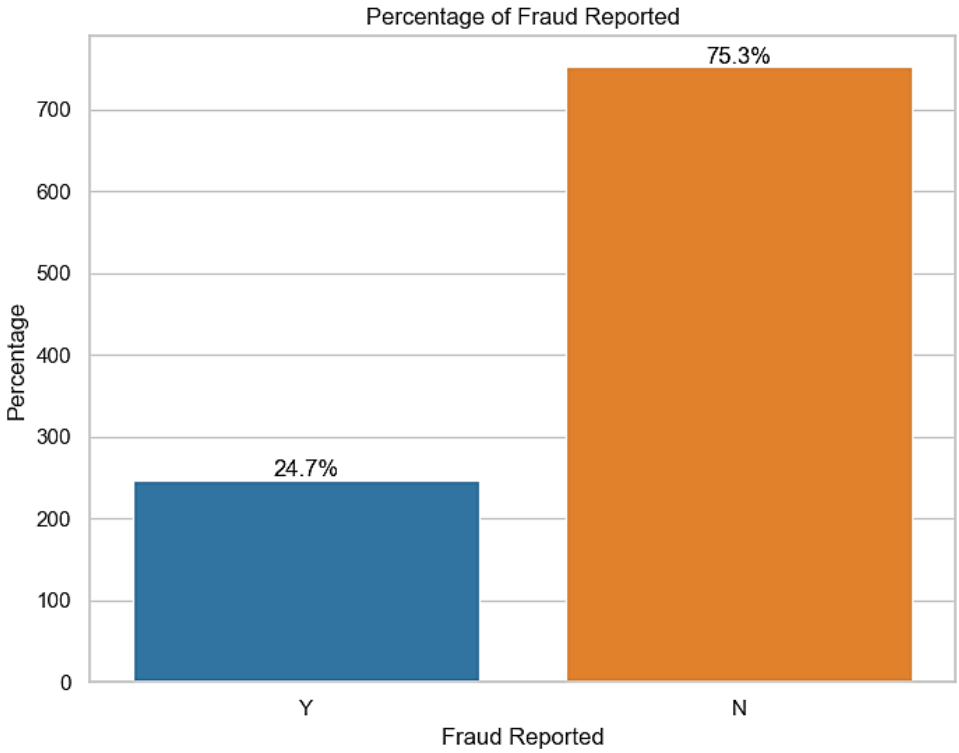


Figure 2. Bar plot illustrating the class distribution, highlighting the imbalance towards the genuine class.

3.3 Exploratory Data Analysis (EDA)

During the initial exploration of the dataset, it was observed that some features were incorrectly categorized as numerical. Additionally, missing values were represented by "?" in the "*PROPERTY_DAMAGE*" and "*POLICE_REPORT_AVAILABLE*" columns. To avoid data leakage, further

[1] https://www.kaggle.com/datasets/buntyshah/auto-insurance-claims-data/data.

exploration of the data was exclusively conducted on a copy of the training set. All preprocessing steps were subsequently applied to both sets.

3.3.1 Missing values

A significant number of missing values were detected in four categorical features. Before imputing the missing values, various analyses were conducted to determine if any patterns could explain the missing data. For example, a contingency table was created to examine the relationship between missing values in "*PROPERTY_DAMAGE*" and zero values in "*PROPERTY_CLAIM*". However, no significant patterns were identified from the analyses.

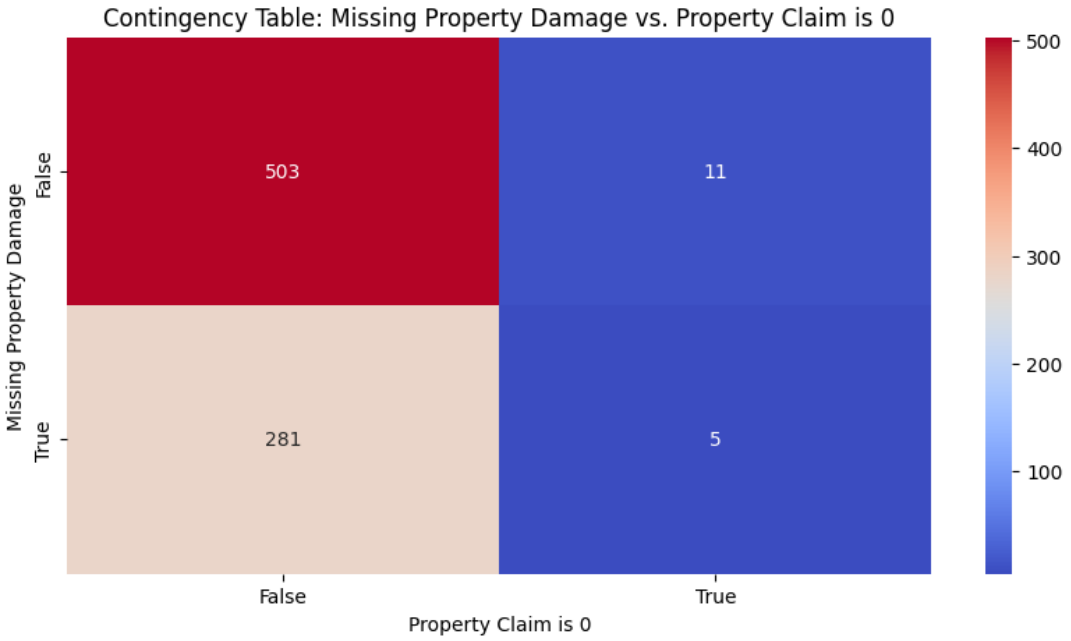


Figure 3. Heatmap of the contingency table showing the relationship between missing values in property damage and zero values in property claim.

