Machine Learning Project

IDENTIFICATION OF FRAUDLENT CLAIMS IN CAR INSURANCE

Brian Douglass | Svetlana Temirov | Priyanka Patel

DATA 7703 | SEMESTER 2, 2022 UQ

We give consent for this to be used as a teaching resource.

Abstract

The cost of Insurance fraud not only affects insurance companies but also their policy holders through the increasing of premiums to pay for fraudulent claims. Being able to detect fraudulent cases based off features within the insurance claims datasets would help insurance companies to identify, investigate and limit fraudulent activity. This study has looked at a dataset involving car insurance claims with 6% fraudulent and 94% non-fraudulent flagged cases. The imbalanced nature of the data required synthetic data points to be created to allow for analysis. The use of SMOTE-Tomek to both over-sample with SMOTE and then under-sample based off those samples with Tomek-links was used. Once the data set was balanced, a variety of machine learning techniques were employed to analyse the features including Random Forest, Support Vector Machines and Extreme Gradient Boost. These three methods were then tested in an ensemble. Each model was tuned with iterations through a variety of hyperparameters to achieve the best result. The successful model was deemed to be one that had a high True Positive identification and a low False Positive identification based off a threshold ratio. A successful model needed to be cognisant of higher instances of false positive identification as it would result in more human resources being required to investigate flagged cases in future real-life data. To measure the success of each model, an agreed threshold of 1:6 for True Positives to False Positives was used with the highest True Positive below this threshold for each model considered the successful model. This allows for the threshold to be modified depending on the resources available to investigate with a lower threshold providing less True Positives but subsequently a larger drop in False Positives. This analysis identified Random Forest was the most successful model based off these criteria with Extreme Gradient Boosting also performing well and Support Vector Machine performing poorly. The Random Forest model also performed better than the Ensemble of all three models.

where does this magic number come from?

Table of Contents

Table of Figures

Table of Tables

Introduction

The car insurance industry in Australia is worth an estimated A$24 billion with approximately 10% of the insurance premiums paid each year being returned through claims to policyholders[1]. Unfortunately, fraud in the industry is rife with some surveys identifying up to 50% of claims being inflated or fabricated[2]. The industry defines fraud as being when the claimant achieves a more favourable outcome than they are entitled to and can include embellishment of the value of the claim, misrepresentation of the situation, the cause of the incident, the extent of damage due to the incident up to complete fabrication of the incident altogether[3]. Most of the insurance fraud is done by otherwise law-abiding citizens who believe the overstating of losses or altering key facts is a victimless crime. Unfortunately, in doing so, the Insurance Council of Australia claims premiums are increased by approximately $70 each year to account for these fraudulent activities[4].

There is currently no legislation specifying insurance fraud as a specific offence[5]. This coupled with the disparate nature of insurance claimants and the range of fraudulent activities that can be undertaken makes policing in this area difficult. As such the identification and investigation of fraudulent activities is generally left to the industry to self-manage. In line with this, our study has used a publicly available car insurance dataset which has both fraudulent and non-fraudulent entries to showcase the application of various machine learning techniques for identifying fraudulent activities on unknown data.

Manual identification of fraud is cumbersome and although the cost saving through the non-payment of incorrect claims can justify the expenditure of human resources, external audits and expert inspections; it is our belief that these resources can be better spent investigating flagged cases rather than in the effort of finding them.

The impact of a successful fraud detection model will reduce the premium costs as there will be less buffer needed to account for fraudulent activities, as well as reduce the time, effort and manpower required to identify fraudulent claims. Thus, benefiting both the insurance providers and policyholders.

Fraud detection has been extensively studied in machine learning. Minastireanu et al. have surveyed and categorized different ML methods based on type of methods and techniques[6]. Bouzgarne et al. have studied application of ML

[1] Tony Baldock, *Insurance Fraud*, Australian Institute of Criminology, url: https://www.aic.gov.au/sites/default/files/2020-05/tandi066.pdf

[2] Insurance Council of Australia, *Insurance Fraud*, url: https://insurancecouncil.com.au/consumers/insurance-fraud/

[3] T. Baldock.

[4] T. Baldock.

[5] T. Baldock.

[6] Minastireanu, Elena & Mesnita, Gabriela. (2019). An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection. Informatica Economica, 23. 5-16. 10.12948/issn14531305/23.1.2019.01

techniques for detecting insurance frauds and dealing with imbalanced classes[7]. In another research, Prasasti et al. applied machine learning techniques using Synthetic Minority Oversampling method for addressing automobile insurance fraud detection[8]. These papers will act as a starting point for our initial analysis and demonstrate insurance fraud can be efficiently and effectively classified through machine learning techniques.

To train the machine learning models we have access to a sample dataset consisting of just over 11,000 car insurance claims which have already been manually identified and flagged as being either fraudulent or not[9]. Initial exploration of the dataset, covered in detail in the next section: Exploratory Data Analysis, reveals that 6% of the data is classed as fraudulent claims while the remaining 94% are legitimate. Although the sample data is heavily skewed towards non-fraudulent claims this does not prevent the use of the dataset for training of machine learning models. Special consideration will be needed when splitting the data into subsets to retain an adequate representation of both classes which will be covered in the section: Dealing with Unbalanced Data.

The success metrics against which we will measure our models can be validated against the widely available metrics of generalisation success as well as with respect to the real world application we are designing this model for. The dataset lends itself to several thoughts for success. For example, we do not want to have a large number of false positives in our classification as this will increase the number of claims to be manually investigated for fraud, which undermines the requirement of the model to perform better than the current manual identification of fraudulent claims. Our approach in the metrics for determination of success will be covered in the Model Success Criteria section.

While most previous approaches to fraud detection with machine learning focused on implementing a single methodology approach and comparing them against each other with a variety of tuning and boosting methods. Our approach will expand on the single methodology by training 3 independent machine learning algorithms and then using an Ensemble method to combine the results. Our aim is that the ensembled models will produce a better generalisation result than each of the independent models.

Exploratory Data Analysis

The dataset used for this project was obtained from a Kaggle repository[10] and contains 11,566 records of car insurance claim information. The data is recorded in 34 columns which consist of 33 features and one label column

[7] Bouzgarne Itri, Youssfi Mohamed, Bouattane Omar and Qbadou Mohamed, "Empirical Oversampling Threshold Strategy for Machine Learning Performance Optimisation in Insurance Fraud Detection" International Journal of Advanced Computer Science and Applications(IJACSA), 11(10), 2020

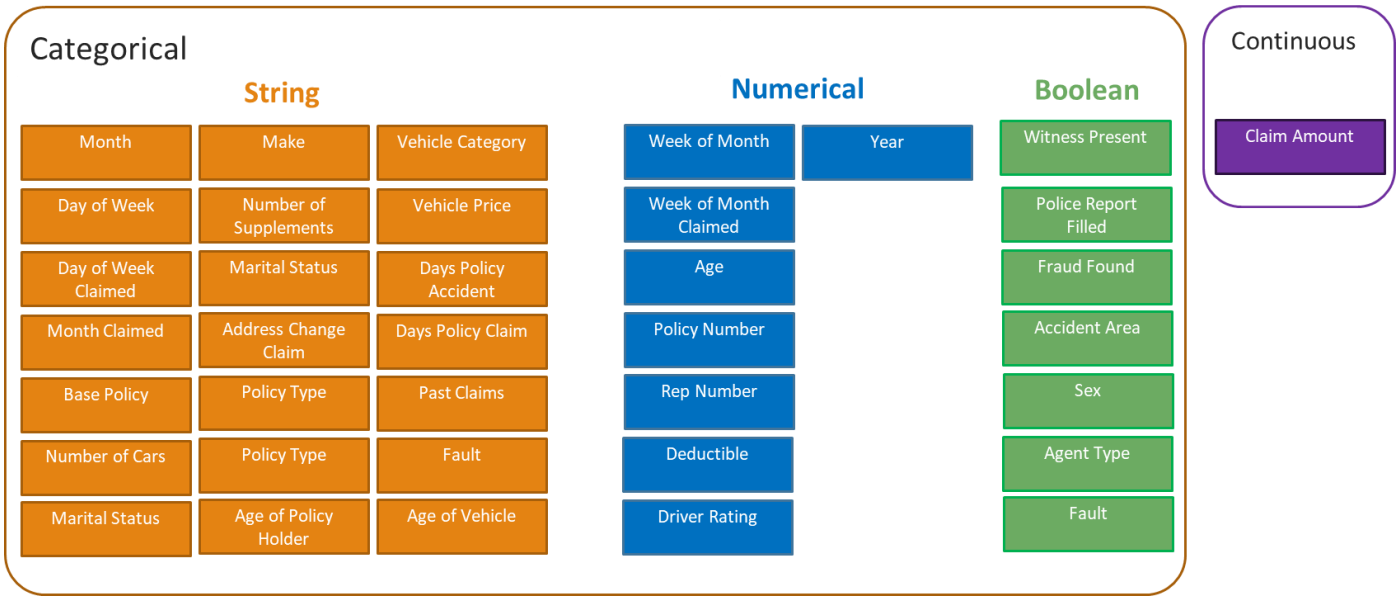
[8] I. M. Nur Prasasti, A. Dhini and E. Laoh, "Automobile Insurance Fraud Detection using Supervised Classifiers," *2020 International Workshop on Big Data and Information Security (IWBIS)*, 2020, pp. 47-52, doi: 10.1109/IWBIS50925.2020.9255426

[9] Dataset Used: https://www.kaggle.com/datasets/incarnyx/car-insurance-fraud

[10] Dataset Used: https://www.kaggle.com/datasets/incarnyx/car-insurance-fraud

identifying each car insurance claim as fraudulent or not. The features of the dataset in the raw form are shown in Figure 1.

Figure 1. Dataset Features - Raw Data



All columns containing string values were originally given the “object” datatype upon importing the dataset. For ease of future analysis, these columns were converted to “string” datatype. Table 1 shows the datatypes as well as a sample record of each of the columns. For columns with only 2 unique values, the options for the records are listed.

Table 1. Column Descriptions

Column	Description	Sample Record	Number of Unique Records	
Month	String	Dec	12	
Week of Month	Integer	3	5	
Day of Week	String	Wednesday	7	
Make	String	Honda	19	
Accident Area	String – Boolean	Urban	2	Urban or Rural
Day of Week Claimed	String	Tuesday	8	Includes '0'
Month Claimed	String	Jan	13	Includes '0'
Week of Month Claimed	Integer	4	5	
Sex	String – Boolean	Male	2	Male or Female
Marital Status	String	Single	4	
Age	Float	41	66	
Fault	String – Boolean	Third Party	2	Policy Holder or Third Party
Policy Type	String	Sport – All Perils	9	
Vehicle Category	String	Utility	3	
Vehicle Price	String – categorical Binned data	20000 to 29000	6	
Fraud Found	Integer – Boolean	1	2	1 – Fraud, 0 – Not Fraud
Policy Number	Integer – unique identifier	209	11565	
Rep Numerical	Integer	13	16	
Deductible	Integer	400	4	
Driver Rating	Float	3	4	
Days Policy Accident	String – categorical Binned data	More than 30	5	
Days Policy Claim	String – categorical Binned data	15 to 30	4	
Past Number of Claims	String – categorical Binned data	2 to 4	4	
Age of Vehicle	String – categorical Binned data	4 years	8	
Age of Policy Holder	String – categorical Binned data	26 to 30	9	
Police Report Filed	String – Boolean	No	2	Yes or No
Witness Present	String – Boolean	No	2	Yes or No
Agent Type	String – Boolean	External	2	External or Internal
Numerical of Supplements	String – categorical Binned data	1 to 2	4	
Address Change Claim	String – categorical Binned data	4 to 8 years	5	
Numerical of Cars	String – categorical Binned data	3 to 4	5	
Year	Integer	1994	3	
Base Policy	String	Liability	3	
Claim Size	Float – continuous variable	11802.52	3332	

Most of the features are string types as well as categorical values. This implies that when training models on this dataset it needs to be noted that test data should consist of the same categories as the training set, as unseen categories will result in poor generalisation performance.

Initial Data Analysis

Initial exploration of the dataset reveals that 5.92% of the data is classed as fraudulent claims while the remaining 94.08% are legitimate. This distribution of the data indicates that suitable methods for skewed data analysis should be implemented when working with the dataset and building machine learning models.

The dataset also revealed a small number of missing values. There are 11 incomplete records in the entire dataset, which represent 0.1% of the dataset. Each of the 11 incomplete records only have one missing feature out of the 33. 5 of the records have the age of the driver missing (column name “Age”) while 6 of the records have the driver rating missing (column name “DriverRating”). Given that the missing data represents a negligible portion of the dataset, imputation will not be conducted and the incomplete records will simply be removed and not included in any future analysis.

In addition to the missing values, the initial exploration identified 2 types of inconsistent entries:

1. There is 1 record where both the month and day claimed features are zero
2. There are 253 records where the age feature is zero.

The singular record where the features of month and day claimed are zero was removed from the dataset. All records which will be used for training machine learning models will now have valid months and days entered for the columns of month and day claimed. And the number of unique values for these columns shown in Table 1 is reduced to 12 and 7 respectively.

For all records where the “Age” feature has a value of zero, it was discovered that that the “Age of Policy Holder” feature has a value of “16 to 17” for these records. Therefore, these records were imputed and assigned a value of 16. This imputation is not perfect as there is often a discrepancy in the dataset between the “Age” and the “Age of Policy Holder” features, where the value of the “Age” feature is outside of the range indicated by the ”Age of Policy Holder” feature (for example: All records where “Age” is 26, “Age of Policy Holder” has a value of “31 to 35”). However, with the assumption that no person under 16 would be able to drive a car and using the “Age of Policy Holder” as a guide, the records where “Age” was zero were imputed and assigned a value of 16 as the lowest age range of the “Age of Policy Holder” bin.

Once the missing records were removed (12 records removed in total) and the “Age” records were imputed as described above, the clean dataset resulted in a split of 5.93% of the records labelled as fraudulent and 94.07% as legitimate with 11,553 records in total. Thus, the imputation and removal of missing values did not affect the skewness or the size of the dataset.

Feature Analysis

Histograms were plotted for each of the features in the dataset. Figure 2 shows the distribution of the "Age" feature in both fraudulent and non-fraudulent claims. It was observed that largely the fraudulent claims follow a similar distribution as the legitimate claims. The other features in the dataset demonstrated this pattern as well as can be seen with the histograms of all the features in Appendix B.

Given that most of the features in this dataset are categorical in nature, the only features in which a skewed normal distribution is observed are the "Age" and "Claim Size" features (Age distribution is features in Figure 2 while claim size distribution is found in Appendix B). These distributions are only evident because age has multiple unique values compared to the other categorical features and claim size is the only continuous variable in the dataset.

