Auto Insurance Fraud Detection

Workshop I – "Actuarial Science"

Raikibul HASAN
Master of Data Science
University of Luxembourg

In this project, we were asked to experiment with an insurance dataset from Kaggle, and to explore how machine learning model can be used to forecast the target (fraud detection). We were expected to gain experience using our knowledge of actuarial science, common data mining and machine learning library and were expected to submit a report about the dataset and the code. After performing the required tasks on a dataset of my choice, herein lies my final report.

One of the major and most prevalent issues that insurers deal with is fraud. This report focuses on vehicle insurance company claim statistics. The objective of this project is to develop a system that can identify fraudulent auto insurance claims.

Keywords: Actuarial Science, Fraud detection, Machine Learning, Auto insurance.

Visit the repository in GitHub: https://github.com/rakiiibul/actuarial_science

TABLE OF CONTENTS

I. Contents

I. Data and Associated insurance problem

A. Dataset description

The “Auto Insurance Claims Data” dataset has 1000 rows and 40 columns. This dataset includes customer information (age, insured_sex, insured_education_level, insured_hobbies) as well as information on the insurance policy (policy_number, policy_state, policy_annual_premium). Additionally, it contains information about the accident that served as the foundation for the claims. The column titles include things like the incident location, severity, auto model, auto year, and bind date for policies. The target column named fraud reported which tell us the claimed reported as fraud or not.

B. Problem Statement

One of the major and most prevalent issues that insurers deal with is fraud. This report focuses on vehicle insurance company claim statistics. For each insurer, fraudulent claims can be quite expensive. It is crucial to understand which assertions are true and which are false. Insurance companies are unable to individually examine every claim since doing so would be prohibitively expensive. In this report, we'll make use of data, the insurers' most valuable resource while battling fraud. We use a number of attributes that the insurer includes in the data regarding the claims, insured individuals, and other conditions.

II. Data Preprocessing

A. Missing Values & Data cleaning

The first step that we need to do is data cleaning. Some columns include the null value, which is used to denote missing, ambiguous, or possibly nonexistent values. we can see some missing values denoted by '?' so, first we replace these missing values with ‘NA’. Columns property damage, police report available, collision type, and _c39 in our dataset have null values in amounts of 360, 343, 178 and 1000, respectively. The total amount of missing values in this dataset is about 5%. Let us drop the _c39 column as it has all the values are missing. So, the amount of missing data is change to 2.4%. Missing values in the dataset can be replaced in a variety of ways, like we can replace these with mean, median or mode. In our case we replace missing values with model. Exploratory data analysis was conducted making two subset of data one for numerical and another one is categorical data. The target variable is Fraud_reported column. There were 753 non-frauds opposed to 247 frauds. In contrast to 75.3% non-false claims, 24.7% of the data were fraudulent claims.

B. Exploratory Data Analysis

The category and numerical data are visualized using bar plots and histograms. Additionally, box plots are employed to show the numerical data, aiding in the identification of outliers for the following phase. To see the relationships between the variables, utilize the correlation matrix. High relationship between age and months_as_customer, according to the correlation matrix. Additionally, there is a significant correlation among the total_claim amount, injury claim, property claim, and vehicle_claim which is the sum of all others claim.