As a result, the missing values for "*PROPERTY_DAMAGE*" and "*POLICE_REPORT_AVAILABLE*," originally represented as "?", were interpreted and imputed as "NO." For the "*COLLISION_TYPE*" feature, the mode was used to fill in the missing values. Missing values in "*AUTHORITIES_CONTACTED*" were imputed with the value "None".

3.3.2 Feature distributions

Both numerical and categorical feature distributions were visualized and examined. Many zero values were observed in some of the numerical features, such as "*CAPITAL GAINS*", "*CAPITAL LOSS*" and the different claim amounts. Since the numerical features will be scaled during preprocessing, this was not considered problematic. For categorical features, it was noted that vehicle theft and incidents where the car is parked were less common than other incident types.

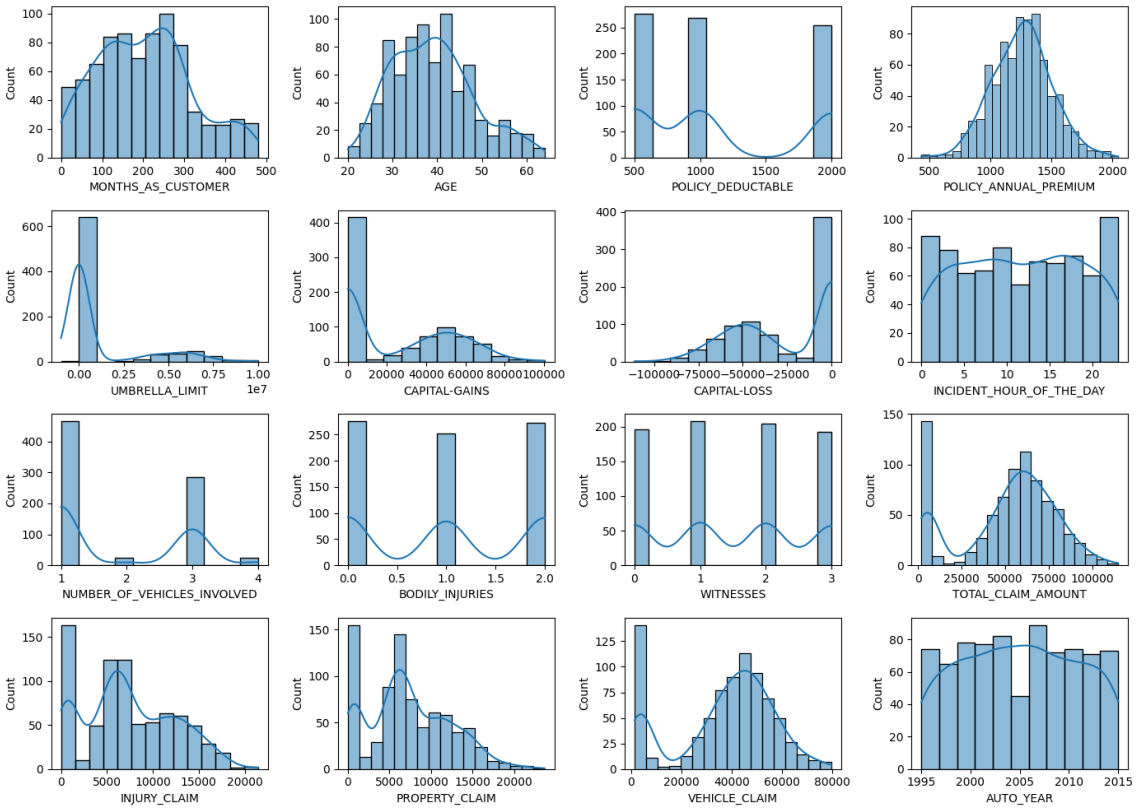


Figure 4. Histograms illustrating the distribution of numerical features, showing their spread across various bins.

3.3.3 Outliers

Outliers were detected through boxplot visualization and percentage-based methods, particularly in the "UMBRELLA_LIMIT" feature. Other features, such as policy annual premium, age, and property claim, also contained some outliers, although fewer.