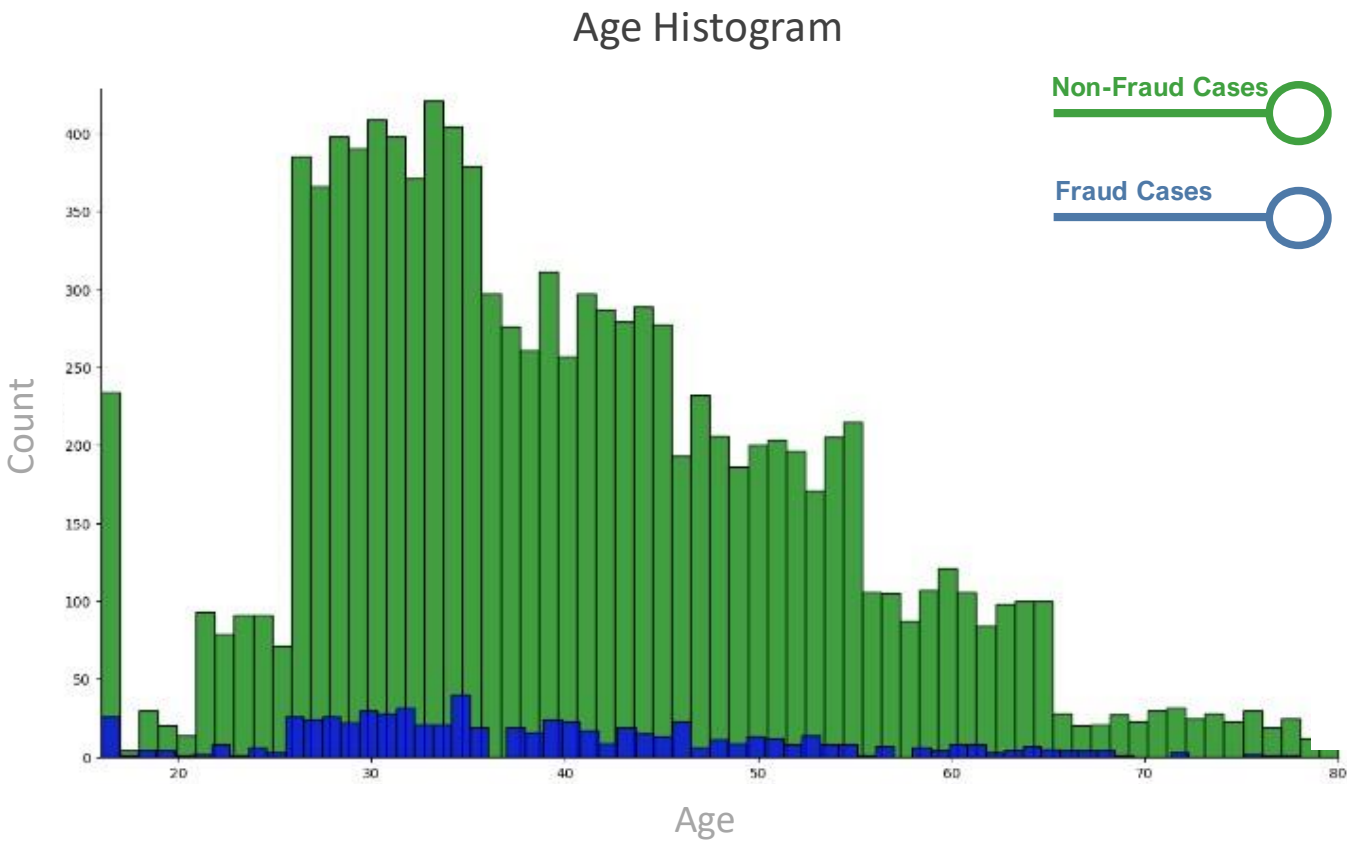


Figure 2. Histogram of Age for Legitimate and Fraudulent Claims

Table 2 shows high level statistics for the numeric features of the dataset. There are no notable anomalies identified in the statistics by feature. The high standard deviation in policy number is expected as this feature is a unique identifier of policies and therefore would not be expected to follow a normal distribution. The low variance and mean for the label "Fraud Found" is also expected since the data is heavily skewed to legitimate claims which are denoted with the numeric value 0 in the dataset.

Table 2. Statistical Description of Numeric Features

	count	mean	std	min	25%	50%	75%	max
WeekOfMonth	11,553	2.78	1.28	1	2	3	4	5
WeekOfMonthClaimed	11,553	2.70	1.26	1	2	3	4	5
FraudFound_P	11,553	0.06	0.24	0	-	-	-	1
PolicyNumber	11,553	7,712	4,453	1	3,854	7,713	11,542	15,420
RepNumber	11,553	8	4.62	1	4	8	12	16
Deductible	11,553	408	43.42	300	400	400	400	700
Year	11,553	1,995	0.80	1,994	1,994	1,995	1,996	1,996
Age	11,553	40	12.74	16	31	38	49	80
DriverRating	11,553	2	1.12	1	1	2	3	4
ClaimSize	11,553	22,966	26,995	0	4,149	8,131	46,490	141,394

Table 3 shows the description of the string columns in the dataset. The most common value, or categorical bin, in the dataset is represented in the “top” column with the number of occurrences in the “frequency” column. There is a strong skewness towards certain features in the data set, for example most of the records have no police reports or witnesses present and the majority of records feature a male driver and are in an urban class accident area. There are other features which have a wider division in the bins, for instance the make of the car, where the most common car being “Pontiac” doesn’t represent a majority of the dataset.

Table 3. Description of String Columns

	count	unique	top	frequency
Month	11,553	12	Jan	1,054
DayOfWeek	11,553	7	Monday	1,995
Make	11,553	19	Pontiac	2,895
AccidentArea	11,553	2	Urban	10,345
DayOfWeekClaimed	11,553	7	Monday	2,808
MonthClaimed	11,553	12	Jan	1,083
Sex	11,553	2	Male	9,758
MaritalStatus	11,553	4	Married	7,982
Fault	11,553	2	Policy Holder	8,385
PolicyType	11,553	9	Sedan - Collision	4,183
VehicleCategory	11,553	3	Sedan	7,246
VehiclePrice	11,553	6	20000 to 29000	6,104
Days_Policy_Accident	11,553	5	more than 30	11,427
Days_Policy_Claim	11,553	3	more than 30	11,493
PastNumberOfClaims	11,553	4	2 to 4	4,135
AgeOfVehicle	11,553	8	7 years	4,386
AgeOfPolicyHolder	11,553	9	31 to 35	4,193
PoliceReportFiled	11,553	2	No	11,224
WitnessPresent	11,553	2	No	11,485
AgentType	11,553	2	External	11,382
NumberOfSuppliments	11,553	4	none	5,324
AddressChange_Claim	11,553	5	no change	10,705
NumberOfCars	11,553	5	1 vehicle	10,687
BasePolicy	11,553	3	Collision	4,469

EDA Conclusions

The dataset, although heavily skewed with respect to the label, does show good potential for machine learning applications. There were very few missing values and features in need of imputation. Additionally, the size of the dataset is sufficient to split into multiple sets. This allows for the employment of multiple machine learning techniques and tuning of models to manage the highly categorical and skewed nature of the data.

The overall similar trending distribution of fraudulent compared to non-fraudulent claims across all features coupled with certain features being more heavily represented in the dataset indicates that it could be possible to reduce the dimensionality of the dataset to simplify the training of machine learning models.

Dealing with Unbalanced Data

The difficulty with this dataset is the number of cases of fraudulent behaviour is significantly smaller than non-fraudulent. This is likely to also be the case for future real unseen data as whilst fraud is common in the industry, fraudulent cases do not represent a majority[11]. Our dataset has only 6% of the data set flagged as fraudulent with this imbalance potentially causing difficulties in prediction with such a small number of samples in the minority class.

With any dataset, stratified sampling could be used to sample the dataset but with the very low occurrences of the minority class we would expect over representation of the same data points in this class against random sampling of the majority class. Although the sample distribution would represent the population, training on these models could show bias towards the majority class and may struggle to identify the minority class in a validation or secondary data set.

To deal with the unbalanced data, other methods could be employed to reduce the imbalance such as under or over sampling. Due to the very low number of fraudulent cases compared to non-fraudulent, the use of under sampling would potentially lose a lot of information. Similarly, with over-sampling this could result in a lot of duplicate data points also creating issues. To overcome the imbalance, synthetic data points were created of the minority fraudulent class using Synthetic Minority Oversampling Technique (SMOTE). This technique allows for the creation of new points rather than duplicates by drawing a random sample from the minority class and identifying the *k* nearest neighbours of these points. From this neighbour a vector is created between the two points and multiplied by another random vector to produce a new synthetic data point.

The use of SMOTE allows for the synthetic datapoints to be created and tested against a range of *k* values. To further enhance SMOTE with minority classes, Tomek-links provides a methodology of reducing the synthetic data that is close to the majority class and could impact training of the model. To achieve this, Tomek-links looks at the Euclidean distance between samples of the majority class and compares it to all samples in the minority class and those that are close to each other are removed. Both methodologies were applied to the dataset and used the Random Forest algorithm with a variety of *k* values to model the data as seen in Figure 3. The same methodology was also applied with the validation set as seen in Figure 4.

[11] T. Baldock.

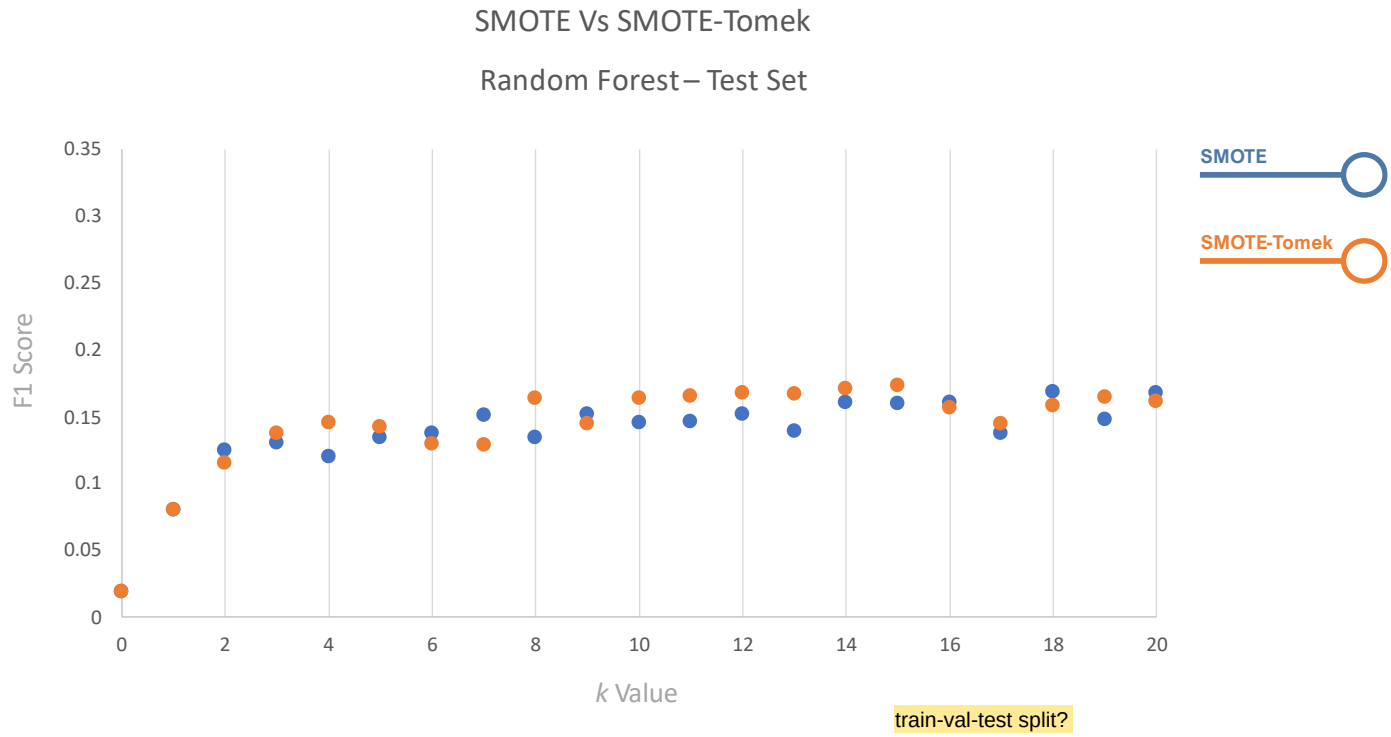


Figure 3. SMOTE Vs SMOTE-Tomek Performance on Test Set Random Forest Model

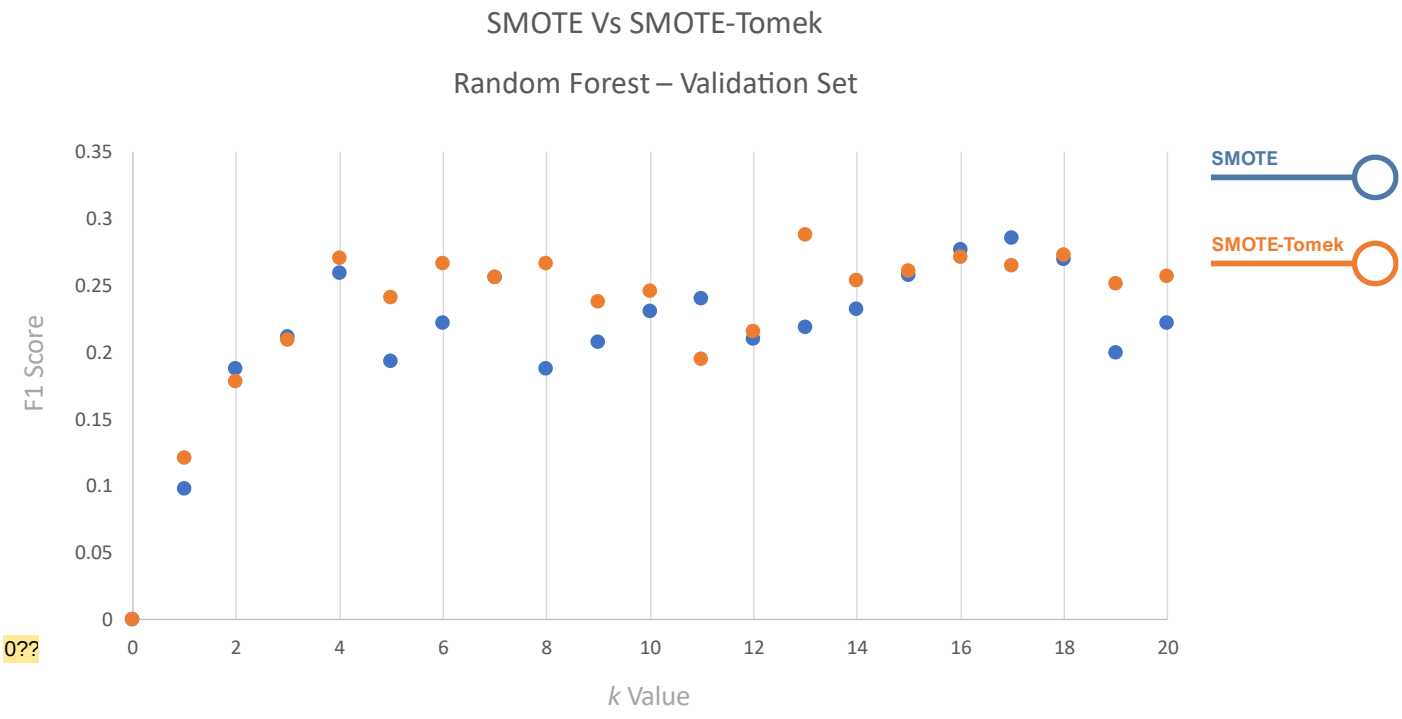


Figure 4. SMOTE Vs SMOTE-Tomek Performance on Validation Set Random Forest Model

It is important to note that the random forest model used to test whether SMOTE and SMOTE with Tomek Links will be beneficial for our data set was not a tuned model and its purpose was solely to see if there is a performance improvement by utilising SMOTE or SMOTE with Tomek links. However, the parameters of this random forest model were identical between the two techniques tested.