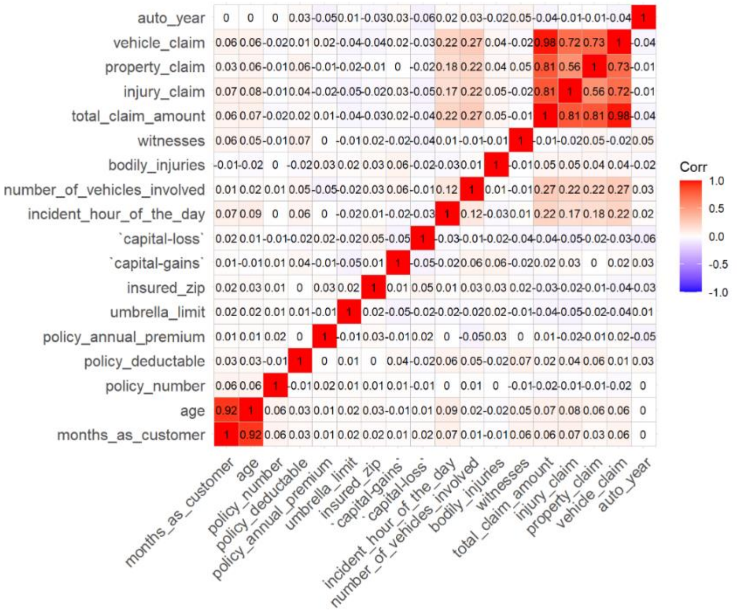
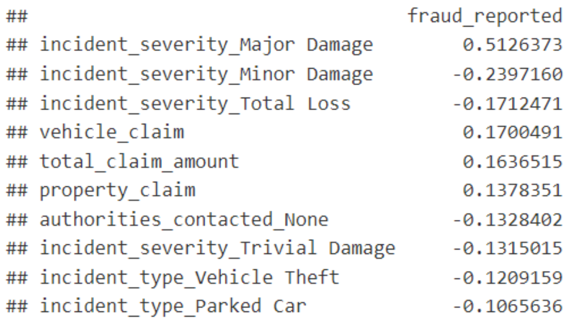


Figure 1: Correlation among variables

C. Feature Engineering & Feature Selection

Handling outliers is the most crucial step in the feature engineering process since it guarantees that our model is trained on accurate data, which produces accurate models. From box plot in previous step, we see some of the columns have outliers The quantile is used for removing outliers.

The second step is feature encoding, converts categorical data into numerical values that ML systems can comprehend. One hot encoding is used for this purpose. Later, we also look for the correlations that are most closely related to our target variables and select those which are above 10%.



```
##                                     fraud_reported
## incident_severity_Major Damage          0.5126373
## incident_severity_Minor Damage         -0.2397160
## incident_severity_Total Loss           -0.1712471
## vehicle_claim                           0.1700491
## total_claim_amount                      0.1636515
## property_claim                          0.1378351
## authorities_contacted_None             -0.1328402
## incident_severity_Trivial Damage       -0.1315015
## incident_type_Vehicle Theft            -0.1209159
## incident_type_Parked Car               -0.1065636
```

Table 1: Highly co-related features with Target Variable

III. Description of the models used

A. SVM Classifier

A supervised machine learning approach called Support Vector Machine (SVM) is used for both classification and regression. Although we also refer to regression concerns, categorization is the most appropriate term. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM method. The number of features determines the hyperplane's size. The hyperplane is essentially a line if there are just two input features. The hyperplane turns into a 2-D plane if there are three input features. Imagining something with more than three features gets challenging.

B. Decision tree

An algorithmic strategy that can split the dataset in numerous ways based on various criteria can be used to create decision trees. A tree's decision nodes, where the data is divided, and leaves, from which we derive the result, are its two key components.

The most effective and well-liked categorization and predictions technique is the decision tree. A decision tree is a tree structure that resembles a flowchart, where each internal node represents a test on an attribute, each branch a test result, and each leaf node (terminal node) a class label.

C. Logistic regression

The probability of a target variable is predicted using the supervised learning classification algorithm Know ad logistic regression. Since the target or dependent variable is binary in nature, there are only two potential classes: 1 (which denotes success/yes) and 0 (which denotes failure/no). A logistic regression model makes mathematical predictions about P(Y=1) as a function of X. One of the most basic machine learning algorithms, it may be applied to a number of classifications issues.

D. Random forest

As is common knowledge, a forest is made up of trees, and a forest with more trees will be more sturdy. Similar to this, a random forest algorithm builds decision trees on data samples, obtains predictions from each one, and then votes to determine the best option. Because it averages the results, the ensemble method—which is superior to a single decision tree—reduces over-fitting.

IV. Analysis of the Results & Conclusion

We split data, 80% for training and 20% for testing. Our top model, Support Vector Machine (SVM), has an accuracy rating of 83%. Out of 34 affirmative cases, our model predicts 24 true positive cases, and 142 true negative cases, out of 166 cases. Out of 200 instances, it predicts 21 false positive cases and 13 false negative cases. This results in a f1 score of 89.31%.