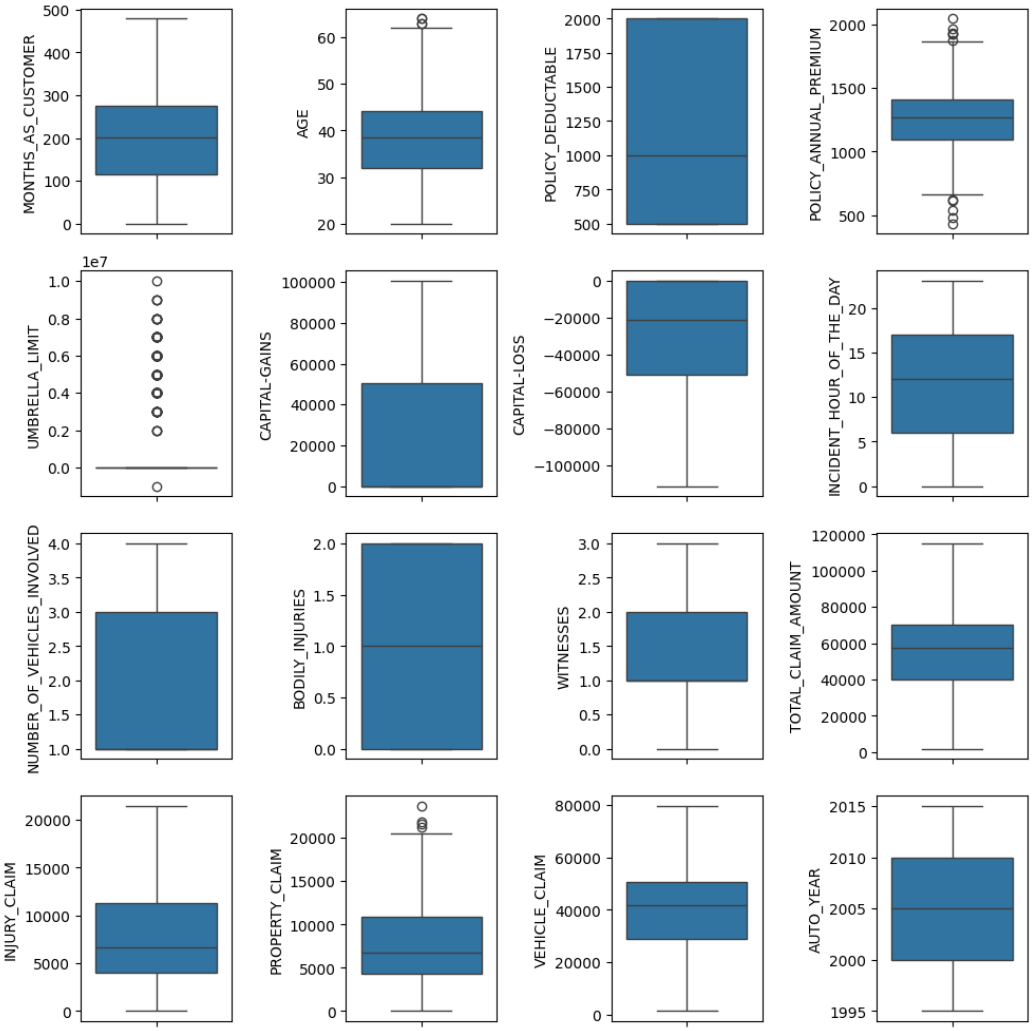


Figure 5. Box plot visualization highlighting outliers in some of the numerical features.

3.3.4 Correlation Analysis

Using Spearman's rank to calculate correlations between numerical features revealed that "*AGE*" and "*MONTHS_AS_CUSTOMER*" are highly correlated. The "*TOTAL_CLAIM_AMOUNT*" is also highly correlated with the "*VEHICLE_CLAIM*" amount, as well as the two other claim amounts. All the claim amounts are correlated with each other.

3.3.5 Feature Selection

During the feature selection process, Mutual Information (MI) scores were calculated, with zero indicating independence and higher values indicating greater dependency. Features with MI scores below a threshold of 0.01 were removed and assessed during cross-validation. However, no significant improvement was observed with these features dropped, likely due to the small dataset size. Therefore, the decision was made to retain all features.