From the figures above, both techniques do show an improvement in the F1 Score on the data sets over a range of *k* values. For reference, the *k*=0 data point was used to show the score when the model was run without these techniques. The figures show SMOTE with Tomek Links has a slightly better performance for both the test and validation results.

Due to the variability of the scores achieved over a range of *k* values, a one size fits all approach could not be used and as such the *k* value will be tuned for each of the models covered in this analysis. This will ensure that for each of the models an optimal value for *k* will be chosen along with the tuning of other hyperparameters.

Feature Selection is it needed?

To achieve dimensionality reduction, we implemented a Principal Component Analysis (PCA) as well as an empirical analysis of the features through domain knowledge and from the results of the EDA.

PCA – Principal Component Analysis

PCA is for feature extraction, not feature selection

PCA was conducted on the data set and didn't show any significant reductions by employing this method. The scree plot in Figure 5 shows that 80% of the data is explained by 22 out of 30 features.

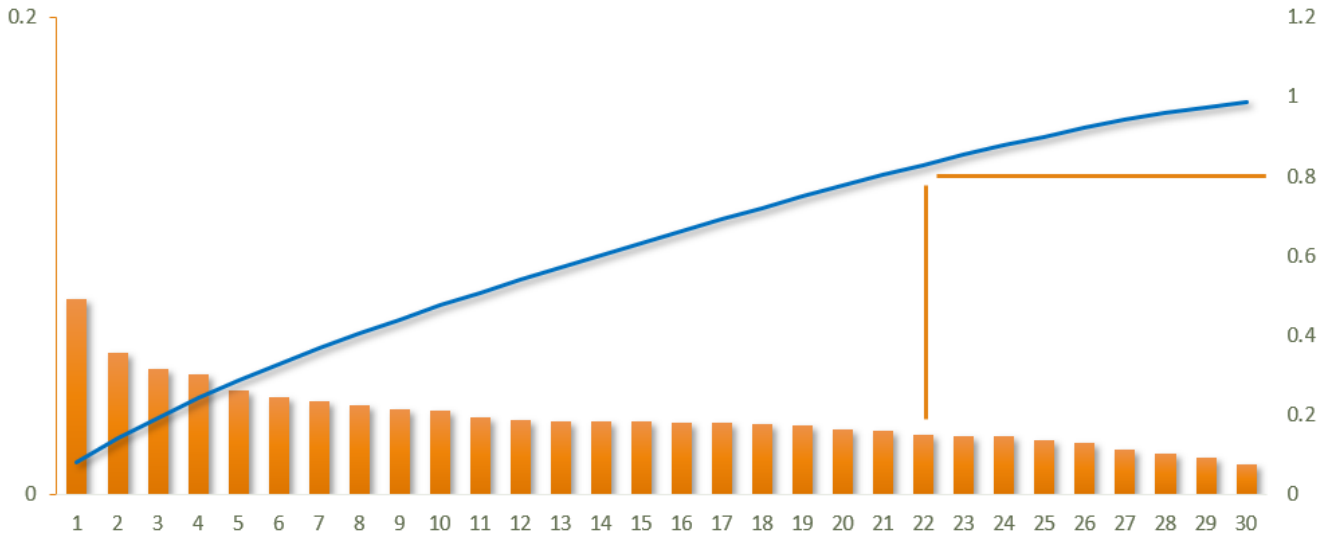


Figure 5. PCA Scree Plot

The large number of features required to achieve 80% meant PCA was not used for the remainder of the analysis. Instead feature selection to reduce dimensionality was employed.

Feature Reduction

Dimensionality reduction was achieved initially through the analysis done in EDA and domain knowledge to identify features which were unlikely to contribute to generalisation. From this, the feature "PolicyNumber" was removed as

it was a unique identifier of each insurance policy and is not related to the nature of a claim. The dataset was already identifying the past number of claims by the same policyholder in the feature "PastNumberOfClaims" further removing the need to keep the "PolicyNumber" feature for model training. The dataset without this feature will be referred to, for the rest of the report as "full dataset".

There were also five features of extremely low variance discovered in the EDA. These features, dubbed as "redacted" were:

- **Police Report Filed** – Where out of the two possible values of "Yes" and "No", "No" represents 97% of the records
- **Witness Present** – Where out of the two possible values of "Yes" and "No", "No" represents 99% of the records
- **Days Policy Accident** – Where out of 5 possible values, "more than 30" represents 99% of the records
- **Days Policy Claim** – Where out of 5 possible values, "more than 30" represents 99% of the records
- **Agent Type** – Where out of two possible values of "External" and "Internal", "External" represents 99% of the records

The dataset without these features (and without the unique identifier) will be referred to, for the rest of the report as "redacted dataset".

There were an additional 11 features that were considered for testing and potential removal, these features are listed below and are referred to in the report as "grey features".

- **Address Change Claim** - 93% of records have a value of "no change", out of 5 possible values
- **Number of Cars** - 93% of records have a value of "1 vehicle", out of 5 possible values
- **Accident Area** - 90% of records have a value of "Urban", out of two possible values of "Urban" and "Rural"
- **Sex**
- **Month**
- **Week of Month**
- **Day of week**
- **Month Claimed**
- **Week of Month Claimed**
- **Day of week Claimed**
- **Year**

Some of these features also have a relatively low variance with a large majority of the dataset belonging to one of a few possible values for the feature. However, since that majority is less than the features in the redacted category, we considered them separately.

The feature "Sex" refers to the gender of the driver with two values of male or female, and while 84% of the records are male, it is not as strong of a skew as some of the other features considered. The issue that we determined with this feature is the possibility of generating a discriminatory machine leaning algorithm if the gender of the driver is a common feature used in prediction of fraud which may rise ethical concerns.

The date and time related features of Month, Week of Month and Day of Week were also included in this list as we do not have the domain knowledge to know if they refer to the date the record has been entered into the dataset, or to the date of the accident. Similarly, for the features Month Claimed, Week of Month Claimed and Day of Week Claimed, we do not know if they refer to the time and date of the claim being paid out or lodged. Therefore, these features could either be relevant or irrelevant to fraud detection and are included in this grey feature list.

The dataset without these features (and without the unique identifier and redacted features) will be referred to, for the rest of the report as "minimal dataset".

Table 4. Datasets for Training Models

Dataset	Description
Full Dataset	Dataset of all the features except the unique identifier "Policy Number"
Redacted Dataset	Dataset without the unique identifier and all the features form the redacted list
Minimal Dataset	Dataset without the unique identifier, redacted list and all the features form the grey area list

Selection of Features for Model Training

Except for the removal of the unique identifier, the feature selection was done empirically resulting in the relevance of certain features being a hypothesis in need of testing. Therefore, all three of the models, Random Forest, Support Vector Machine and Extreme Gradient Boost were trained and tuned on all three of the datasets. The results of the models on these datasets were reviewed before the selection of the dataset was made. The ensemble was then run on the selected dataset.

Data Breakdown

To achieve the analysis as well as test and validate the results, the full data set was broken into a training, test and validation set as per Figure 6, where X is a table of features and y is the label. A balanced data train set was created using SMOTE-Tomek to train the models with the value of *k* being tuned with each model. Once the models were trained, they were tested against the testing set. The validation set showed how well the model worked on unseen data with the same features to ensure the model accuracy and to identify overfitting.

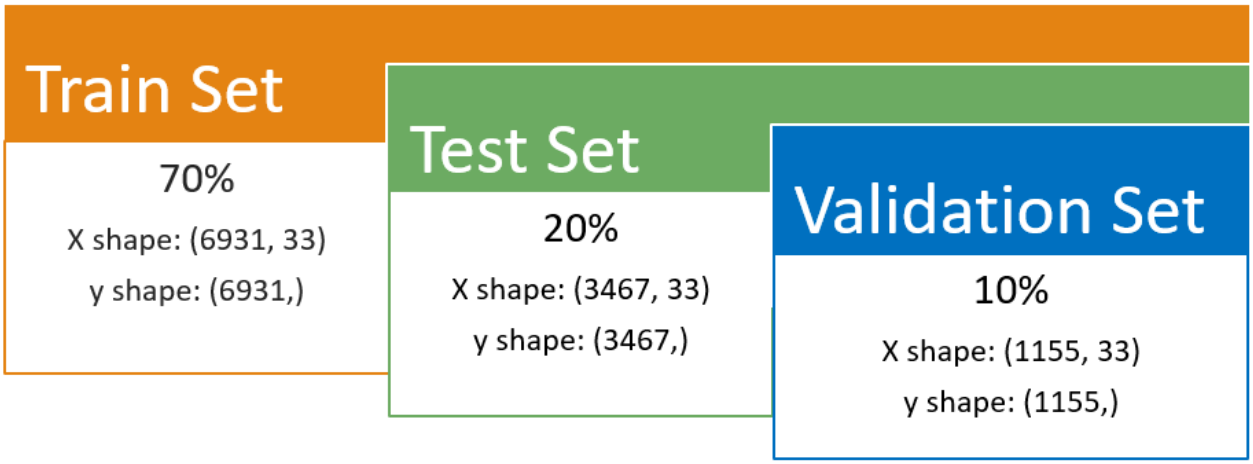


Figure 6. Train, Test and Validation set breakdown

Figure 6 demonstrates the split on the raw dataset before any feature reduction. For each of the three datasets described in Feature Reduction, a split was done on the same percentage values.

When analysing the training, test and validation subsets, we see that the split between classes is still maintained as the per the original dataset, with 94% of records labelled as non-fraudulent and 6% as fraudulent. SMOTE with Tomek Links, as discussed in the Dealing with Unbalanced Data section was applied to the training set for each model.

Model Success Criteria

The successful model will need to achieve the following success metrics:

1. Identify true positives; and
2. Limit the number of false positives.

An increase in the false positive identification would subsequently increase the human resource required to investigate those flagged cases. So, although the true positive identification is the model's aim, the ideal model could

accept a lower true positive identification if the commensurate reduction in the false positive identification was significantly better.

As we are dealing with unbalanced datasets, accuracy cannot be used as a measure as successful prediction. Since even if a model predicts all instances to be non-fraudulent, the accuracy will still be around 94% due to the heavy skewness towards non-fraudulent cases.

The use of an F1 Score or combinations of precision and recall have been found to be also skewed towards a lower true positive identification by also keeping the false positive numbers low. This could be a good metric, but it doesn't allow for modification of the threshold of acceptance between true positive and false positive identification. good

Therefore, to measure the success of the models, a balance where a high number of true positives are predicted while minimising the number of false positives needed to be found. This will allow for the selected model result to have the lowest possible number of flagged records with the highest number of fraudulent cases. The requirement for maximising true positives also has the secondary benefit of minimising false negatives, which in our application, would be fraudulent records which avoid detection.

To achieve identification of this success criteria many different methods were attempted to identify the best values including ratios, gradient ascent along the F1 score and minimum / maximum threshold criteria. These models all had bias towards either higher True Positives or lower False Positives and not achieving the optimum results for both. Whilst the identified points are subjective as to what constitutes a high true positive and a low false positive point as well as what an insurance company is capable of investigating, the use of a threshold ratio was employed to choose the best value of the model. This would have the additional benefit of being able to be adjusted as requirements dictate. The threshold ratio used was 1:6 between true positives and false positives with anything below this ratio deemed an adequate model. The highest true positive value under this threshold would therefore be considered the most successful model.

Machine Learning Models

Three types of machine learning model were trained in the scope of this project. Random Forest (RF), Support Vector Machine (SVM) and Extreme Gradient Boosting (XGB). The models were individually tested and tuned and the best performing models were then combined in an ensemble model. Multiple ensemble techniques were tested and the best one selected.

Although our original proposal only discussed using RF and SVM models, upon further consideration, it was decided to incorporate the XGB model to provide a third, and potentially 'tie breaking' model if required in a majority vote ensemble method.

RF - Random Forest

Random Forests are a common and very powerful machine leaning technique which builds a 'forest' of decision trees in order to predict a value[12]. The application of random forests in classification problems indicates that it is likely to be a good technique to implement on our dataset.

Hyperparameters - Setup and Tuning

We tuned the RF model on each of the three datasets described in the Feature Reduction section. For each of the datasets we tuned the following parameters:

- *k* for SMOTE with Tomek Links in a range of 0 to 15, with 0 not using SMOTE and Tomek links for comparison.
- maximum splitting features *mF* in a range from 1 to the total number of features in the dataset
- maximum depth of the decision trees in the random forest, in a range from 1 to 10 and an unlimited depth.

For all the RF model trained we maintained the number of estimators, that is the number of trees in the forest, to be 100.

Results of Tunning and Hyperparameter Selection

The resulting tuning of the parameters described in the previous section produced:

- 5281 RF models on the full dataset
- 4481 RF models on the redacted dataset
- 2721 RF models on the minimal dataset

[12] Cha Zhang, Tunqian Ma. *Ensemble Machine Learning Methods and Application.* Chapter 5, pages 158-159. Springer Publishing. ISBN: 978-1-4419-9325-0, DOI: 10.1007/978-1-4419-9326-7

For all of these results, the relationship between the true positives and false positives predicted was computed. The results can be seen in Figure 7 for the full dataset, Figure 8 for the redacted dataset, and Figure 9 for the minimal dataset.

A 1:6 linear relationship as described in the model success criteria was plotted in each of the figures to show overall model performance and to provide comparison between models as well as to illustrate the optimum model selection. It is shown that for each of the datasets, the relationship between true and false positives for all models is nearly linear. It is also shown that the redacted dataset has a slightly better performance as the cluster of models in Figure 8 has a lower ratio of true to false positives. Indicating the for the RF models, the redacted dataset was the best fit.