A. Evaluation Metrices

F1-score: A better indicator of cases that were mistakenly classified than the accuracy matrix is the F1 score, which is the harmonic mean of precision and recall. The formula of F1 score:

$$\text{F1-score} = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 * \frac{(\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})}$$

Model	SVM	Decision Tree	Logistic Regression	Random Forest
F1-Score	89.31	88.68	89.3	88.75

Table 2: F1 score of used models

Precision: This term refers to the percentage of correctly detected positive instances among all of the projected positive cases. When the costs of False Positives are high, it is therefore helpful.

Model	SVM	Decision Tree	Logistic Regression	Random Forest
Precision	91.61	88.41	91.61	87.95

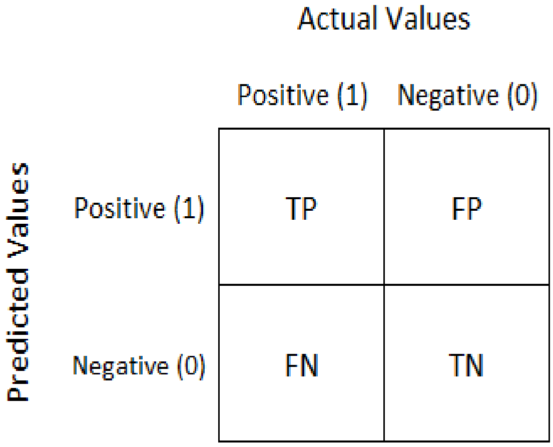
Table 3: Precision score of used models

Recall: This statistic represents the proportion of all correctly identified positive cases among all actual positive cases. When the cost of False Negatives is high, it is crucial.

Model	SVM	Decision Tree	Logistic Regression	Random Forest
Recall	87.11	88.95	87.11	85.88

Table 4: Recall score of used models

Confusion matrix: A table that frequently describes how a classification model, also known as a "classifier," performs given a set of test data for which the true values are known.



Predicted Values	Actual Values	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

TP - the true-positive, which cases are positive, and our model predict positive.TN-true negative, our actual value are negative, and model also predict negative. FP- false Positive: the cases were negative but were predicted to be positive, and False Negative: the cases are positive but were anticipated to be negative. In the test set, we have 166 Actual Negative and 34 positive cases. The results are given bellow:

SVM		Decision Tree		Logistic Regression		Random Forest	
24	21	18	18	20	18	17	17
13	142	19	145	17	145	20	146

Table 5: Confusion Matrix of different models

Accuracy: It is one of the more visible indicators and represents the total number of cases that were correctly identified. When all classes are equally essential, it is most frequently utilized. (The bellow table shows the comparison of different model, The SVC_balanced is covered with balanced data, basically we again try to look for the change if we balance our dataset is the model accuracy improve or not).

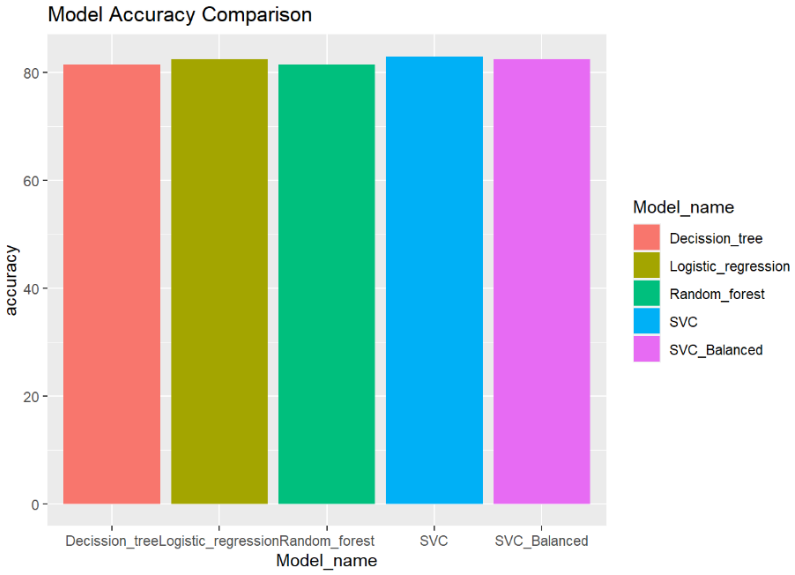


Figure 2: Model Accuracy Comparison

B. Discussion and Conclusion:

We deploy four distinct classifiers for this project: SVM, Decision Tree and Logistic Regression and Random Forest. These four models are used to examine three different approaches to addressing imbalance classes, hyper parameter adjustment, and showing roc curve (figure:3 Appendix) of the models.