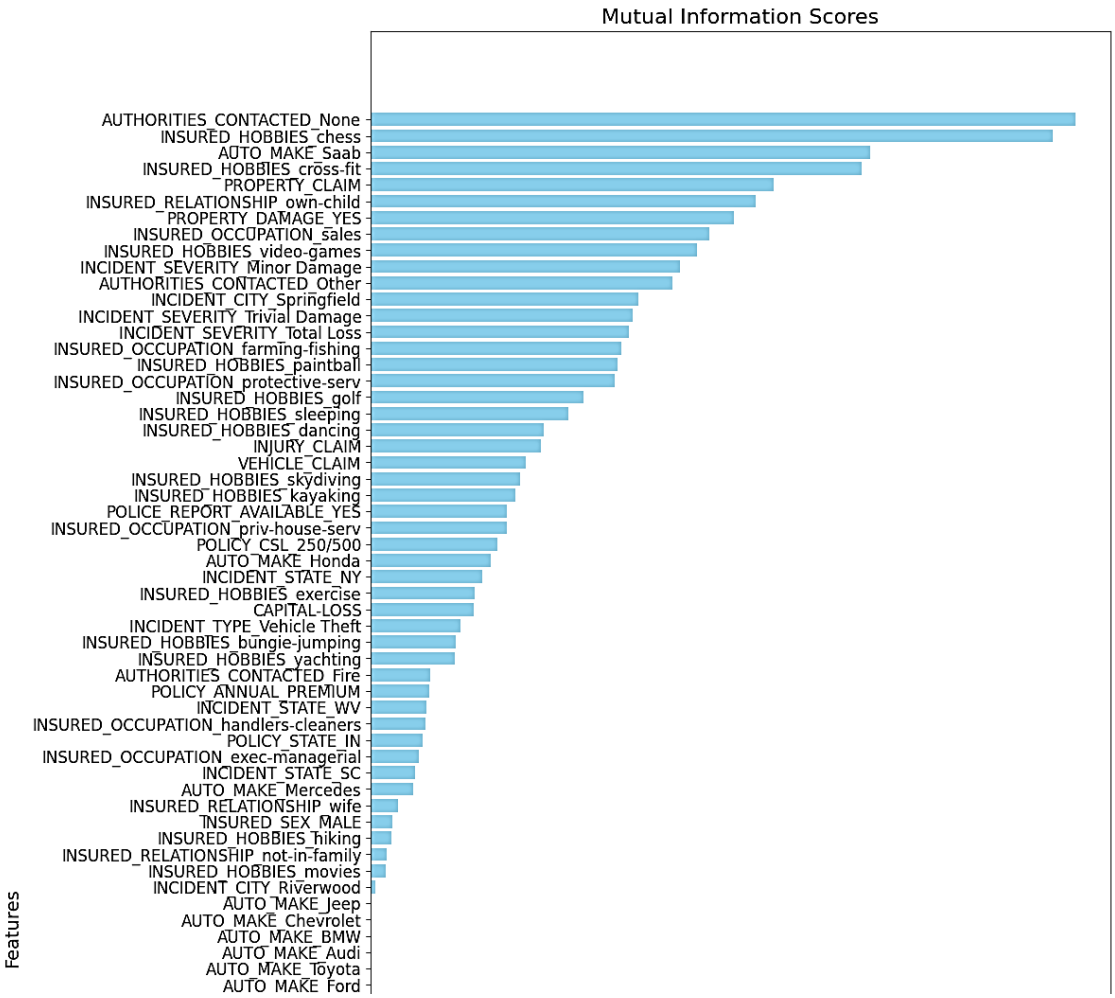


Figure 6. Illustration of the MI-scores for feature importance calculations. The full plot is accessible in the notebook.

3.3.6 EDA Summary

- The dataset displayed balanced distributions except for frequent zero values in some numerical features, which will be handled during scaling.
- Significant correlations were observed between certain numerical features.
- Outliers were notably present in "UMBRELLA_LIMIT".
- Several categorical features contained numerous unique values, which questions their relevance in prediction.
- Using only the most important features based on their MI-score did not significantly improve the model's predictive abilities.

3.4 Data Preprocessing

Following the EDA, a custom preprocessing pipeline was developed to apply the necessary data cleaning procedures and transformations. To ensure consistent preprocessing for both the training and test sets, the pipeline includes the following steps:

- Replacing "?" with NaN
- Dropping columns that are either empty, contain too many unique values or correlate with other features
- Imputation of missing values as described earlier
- Dummy transformation of categorical features
- Scaling / normalization of the data (MinMaxScaler)

3.4.1 Applying SMOTE

After preprocessing, SMOTE (Synthetic Minority Over-sampling Technique) was applied to balance the class distribution. Below is the difference in class distribution before and after applying SMOTE.

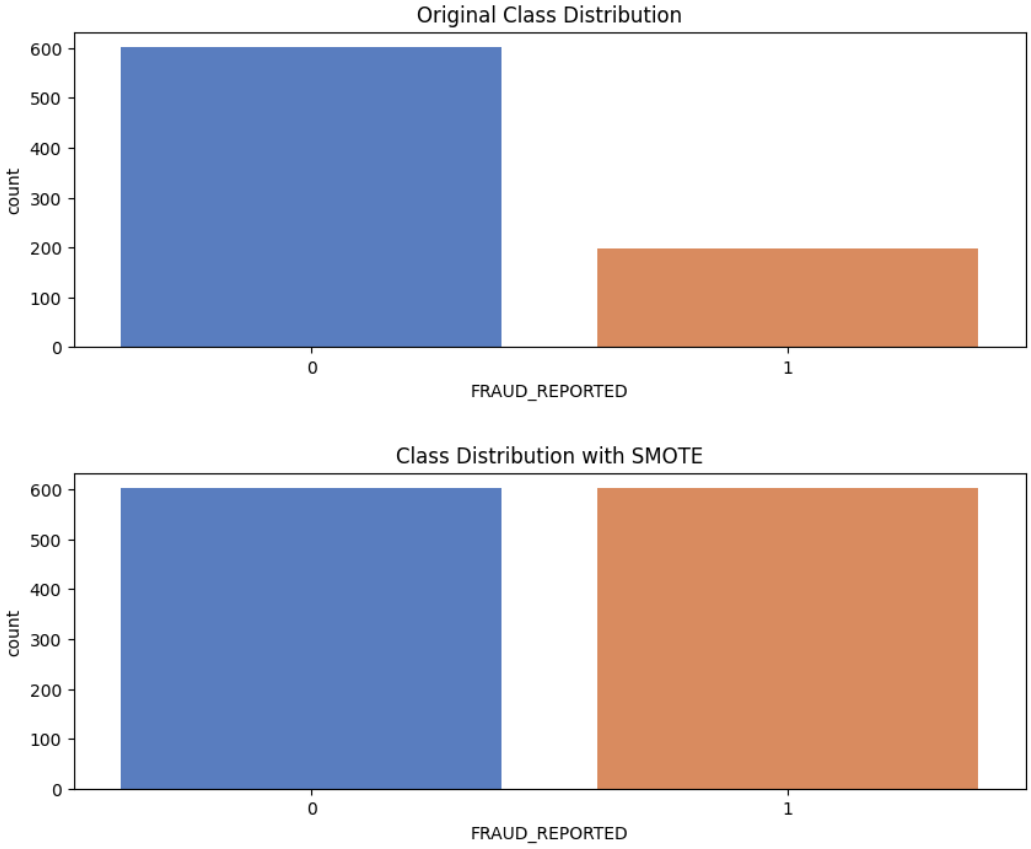


Figure 7. Count plots demonstrating the difference in class distribution before and after applying SMOTE.