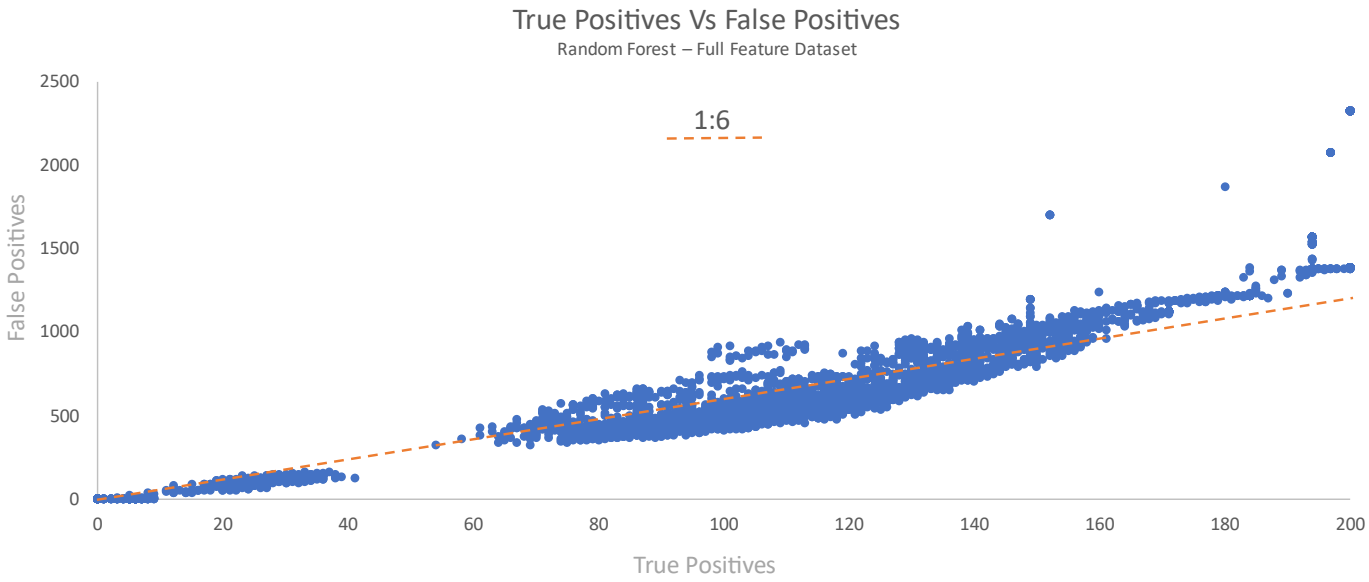


Figure 7. RF Results on Full Dataset

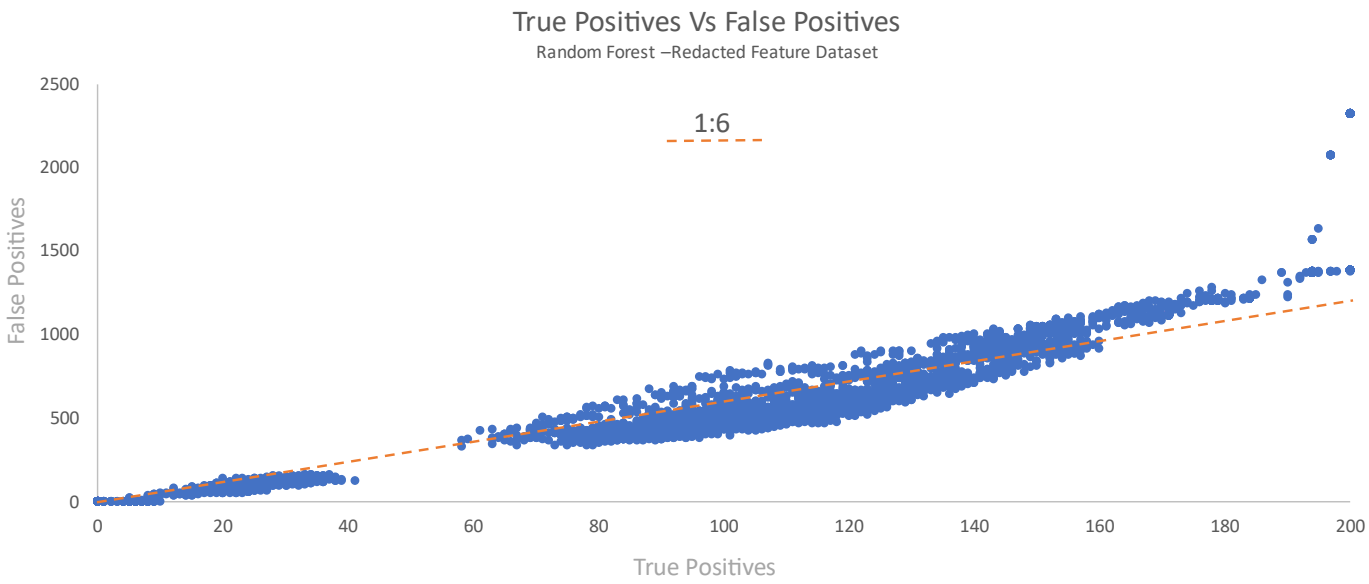


Figure 8. RF Results on Redacted Dataset

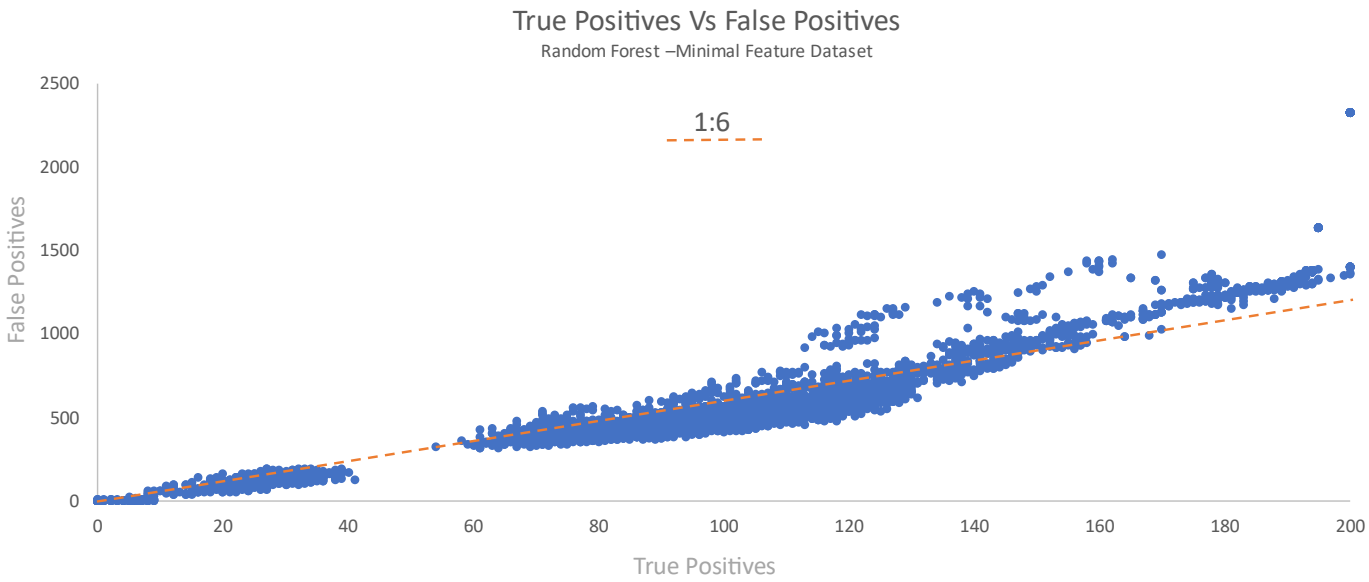


Figure 9. RF Results on Minimal Dataset

Closely reviewing the parameters of the redacted dataset, we selected a model with a *k* value of 4, maximum splitting features of 18 and maximum depth of 4, as seen in Figure 10.

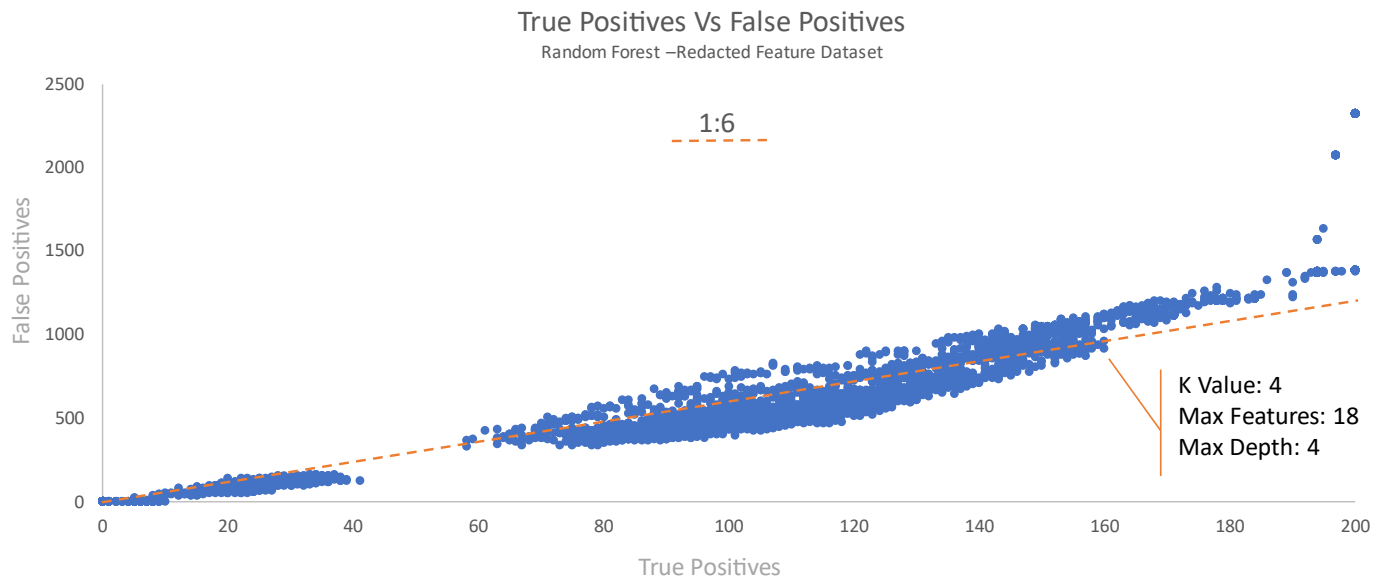


Figure 10. Selected Random Forest Model

These hyperparameters selected produced a model which identified close to 80% of all the true positive cases, while flagging a total of 1079 records as potentially fraudulent which represent a third of the size of the test dataset, thus resulting in the best true positive to false positive ratio while also capturing a significant number of true positives. Moreover, this configuration of hyper parameters also performed similarly on the validation set, indicating a lack of over-fitting. The results on the test and validation sets with the chosen hyperparameters are shown in the Figure 11.

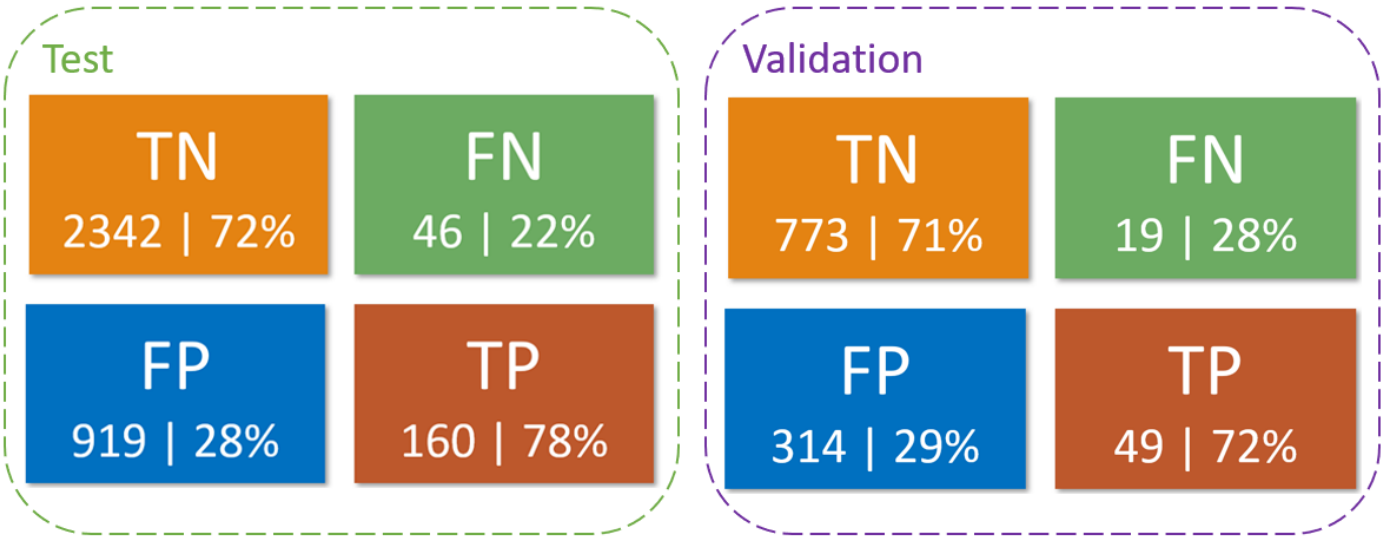


Figure 11. RF Generalisation Performance

ROC Curve

The area under the curve (AUC) in Receiver Operator Characteristic (ROC) gives us an understanding of the trade-off between sensitivity or the True Positive Rate and specificity or 1 – False Positive Rate. By means of comparison, a

random classifier would have a True Positive Rate equal to the False Positive Rate and would therefore give a diagonal line bisecting the graph through the origin on a ROC. The RF ROC Curve in Figure 12 was conducted on the successful model using the tuned parameters from the analysis which gave an AUC of 0.819 showing a good result for the successful model.