A SVM with an F1 score of 89.31% and detect 24 fraud cases out of 34 is the best and final fitted model. In conclusion, the model proved highly accurate at differentiating between legitimate and fraudulent claims.

The objective of this project is to develop a system that can identify fraudulent auto insurance claims. The difficulty with detecting fraud using machine learning is that fraudulent claims are much less frequent than legitimate ones. (Figure-5: Appendix) Fraud detection in the insurance industry is a difficult task due to the various types of fraud and the low proportion of known fraud cases in typical data sets. When creating fraud detection models, it is important to balance the cost of false alarms with the benefits of preventing losses. The study has several restrictions. Another limitation of this study is the limited sample size. Larger data sets increase the stability of statistical models. Machine learning methods can improve the accuracy of predictions, allowing loss prevention teams to achieve a high level of coverage with a low rate of false positives. Fraudulent insurance claims can involve a variety of actions taken by individuals to gain an unfair advantage from the insurance company, such as fabricating the details of an incident, providing false information, and exaggerating the extent of damage.

APPENDIX

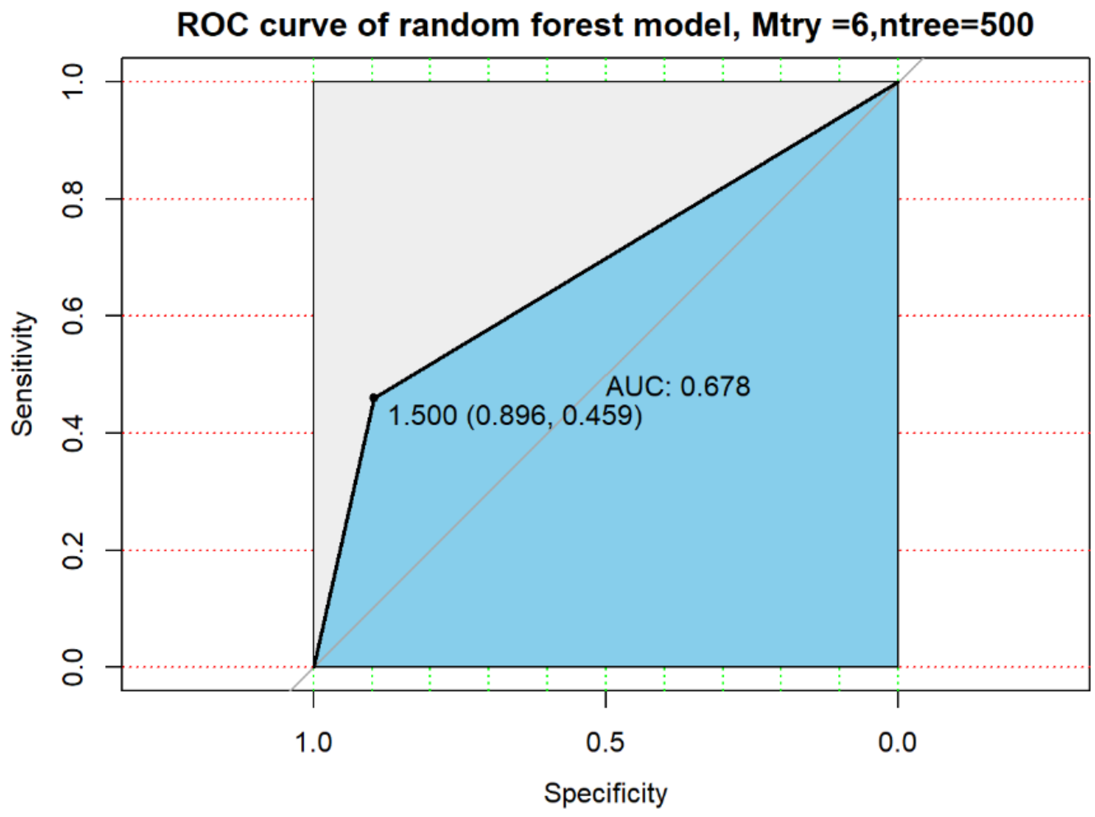


Figure 3 ROC Curve of Decision Tree

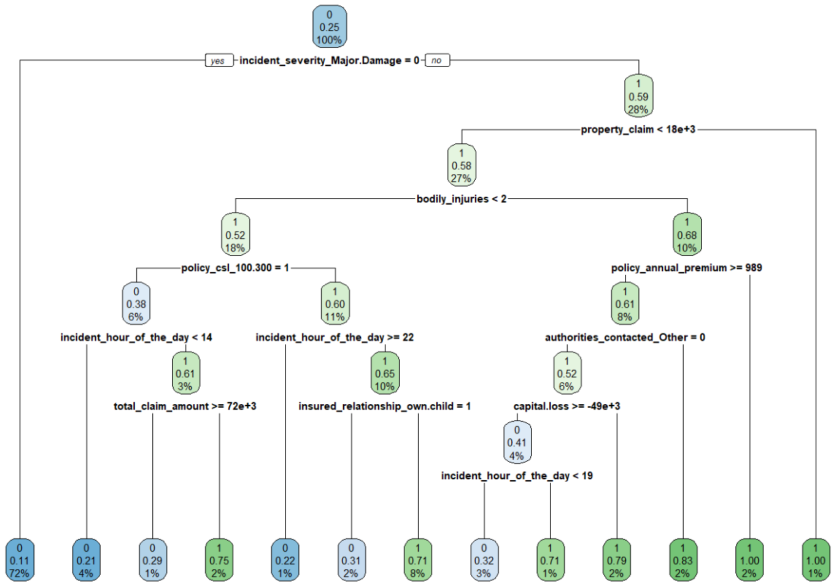


Figure 4: Decision Tree

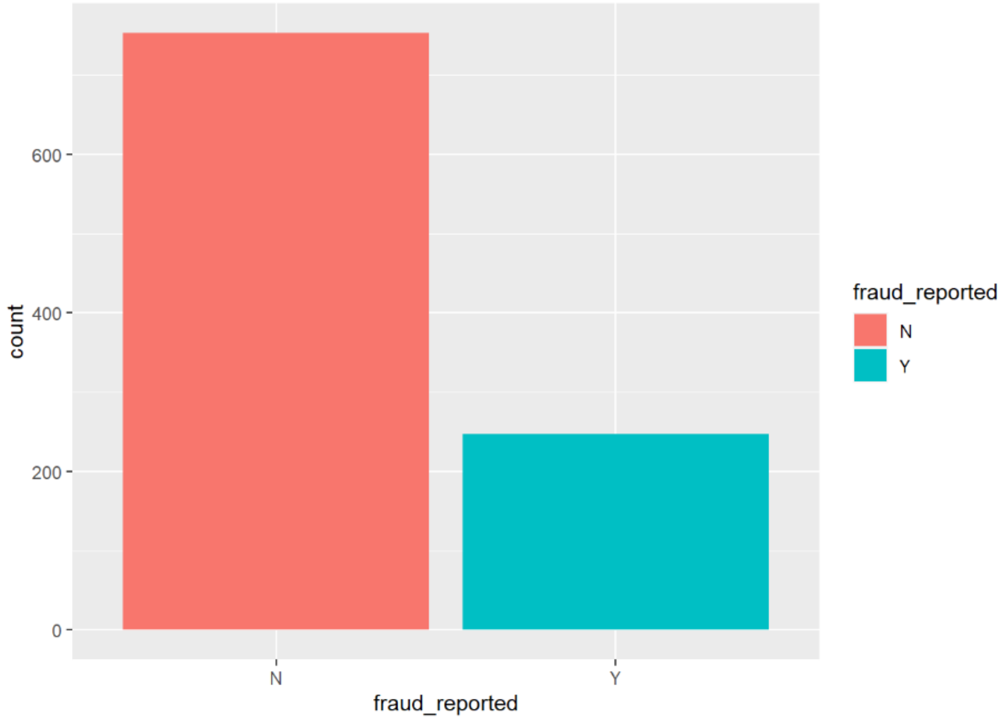


Figure 5: Imbalanced Dataset