3.5 Modeling

To evaluate the impact of applying SMOTE on model performance in detecting fraud, two distinct modeling pipelines were constructed: one using the original dataset and another using the SMOTE-transformed data. Models were evaluated using cross-validation on the training set, with the best performing models being subjected to hyperparameter tuning.

3.5.1 Model Selection

Model selection was guided by theoretical principles and common practices in supervised fraud detection. The focus was on selecting models that balance predictive power with interpretability, ensuring the models are both effective and understandable.

3.5.2 Model Validation

Initial model evaluation was conducted using cross-validation to establish baseline performance metrics. Models were trained with default parameters on both the original and SMOTE-transformed datasets. The primary metrics for evaluation were ROC-AUC score and F-beta score, which are well suited for handling imbalanced datasets. Given the higher importance of recall (minimizing false negatives), the beta value was set to 5, prioritizing recall over precision. However, maintaining a balanced model was deemed crucial to avoid generating excessive false positives, which could put unnecessary burden on the investigation department.

The models showing the best baseline performance (according to the metrics named) were then subjected to hyperparameter tuning to further optimize their predictive capabilities.

3.5.3 Model Optimization

Hyperparameter tuning involved an exhaustive search for the best parameter combinations, evaluated through cross-validation. Among the tuned models, the Support Vector Classifier (SVC) performed the best, closely followed by Logistic Regression. Despite the slightly better performance of SVC, Logistic Regression was selected for the final evaluation on the test data due to its superior interpretability and the minimal difference in performance metrics between the two models.

	Classifier	ROC-AUC	F-beta	FP	FN	Precision	Recall
1	SVC	0.8689	0.8814	94	21	0.6531	0.8939
2	Logistic Regression	0.8588	0.8622	94	25	0.6479	0.8737
3	Gradient Boosting	0.8354	0.8403	110	29	0.6057	0.8535

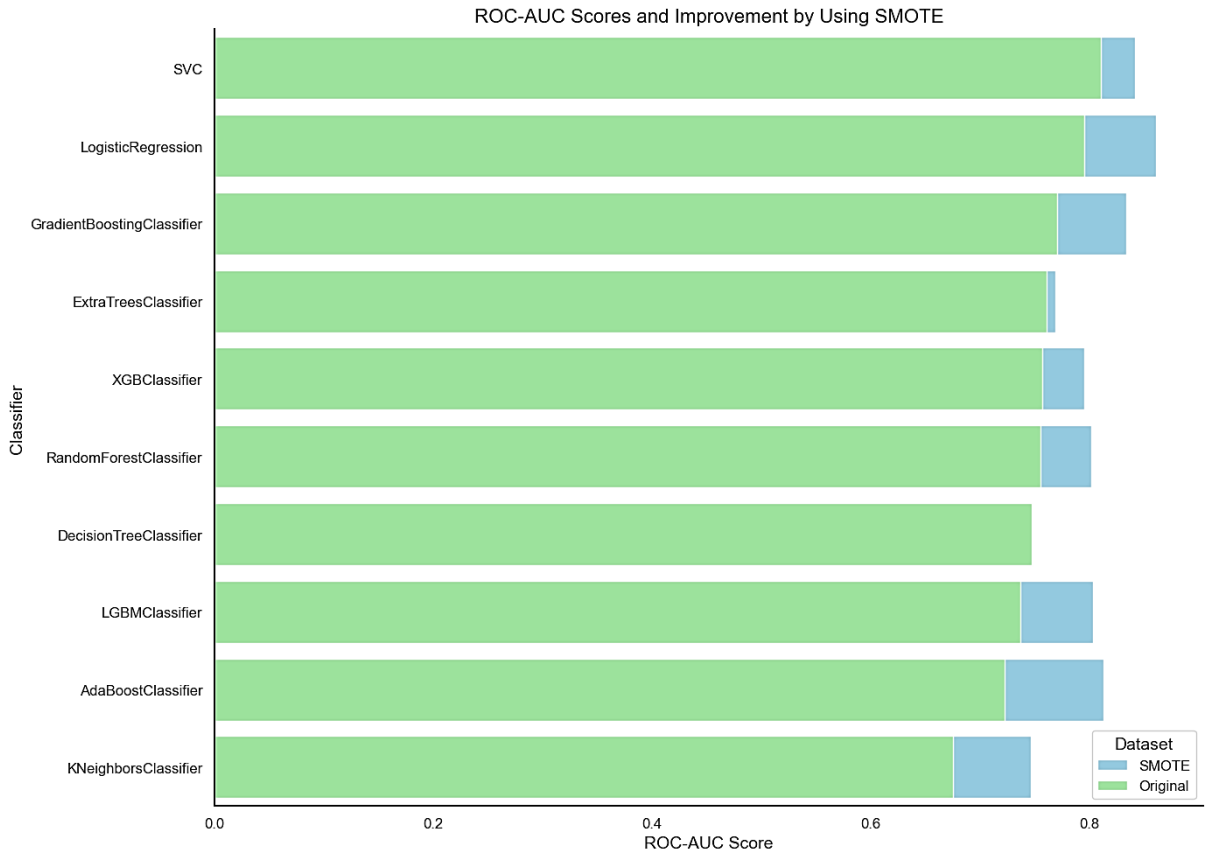
4 Results and Discussion

In this chapter the results are presented and discussed, focusing on model selection, evaluation, and critical analysis of the data, and reflection on the choices made throughout the study.