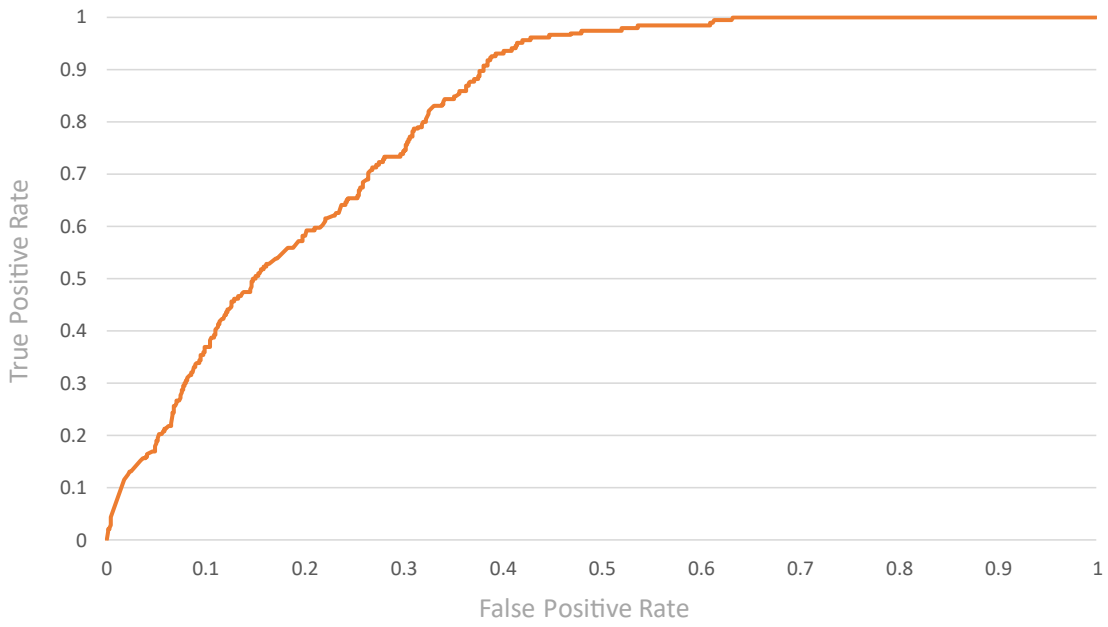


Figure 12. RF ROC CURVE

Model Run Time

As another metric for the analysis, we looked at the time required for the model to fit the data. As more data points come in, the model would have to be re-calibrated and as such a long running model would produce problems in a real-life application. The tuned values for the Random Forest model showed a run time of 1.2 seconds[13] on the full dataset.

Decision Tree Visualisation

To identify where some of the splits were occurring in the random forest, single decision trees were pulled from the forest as can be seen in Figure 13 and Figure 14. With the data set being shown as a single decision tree it highlights the categorical nature of the data and how the decisions are being formed. This view shows the importance of the date features in making the decision splits even if the splits are not being done symmetrically.

[13] All models ran on a PC desktop computer with an Intel i9-7940X CPU @ 3.10GHz 1.99 GHz and 64GB or RAM

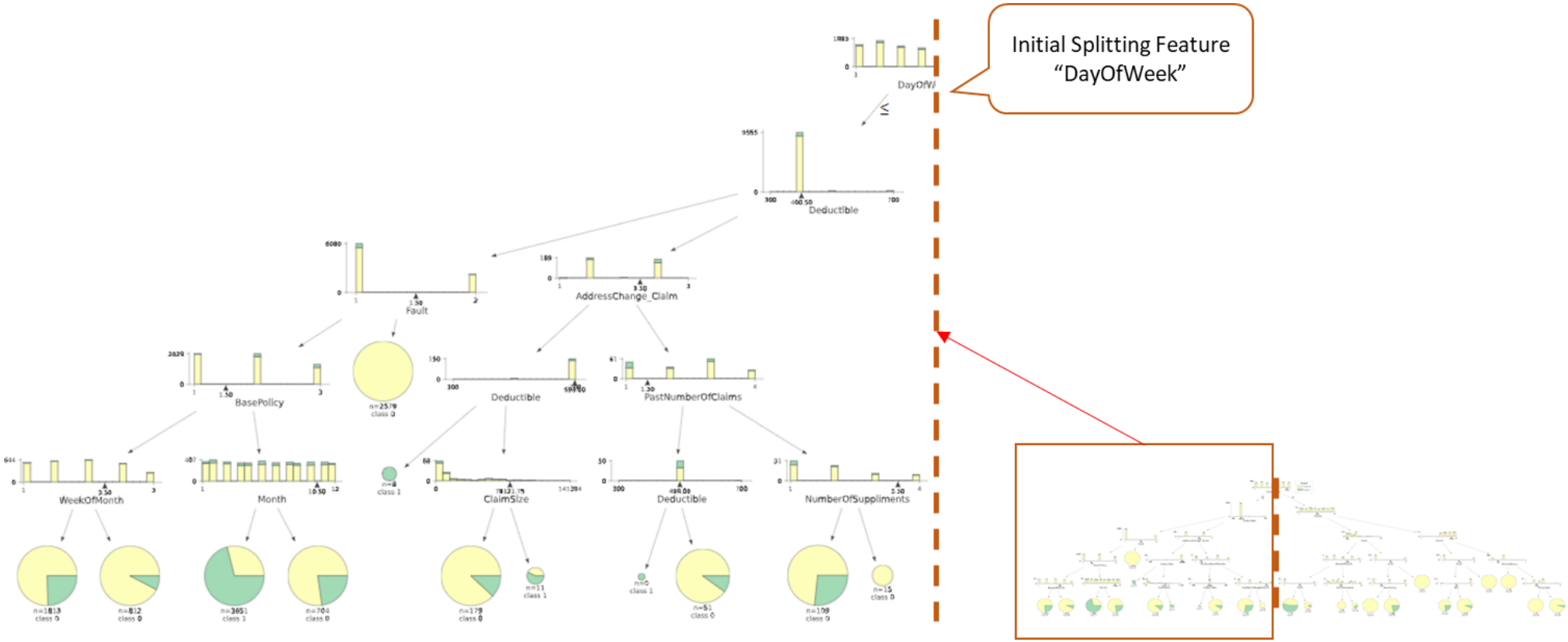


Figure 13. Left side of a Decision Tree from the Random Forest Model

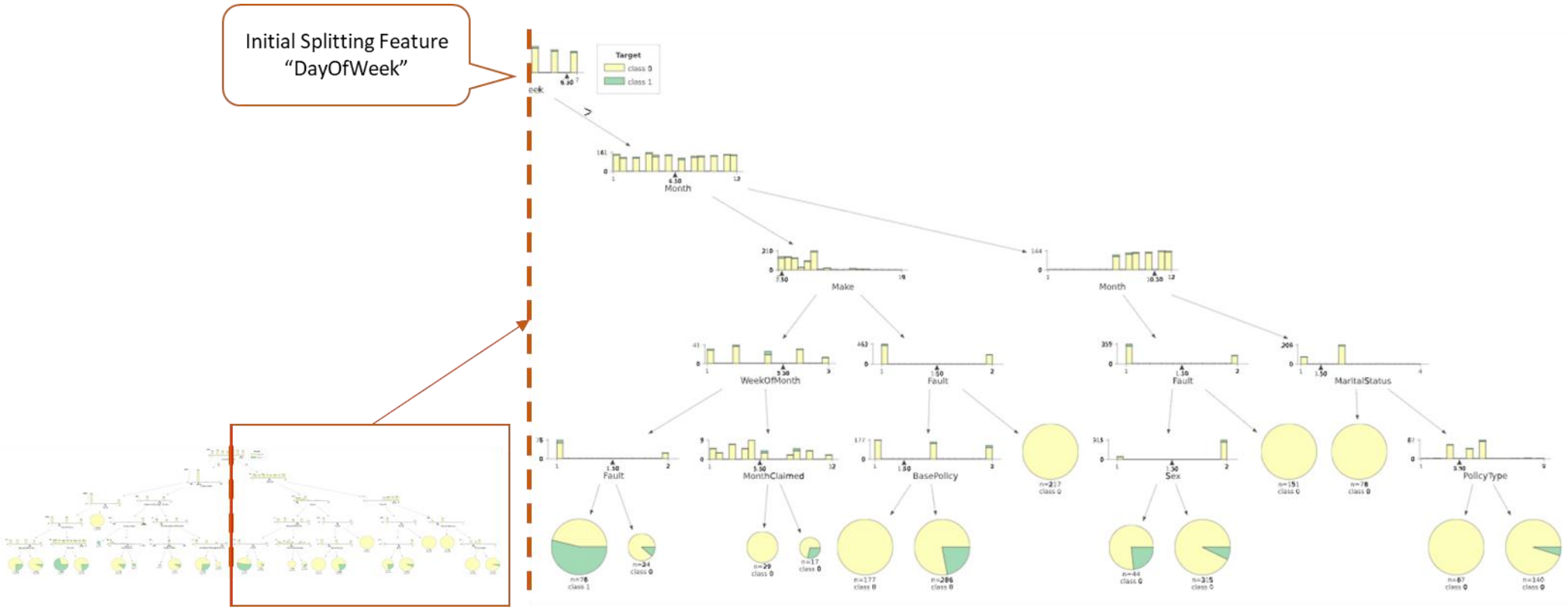


Figure 14. Right side of a Decision Tree form the Random Forest Model

SVM – Support Vector Machine

A support vector machine (SVM) often also referred to as a kernel machine is another classification machine learning technique that was implemented on the dataset.

Different to the tree based model of RF, SVM's algorithm attempts to produce a separation hyperplane between the classes to separate the data. The hyperplane is optimised to be the furthest away from any points while separating the data into the classification regions. This distance to the closest point away from the hyperplane is referred to as the margin. The hyperplane, which can span across many dimensions to classify the data, is guided by a kernel function passed to the model[14]. The additional use of a cost parameter allows the model to penalise points that fall within the margins of the model on either side of this hyperplane.

Set – Up and Tuning of Hyperparameters

As with Random Forest, the SVM model was tuned on each of the three datasets described in the Feature Reduction section. For each of the datasets the following parameters were iterated:

- *k* for SMOTE with Tomek Links in a range of 0 to 15, with 0 being equivalent to not using SMOTE and Tomek links
- C value, denoting cost in the range of 1 to 10.

The 3 main SVM kernels were applied to the redacted data set being:

1. Linear
2. Polynomial (Degrees 3 and 8)
3. Radial Based Function (RBF)

The linear and polynomial (degree 8) models took a lot of computational time to achieve a fit and provided similar ROC AUC results to a polynomial degree 3 and RBF which were faster to fit but still computationally slow when compared to RF. The RBF kernel was selected for further analysis due to being slightly faster.

Results of Tunning and Hyperparameter Selection

The results of tuning of the parameters described in the previous section produced 136 SVM models for each of the 3 datasets giving a total of 408 SVM models. The performance for SVM for all three datasets were similar. Figure 15, Figure 16 and Figure 17 represent the results of the parameter tuning of the SVM models on the full, redacted, and minimal dataset respectively.

Due to the highly similar performance and to remain consistent, we selected the redacted dataset for the SVM model analysis.

[14] Bruno Stacanella. *Support Vector Machines (SVM) Algorithm Explained.* MonkeyLearn Blog. June 23rd 2017. url: https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm

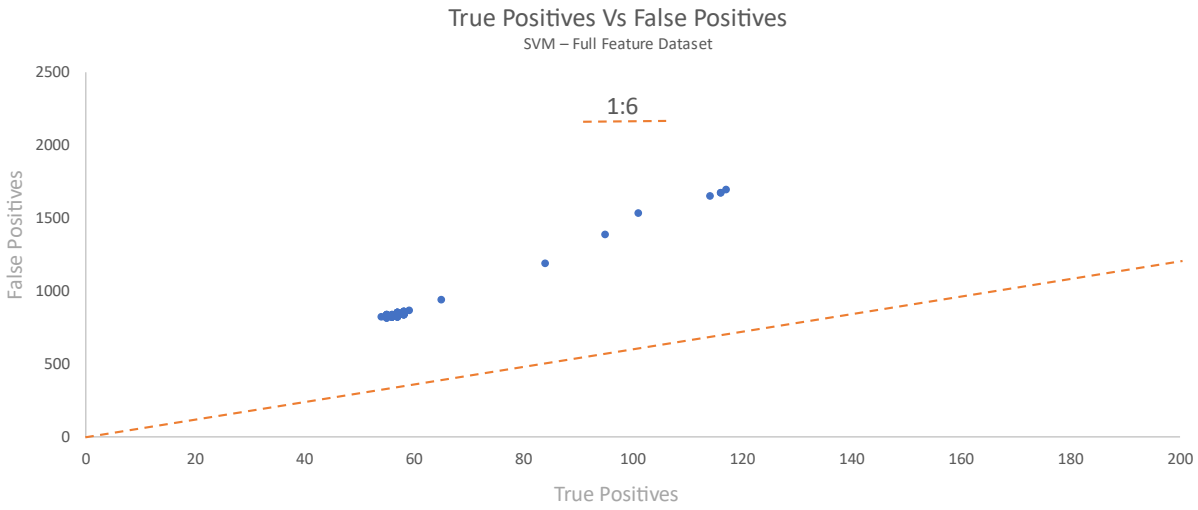


Figure 15. SVM Full Feature Set

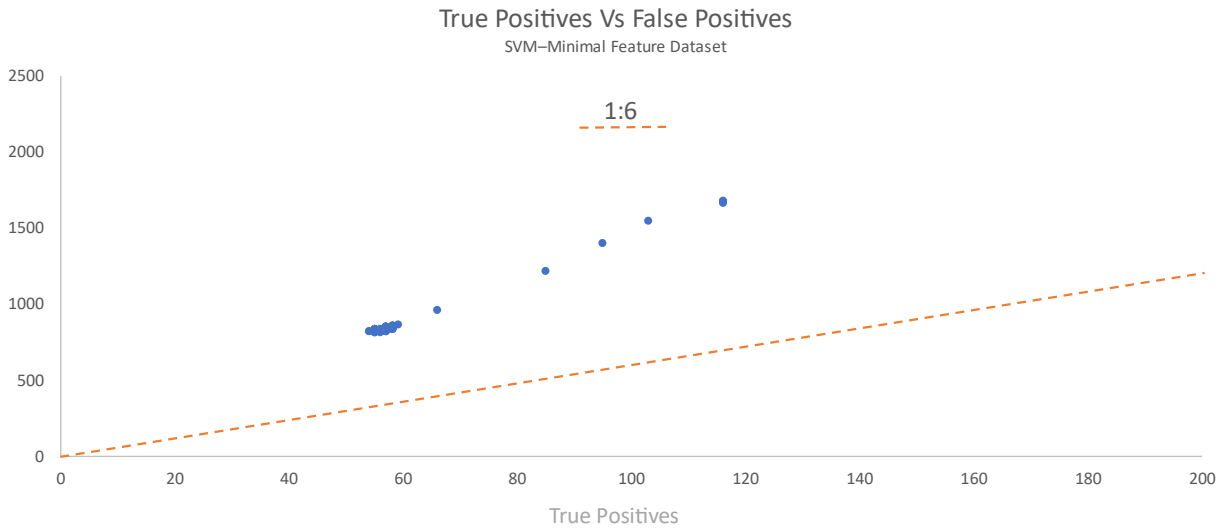


Figure 16. SVM Minimal data set

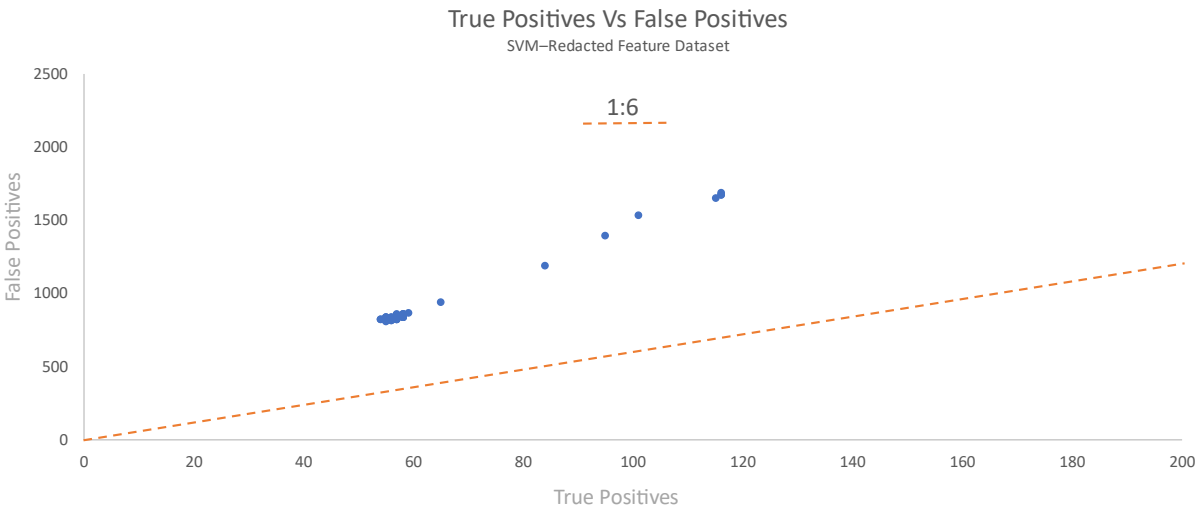


Figure 17. SVM Redacted data set

Figure 18 shows the selected SVM model with the cost of 2 and *k* of 6 being the best result obtained using this method. The selected model lies along the bottom edge of the cluster, which means a better true positive to false positive ratio than models higher up in the cluster.