4.1 Performance Improvement with SMOTE Transformed Data

Applying SMOTE improved the performance of all models, though the improvement was modest for some, like the Extra Trees Classifier and Decision Tree Classifier. The top performing models were the Support Vector Classifier (SVC), Logistic Regression, and Gradient Boosting Classifier, which outperformed the other models on both the original and the SMOTE-transformed datasets.

The visualisation below illustrates the improvement in model performance with SMOTE-transformed data, with the additional ROC-AUC scores highlighted in blue.



SMOTE generally improved recall across most models, as anticipated, because it increases the number of minority class samples. However, it also tended to reduce precision in some cases, which could be due to the introduction of synthetic samples that may not generalize well. The varying performance of models with SMOTE may be due to their different abilities to handle synthetic data effectively. Additionally, the small dataset size, with only 1,000 unique instances with 247 fraudulent

cases, likely affects the effectiveness of SMOTE. It should however be noted that these results reflect baseline model performance, indicating that outcomes could differ with model optimization.

It is also important to note that SMOTE does not always yield better results. When fraud cases are scattered and widely spread across the data, synthetic sampling techniques like SMOTE can introduce bias. This scattering could contribute to the observed reduction in precision and the inconsistent performance of some models, especially given the dataset's limited size.

4.2 Final Model's Performance

The best performing model was Logistic Regression on the SMOTE-transformed dataset. The table below is demonstrating the model's performance metrics evaluated on both training and test data.

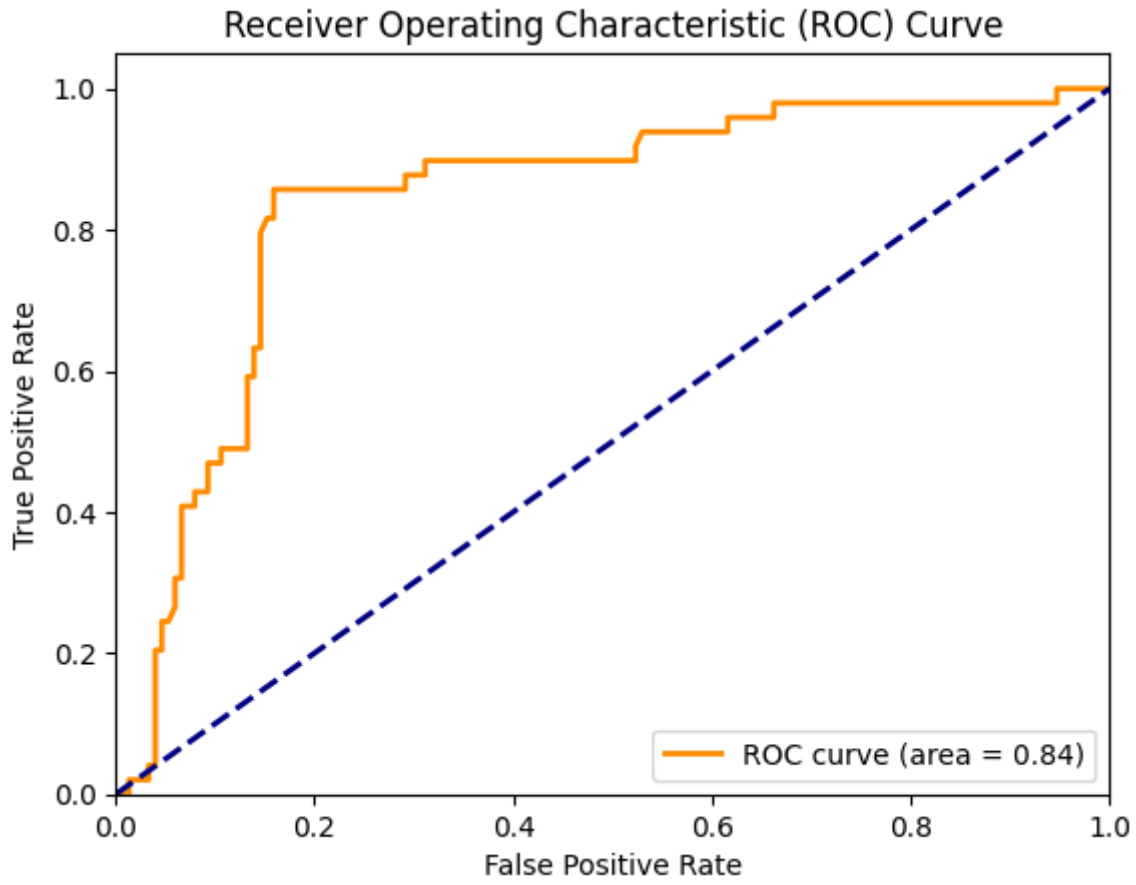
Metric	Value (test)	Value (training)
ROC-AUC	0.8491	0.8689
F-beta (beta=5)	0.8459	0.8814
False Positives (FP)	24	94
False Negatives (FN)	7	21
Precision	0.6364	0.6531
Recall	0.8571	0.8939

The high ROC-AUC Score demonstrates that the model is effective at distinguishing between fraudulent and non-fraudulent claims. However, the precision of 0.6364 suggests that there is room for improvement in reducing false positives, as the model sometimes misclassifies non-fraudulent cases as fraudulent. This could be due to the introduction of synthetic samples from SMOTE, as mentioned earlier, together with the dataset's limited size. Therefore, further data manipulation or model tuning may have limited impact in this case.

Overall, the Logistic Regression model with SMOTE transformation performs well in identifying fraudulent claims. The consistency between training and test metrics also suggests that the model has good generalization capabilities.

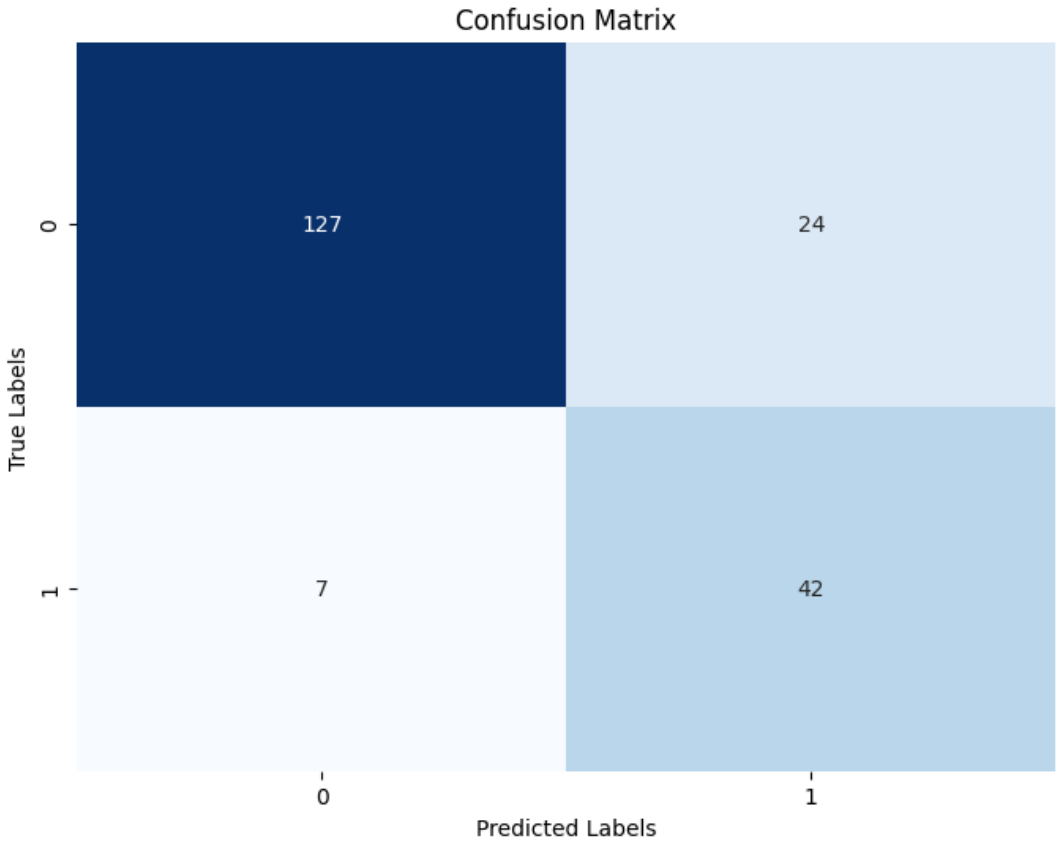
4.2.1 ROC-Curve

The ROC curve, shown below, illustrates how well the model performs at different decision thresholds. It confirms that the model effectively balances sensitivity (true positives) and specificity (true negatives). While the model is generally good at distinguishing between fraudulent and non-fraudulent claims, reducing false positives further could improve its overall performance.



4.2.2 Confusion Matrix

The confusion matrix shows that most cases were correctly predicted. The model incorrectly predicted 7 fraudulent cases as non-fraudulent and 24 non-fraudulent cases as fraudulent. Because the dataset is small (only 200 cases), these mistakes have a bigger impact on the results. With limited data, it's harder to achieve perfect accuracy, and the errors in predicting fraudulent and non-fraudulent cases become more noticeable.



4.2.3 Model Generalization and Overfitting

The model's ability to generalize well was evaluated by comparing its performance on both training and test data. As mentioned earlier, the results were similar for both sets, which means the model is not overfitting and can generalize well to new data. The learning curves also show that the training and validation scores are close to each other, which supports the idea that the model performs consistently across different data splits.

4.3 Feature Importance

The feature importance analysis of the final model provides insights into which factors are most useful for detecting fraud in auto insurance claims. The table below lists the key features along with their importance scores, which indicate how much each feature contributes to the model's ability to identify fraudulent claims.