While there are models that predicted more true positives, seen in the top right of the figure, they come at an expense of a very large number of false positives such that well over a third of the test set was predicted as positive, which would not meet our success metrics.

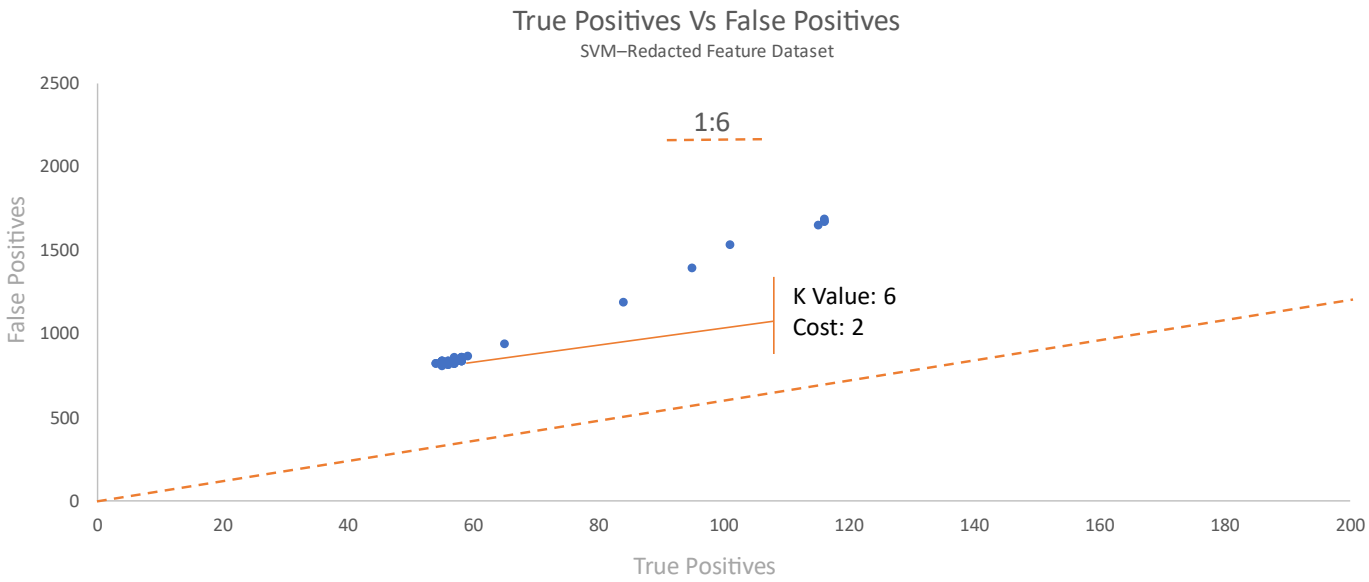


Figure 18. Selected SVM Model

Gamma was also tested in a range of 0.001 to 100 but did not show any improvements in the model so was removed as a parameter. Gamma denotes the influence of a single training example and can be seen as the inverse of the radius of influence. With Gamma not having an effect suggests the model was not able to achieve a sufficient decision boundary around the two classes.

Overall SVM produced a poor generalisation of the test and validation dataset compared to the RF model. With all options in the tuned models having a higher ratio than 1:6 between true positives and false positives. Figure 19 shows the performance of the selected SVM model on the test and validation datasets.

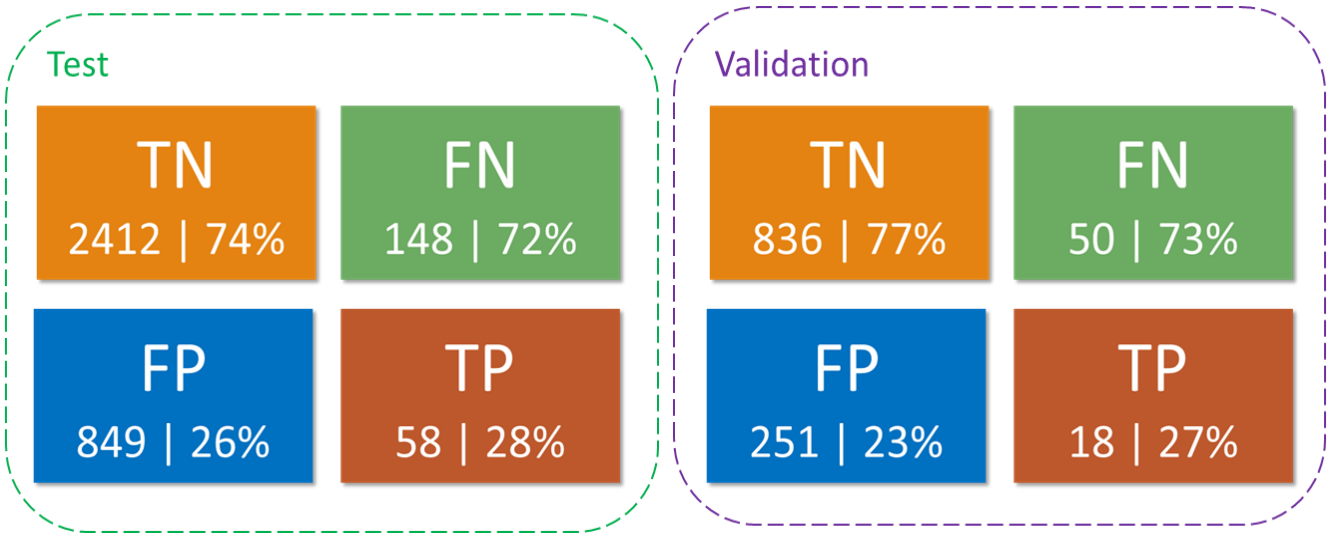


Figure 19. SVM Generalisation Performance

ROC Curve

The selected SVM model gave a ROC AUC of 0.483 on the test data as per Figure 20. This highlights the under fitting of the model and corroborates the issues we had with gamma with identifying a decision boundary. The problem with fitting this model is thought to be due to the categorical nature of this data set.

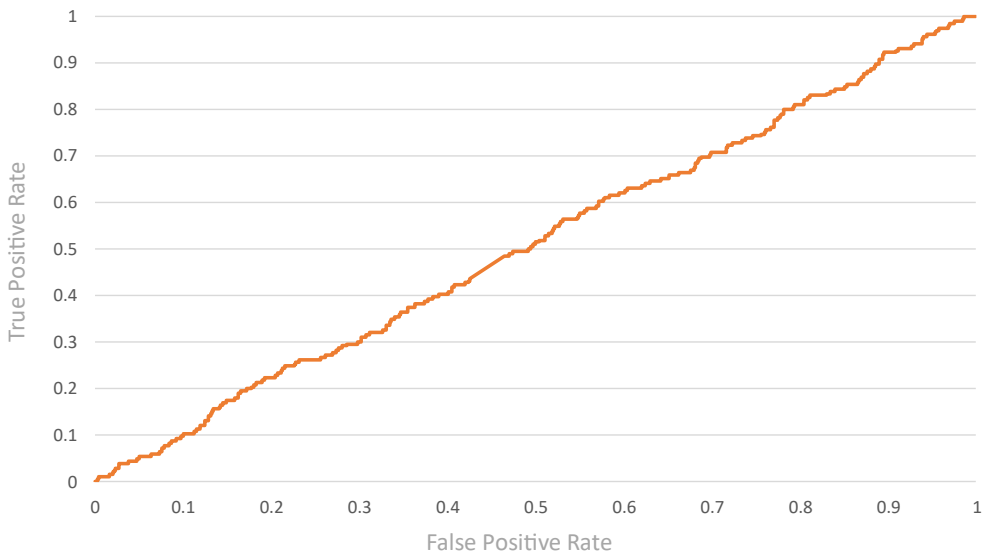


Figure 20. SVM ROC Curve

Model Run Time

The tuned values for the Support Vector Machine model showed a run time of 77.2 seconds on our dataset, which was over 55 times the computational time it took to train the RF model. SVM was the most time-consuming model we trained in the scope of this project and gave the poorest performance.

XGB – Extreme Gradient Boosting

Gradient Boosting is a tree-based decision model similar to RF and is also used in ensembles for classification and regression models to improve results. This is achieved through the adding of trees to the ensemble which are fitted based off their prediction error.[15] eXteme Gradient Boosting (XGB) is an open-source implementation of gradient boosting and can provide a very powerful and fast implementation of the methodology compared to RF and Adaboost methods.

Hyperparameters - Setup and Tuning

We tuned the XGB model on each of the three datasets described in the Feature Reduction section. For each of the datasets we tuned the following parameters:

- *k* for SMOTE with Tomek Links in a range of 0 to 15, with 0 being equivalent to not using SMOTE and Tomek links
- Learning Rate, in a range of 0.1 to 4. With higher learning rates not producing good results.
- Gamma, this controls the minimum loss reduction required to split on a leaf node of the tree. The higher the value of gamma the more conservative the algorithm will be. This was in the range of 1 to 5.
- Number of Estimators, the number of decision trees the model used, in a range of 1 to 100 in increments of 20.
- Maximum Depth, this controls how deep the tree goes, this was tested in a range of 2 to 20 in increments of 2.

Results of Tunning and Hyperparameter Selection

The resulting tuning of the parameters described in the previous section produced 23,041 XGB models for each of the 3 datasets in Figure 21, Figure 22 and Figure 23, for a total of 69,123 XGB models. Using the 1:6 linear relationship as comparison and model selection it can be seen there is a band of optimum results below the threshold with low False Positives and higher True Positives.

[15] Jason Brownless, https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/, Nov 2020

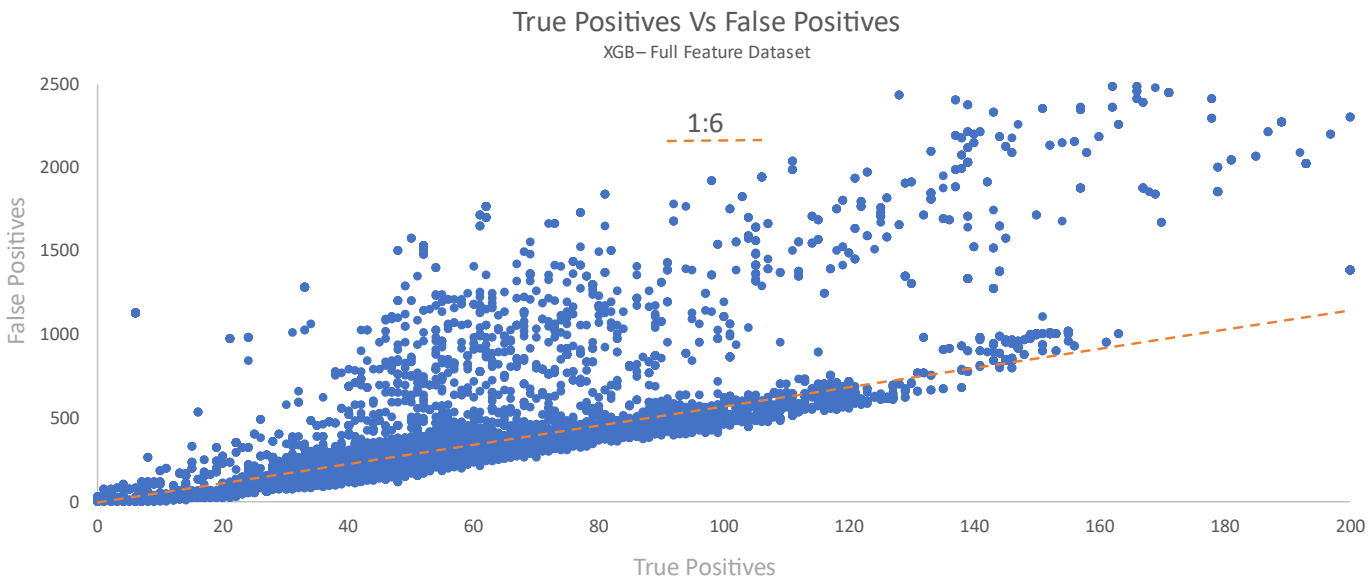


Figure 21. XGB Results on Full Dataset

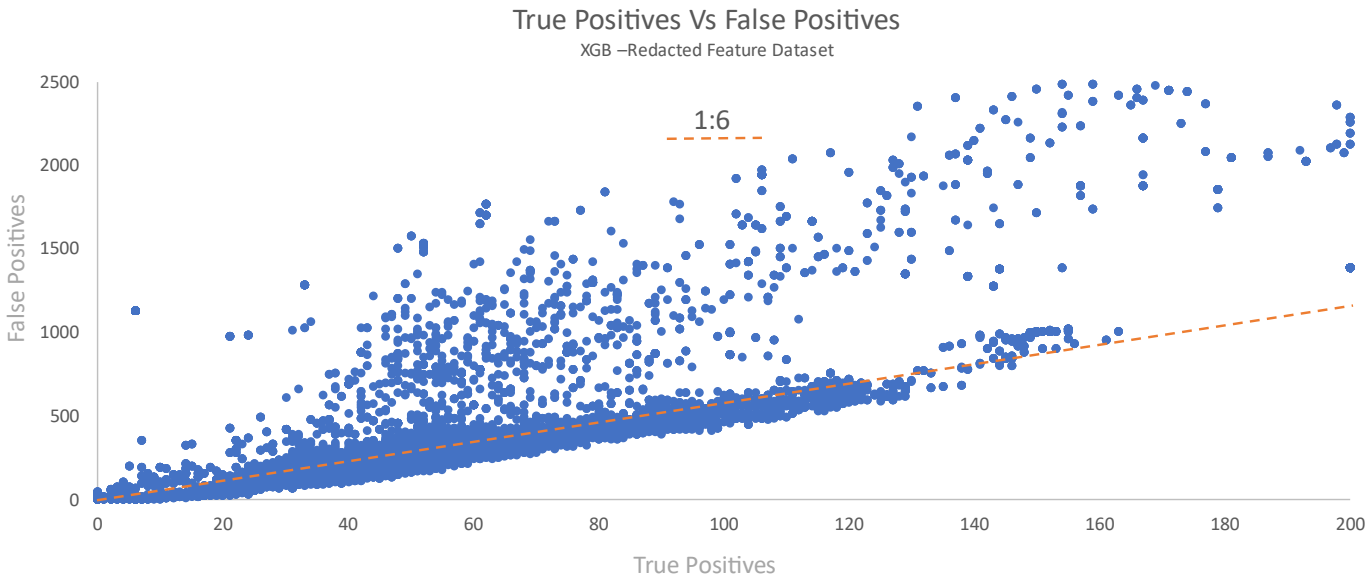


Figure 22. XGB Results on Redacted Dataset

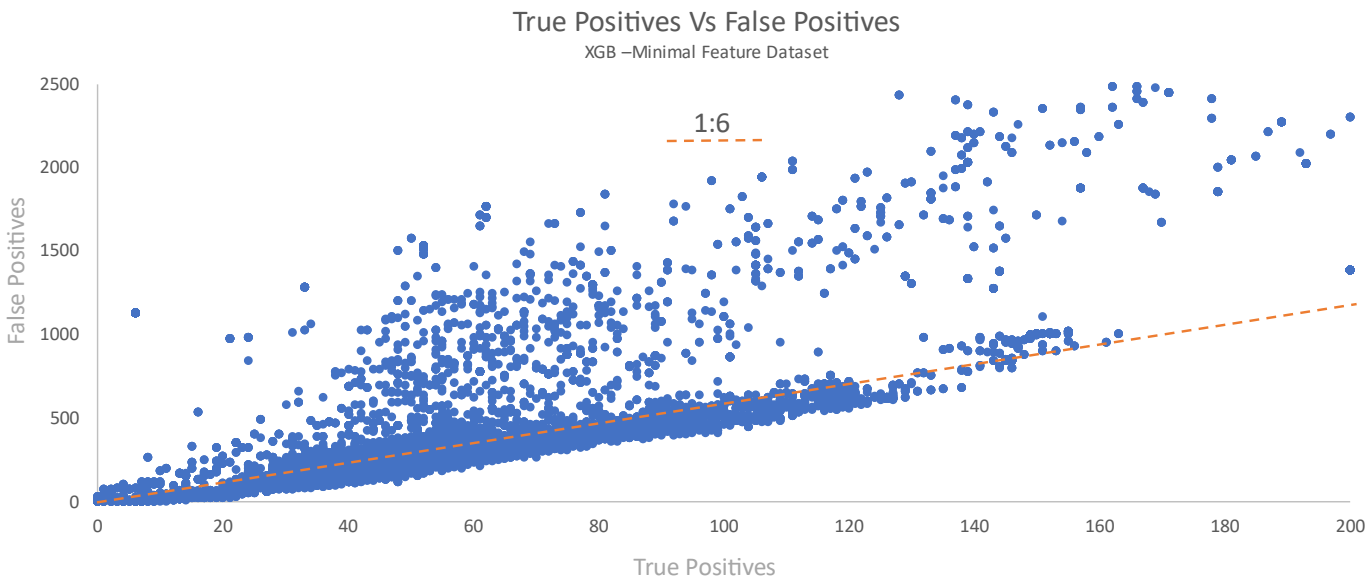


Figure 23. XGB Results on Minimal Dataset

The models again did not produce any significant differences below the threshold between the data sets. As such the redacted data set was used although the same successful model point was the same in all 3 datasets. The selected data point model had a *k* value of 4, maximum splitting features of 18 and maximum depth of 4, as seen in Figure 24.

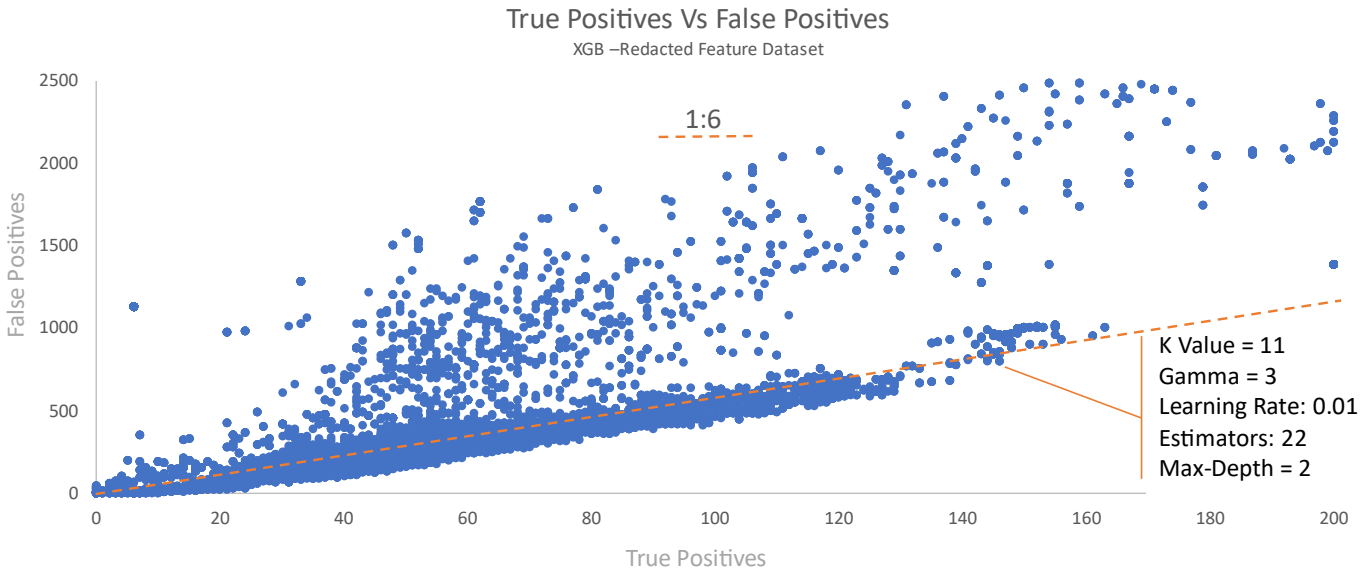


Figure 24. Selected XGB Model

The hyperparameters selected produced a model which identified 71% of all the true positive cases, while flagging a total of 1104 records (32% of the total test dataset) as potentially fraudulent. This was also a similar result on the validation set confirming the model as shown in Figure 25.

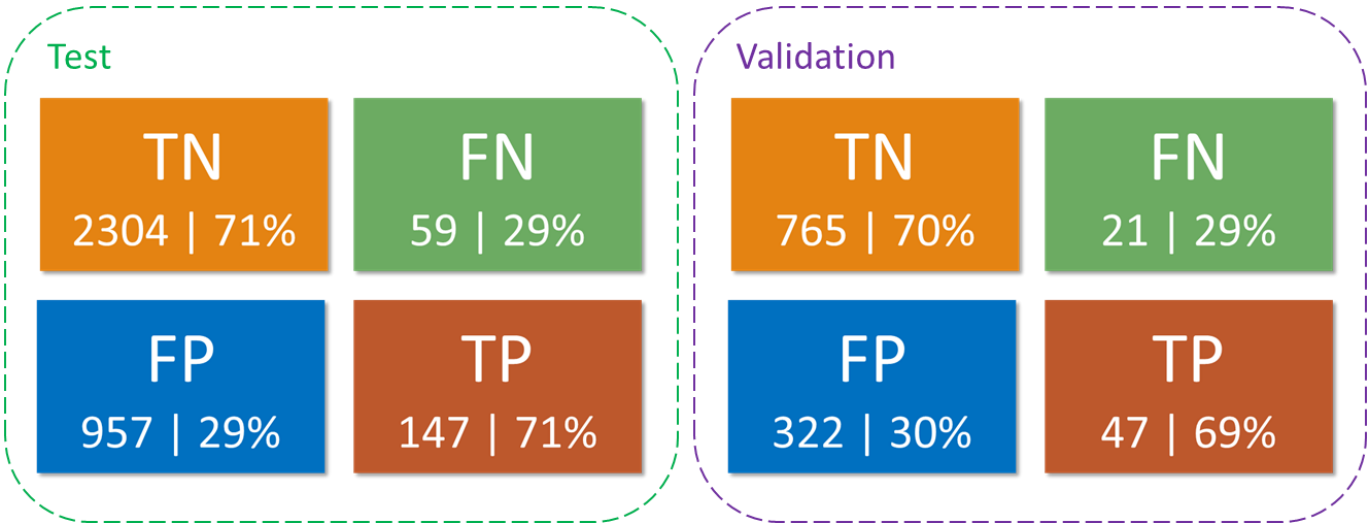


Figure 25. XGB Generalisation Performance

ROC Curve

The successful XGB model ROC AUC in Figure 26 shows similar behaviour to RF with an AUC of 0.793.

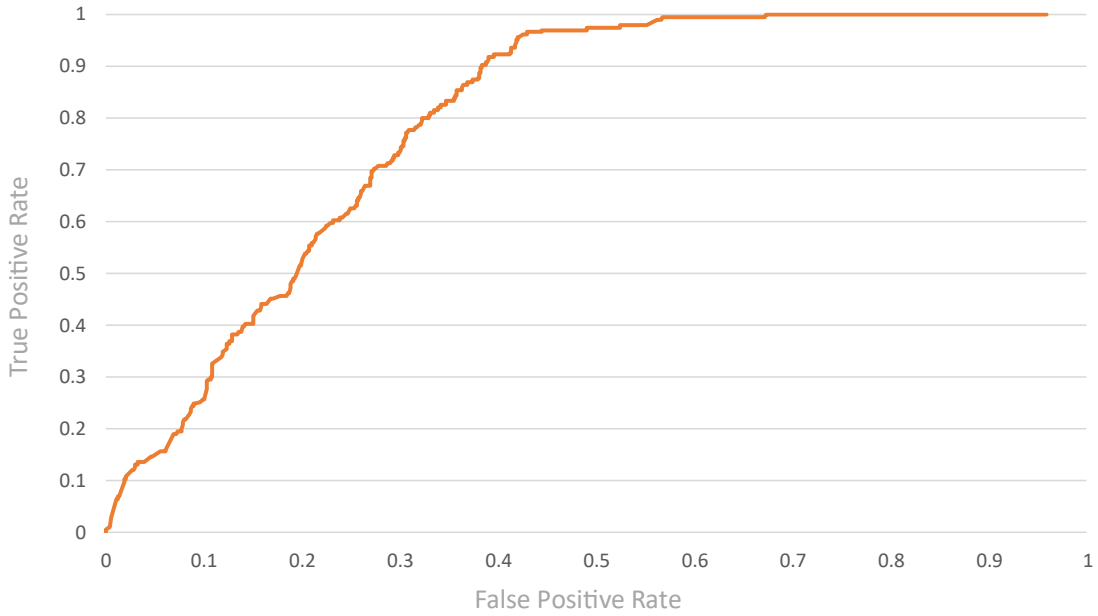


Figure 26. XGB ROC Curve

Model Run Time

The tuned values for the XGB model had a run time of 0.25 seconds on our dataset, proving to be the fastest out of the three trialled models.

Ensemble

Ensemble learning is a technique which combines multiple machine learning models into one[16]. Random Forest and XGB are, in fact also ensemble methods though we treated them as components for a larger ensemble in this project.

The ensemble consisted of the three selected machine learning models described previously. We explored different methods of combining the results of each model. Firstly, we attempted to utilise a stacked classifier method. This method would have allowed us to use the output of individual classifiers as inputs for the final estimator[17]. Unfortunately, this method proved not to be applicable for this project. To stack classifiers, each individual model in the ensemble is required to be trained on the same dataset. In our project, since every model used a different *k* value for SMOTE with Tomek links, the training dataset was not identical between each model.
The option of making all the training sets identical just to use the stacking classifier method in the ensemble was considered. However, given that *k* was tuned for each model to optimise its performance, and all the *k's* are different (4 for RF, 6 for SVM, 11 for XGB) setting a common *k* for all three models would jeopardise their individual performance.

Once stacking classifiers was ruled out as a method, two simple linear approaches were considered. One of majority vote and one of assigning weights to results of different models within the classifiers.

Majority Vote

This method is straight forward. Whatever the majority prediction for the instance of the three models in the ensemble, that value becomes the ensembled prediction. Therefore, if two out of the three models predict a data point as fraudulent, then the ensembled prediction is also fraudulent.

Weighed Predictions

The weighed approach was a build on the majority vote approach. In addition to classing a record as fraudulent when at least 2 models predicted it to be so, a condition was added of: if model X is positive, then the ensembled prediction is positive too. For example: if SVM is positive, or if the majority vote is equal or greater than 2 then the ensembled prediction is positive. This was tested with X being each of the models in the ensemble.

Ensemble Results

Table 5 shows the results of all the ensemble methods tested.

[16] Dr. Nan Ye. Data 7703 UQ Semester 2 2022. Lecture Notes. Week 7 – Ensemble Learning.
[17] Sklearn.ensemble.StackingClassifier documentation. url: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html

Table 5. Ensemble Results

Ensemble	TP	FP	TN	FN	TN to FN Ratio
Majority Vote (MV)	152: 74%	889: 27%	2372: 73%	54: 26%	1:6
MV or RF positive	161: 78%	928: 28%	2333: 72%	45: 22%	1:6
MV or SVM positive	157: 76%	1476: 45%	1785: 55%	49: 24%	1:9
MV or XGB positive	155: 75%	972: 30%	2289: 70%	51: 25%	1:6

The best ensemble model was the weighed approach where the ensembled prediction is fraudulent if the RF or if the majority vote predict it as fraudulent. This model had the highest number of true positives predicted, while also having an acceptable value of false positives and maintaining a result of the 1:6 threshold ratio.

It should be noted that in the seminar component of the project it was stated that the majority vote method produced the best ensemble result. The weighed method described was discovered and tested after the seminar was delivered. As a result, this report diverges form the seminar with regards to the best performing ensemble method.

Model Run Time

The running of just the ensemble component, once the three basis models were fitted took just 0.01 seconds. However, the overall time to run the ensemble model should consider the time it takes to run its components as well. Therefore, the total ensemble time would become the sum of running the RF, SVM, XBG and combining them into the ensemble. This results in a total run time of 78.7 seconds. Most of this time is the 77 seconds required to run the SVM model

Model Results and Comparison

Combining all the models and results in Table 6, it has shown that apart from the SVM model, the tree-based models all performed well.