Key Features	Score
INSURED_HOBBIES_chess	3.0887
INSURED_HOBBIES_cross-fit	2.7745
INCIDENT_SEVERITY_Minor Damage	-2.5400
INCIDENT_SEVERITY_Total Loss	-2.4737
INCIDENT_SEVERITY_Trivial Damage	-2.0892
WITNESSES	0.3644
INCIDENT_STATE_SC	0.3378
INCIDENT_TYPE_Single Vehicle Collision	0.2168
BODILY_INJURIES	0.1289
POLICY_CSL_250/500	0.0754
COLLISION_TYPE_Rear Collision	0.0580

4.3.1 Interpretation

The table above shows that hobbies such as chess and cross-fit are strongly associated with fraud. This might suggest that people with these hobbies are more likely to commit fraud, but it’s important

to remember that these are correlations, not causes. Features like hobbies should be interpreted carefully as they might reflect underlying trends that we don't fully understand. It's also possible that fraudsters might include certain hobbies on purpose to avoid suspicion.

On the other hand, features like minor or trivial damage have a negative impact on fraud prediction, suggesting that serious damage cases are more likely to be fraudulent.

5 Conclusions

In summary, the following conclusions can be drawn in relation to the purpose of the project and the questions that the analysis aimed to answer.

5.1 How do different machine learning models compare in their ability to detect fraudulent auto insurance claims?

The comparison of different machine learning models in detecting fraudulent auto insurance claims revealed that models varied in their performance. Support Vector Classifier (SVC), Logistic Regression and Gradient Boosting Classifier were the top performers, demonstrating stronger predictive capabilities compared to the other models. Overall, models with robust handling of imbalanced data performed better in detecting fraudulent claims.

However, it should be noted that only baseline performance was compared for the models on the original dataset. As such, it cannot be ensured that other methods (like ensemble methods or boosting techniques) would not outperform the other models with optimized hyperparameters.

5.2 What impact does the application of SMOTE (Synthetic Minority Over-sampling Technique) have on the performance of fraud detection models?

The application of SMOTE had a notable impact on the models' performances. In most cases, SMOTE enhanced the recall of the models but in some cases reduced precision. This is likely due to the introduction of synthetic samples that may not generalize well, leading to more false positives. The degree of improvement varied among models. The models that benefitted the most from SMOTE was the Ada Boost Classifier and the KNN. However, they still did not outperform the SVC, Logistic Regression and Gradient Boosting Classifier.

5.3 Which features are contributing the most to predicting fraud?

The analysis of feature importance from the final model highlighted that hobbies such as chess and cross-fit had a positive impact on predicting fraud, where as incident severities classified as minor or trivial damage had a negative impact on fraud prediction. The importance of features such as hobbies may reveal underlying data patterns useful for feature engineering or further investigative analysis. The analysis also calls for further investigation in cases classified as major damage, as these could be more prone to be fraudulent.

Given the limits of the current data, we should be careful with putting too much weight into the feature importance results. However, they could be a useful starting point for further investigation as more data is collected.

5.4 Summary

In summary, the study demonstrated that while different machine learning models vary in their effectiveness at detecting fraudulent claims, Logistic Regression and other robust models like SVC and Gradient Boosting Classifier performed the best, especially with the application of SMOTE. SMOTE significantly improved recall but required careful management of precision. The analysis of feature importance revealed key factors in fraud detection and suggested the need for careful interpretation and further investigation. In many cases of fraud or criminal activity, perpetrators change their methods. This is especially true for organized crime, and less so for individual fraudsters. Therefore, it's crucial to maintain manual handling in fraud investigations and continually retrain the model with new data. Integrating the model into the investigation system ensures it stays updated and learns new trends and patterns as they emerge.

The dataset is relatively small, with only 1,000 individual observations, sufficient for a use case but insufficient for production deployment. More data is desirable for training robust models.

References

Association of Certified Fraud Examiners & SAS Institute (2024). 2024 Anti-Fraud Technology Benchmarking Report.

Bambo, B. (2022). Best Machine Learning Algorithms for Fraud Detection. Retrieved from https://sqream.com/blog/fraud-detection-machine-learning/.

Galli, S. (2023). Dealing with Imbalanced Datasets in Machine Learning: Techniques and Best Practices. Retrieved from https://www.blog.trainindata.com/machine-learning-with-imbalanced-data/.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow (2nd ed.).* O'Reilly Media.

Larmtjänst (2024). Försäkringsbolagen avslöjade bedrägeriförsök för 682 miljoner kronor. Retrieved from https://www.larmtjanst.se/Aktuellt1/Press/2024/forsakringsbolagen-avslojade-bedrageriforsok-for-682-miljoner-kronor/.

Martin, D. (2019). How to do cross-validation when upsampling data. Retrieved from https://kiwidamien.github.io/how-to-do-cross-validation-when-upsampling-data.html.

Svensk Försäkring & Larmtjänst (2024). Försäkringsbedrägerier i Sverige 2023. Retrieved from https://www.svenskforsakring.se/globalassets/rapporter/forsakringsbedragerier/2023-forsakringsbedragerier.pdf/.

Zheng, H., Peng, F., Tian, Y., Zhang, Z., & Zhang, W. (2023). Insurance Fraud Detection Based on XGBoost. *Academic Journal of Computing & Information Science, 6*(8), 68-74.