Table 6. Model Results Comparison on Test Set.

Model	TP	FP	FN	ROC	Total Records Flagged	Precision	Run Time (seconds)
RF	160	919	46	0.819	1079: 31%	15%	1.24
SVM	58	849	148	0.497	907: 26%	6%	77.2
XBG	147	957	69	0.796	1104: 32%	13%	0.25
Ensemble	161	928	45	-	1089: 31%	15%	78.7

The ensemble performance slightly improved on the performance of the RF model in terms of True positives classification and performs similarly to RF by flagging 31% of the dataset as potentially fraudulent with 15% of the flagged cases being true positives.

Limitations

The analysis had several limitations that were identified. The most pertinent was the skewedness towards non fraudulent cases. Although this is representative of real-life data, the creation of synthetic data points does have its limitations and could be an issue when dealing with future unseen data. The more data points the model can be trained on will help to reduce this as a limitation.

There was also only a single data set available to analyse. This limitation was mitigated through the splitting of the data into three sets, namely a training, test and validation set. Future improvement would require additional tuning of the models on new unseen data with the same categorical features.

With only the single data set and the highly categorical nature of that data, any future data sets would have to have the same features for the tuned models to work. Additional features would require re-testing of the models and finding their optimum hyperparameters. PCA may become more important with more features or for determining the feature importance between common features in the data sets.

Conclusion and Discussion

The challenge in this analysis was the determination of what constituted success allowing for quantitative hyperparameter tuning to determine the successful model. Traditional metrics such as accuracy and F1 score did not provide the robustness required for implementation of the model in a real-life situation. It was theorised that insurance companies would be happy to accept some false positives within reason to maximise true positives but an increase in false positives would also result in a higher workload of the human resources required to investigate these cases. As such a trade-off between true positives and false positives was determined to allow for the best model to be chosen. To aid this and allow for future changes in false positive acceptance, a threshold ratio was created. Models that fell below this threshold could then be deemed adequate and the model with the highest true positive result would be classed the successful model.

The novelty of the approach was not only in creating the model but also in the tuning of the SMOTE cluster value of *k*. The results of each model showed a single value of *k* was not sufficient in tuning other model hyperparameters and rather iterating through *k* clusters with other parameters yielded differing results. This is an important observation when dealing with unbalanced data in this way as SMOTE creates the synthetic data points but creating them and testing them in a model should be considered two different objectives to be tuned.

This analysis compared three machine learning models and an ensemble against the data set. It demonstrated the effect of SMOTE *k* cluster tuning as well as model hyperparameters on the tested models and how differing iterations of these parameters could achieve a successful outcome based off the previously discussed success criteria. It identified the use of tree-based decision models as performing the best for this highly categorical data set with RF and XGB providing good results. The use of SVM on the data set provided poor results when compared to RF and XGB as well as the computational time to build the model was much more excessive. Putting the models into an ensemble did yield a good result; however, this result was heavily determined by the RF model contribution and as such the additional computational time to run the ensemble did not produce significantly better results than just RF by itself.

Although the ensemble was the aim of this analysis and was expected to provide better results than the single models, its application in model selection should not be discounted. The analysis, although computationally intensive with the number of models tested, does ideally need to be conducted to identify which approach is best suited to any datasets and in doing so rule out an approach.

The RF model was deemed the successful model for further implementation as it provided the highest true positives with the lowest number of false positives below the threshold. The implementation time of the RF model was higher than XGB so if computational speed was a requirement, XGB would be the preferred model.

The significance of this analysis compared to other studies done in this area is its application to real-life implementation and requirements. Fraud identification will never be perfect and will always require a trade-off between human resource investigation and the time it takes to identify the cases. Our approach has managed to show an ability to reduce the identification time significantly from a manual process and also allow for management of human resource workloads through the threshold ratio being able to modify the models towards what is able to be investigated. This allows for cost benefit decisions to be made in industry if implementing this approach.

Next Steps

Our project's aim of improving results using an ensemble method did not provide the desired outcome of improving on the singular models and was mostly due to the influence of the SVM model. Using a third ensemble model based off another tree-based algorithm or KNN clustering could help improve the ensemble and would be a good next step in improving the model's ability at identifying fraudulent behaviour.

Although RF showed good model performance and provided a significantly better result than random selection there is still improvement that can be done on this model as well. With the addition of more features and more data to train the model, this is expected to improve the model's identification accuracy and reduce the false positive identification. The dataset is also expected to contain more fraudulent cases than have been flagged as fraud and these cases could be contained within the false positives predicted by the model. Through the investigation of the false positive cases predicted by the model it may identify fraudulent behaviour and as such the case status in the data set can be updated to fraudulent. A third target variable of "investigated non fraudulent" could also be assigned for those cases that have been manually identified as non-fraudulent. If the models in this analysis are re-run with this updated data and weights placed on penalising any positively identified fraudulent cases (third target variable), this would help to tune the models to reduce the false positives and improve classification of fraudulent cases.

Appendix A

All coding for this project was done in python utilising existing machine learning, data science, mathematics and visualisations tools and libraries. The codes and files used for this project are described in Table 7 below and is submitted as a .zip file with this report.

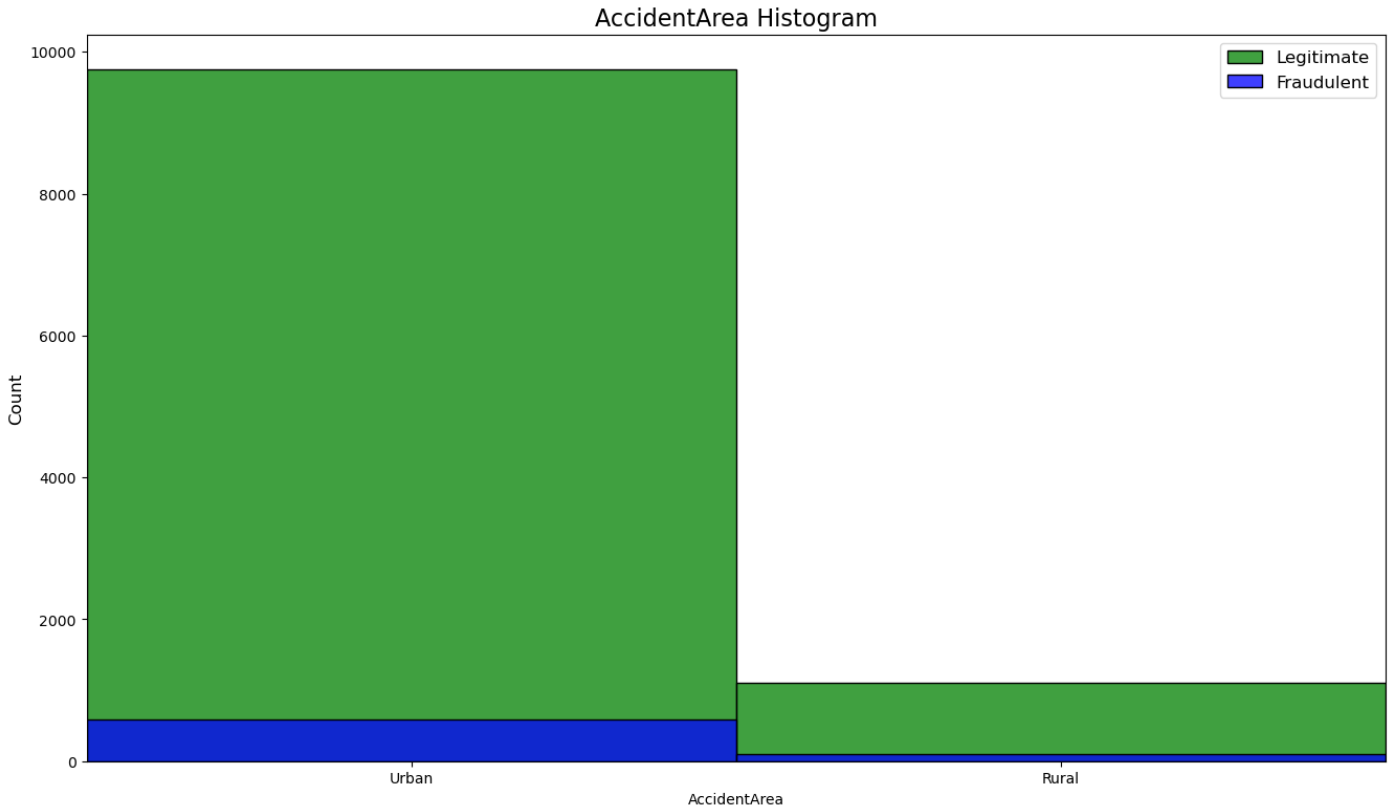
Table 7. Materials Used for Project

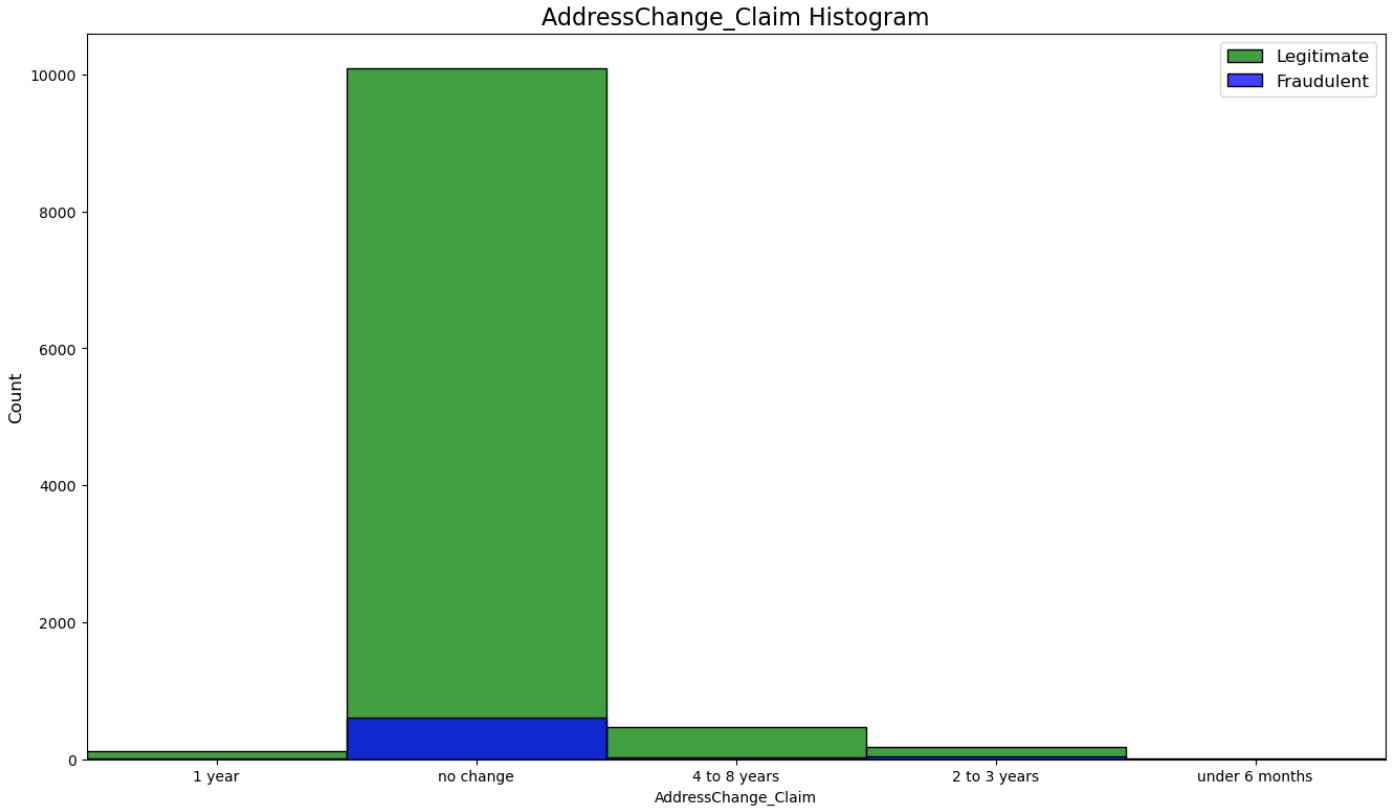
File Types	Name	Description
Code	01_data7703_dataset_eda.py	Initial data analysis, cleaning, histogram visualisations, and creation feature selected subsets for further analysis
	01a_data7703_unbalanced_data.py	SMOTE and STMOTETomek test
	02_data7703_random_forest.py	Tuning and training random forest models, exports of results
	03_data7703_svm.py	Tuning and training SVM models, exports of results
	04_data7703_random_XGB.py	Tuning and training XGB models, exports of results
	05_data7703_ensemble.py	Tuning an ensemble model based on chosen RF, SVM and XGB models , export of results
Datasets	Dataset.csv	Raw data downloaded directly from source
	Dataset_clean.csv	Cleaned dataset (code 01)
	data_factorised.csv	Cleaned factorised dataset (code 01)
	data_clean_reducted.csv	Cleaned dataset with agreed features removed (code 01)
	data_factorised_reducted.csv	Cleaned factorised dataset with agreed features removed (code 01)
	data_factorised_minimal.csv	Cleaned dataset with the redacted and the grey features removed (code 01)
EDA Support	raw_dataset_unique_values.csv	List of all columns in the dataset and their unique values (code 01)
	description_string_columns.csv	Description and statistics of string columns (code 01)
	description_numeric_columns.csv	Description and statistics of numeric columns (code 01)
RF Results	rf_full_data_test.csv	Results of tuning on the test dataset, from training on the full factorised dataset (code 02)
	rf_full_data_validation.csv	Results of tuning on the validation dataset, from training on the full factorised dataset (code 02)
	rf_test.csv	Results of tuning on the test dataset, from training on the redacted factorised dataset (code 02)
	rf_validation.csv	Results of tuning on the validation dataset, from training on the redacted factorised dataset (code 02)
	rf_minimal_data_test.csv	Results of tuning on the test dataset, from training on the minimal factorised dataset (code 02)
	rf_minimal_data_validation.csv	Results of tuning on the validation dataset, from training on the minimal factorised dataset (code 02)
SVM Results	svm_full_data_test.csv	Results of tuning on the test dataset, from training on the full factorised dataset (code 03)
	svm_full_data_validation.csv	Results of tuning on the validation dataset, from training on the full factorised dataset (code 03)
	svm_test.csv	Results of tuning on the test dataset, from training on the redacted factorised dataset (code 03)
	svm_validation.csv	Results of tuning on the validation dataset, from training on the redacted factorised dataset (code 03)
	svm_minimal_data_test.csv	Results of tuning on the test dataset, from training on the minimal factorised dataset (code 03)
	svm_minimal_data_validation.csv	Results of tuning on the validation dataset, from training on

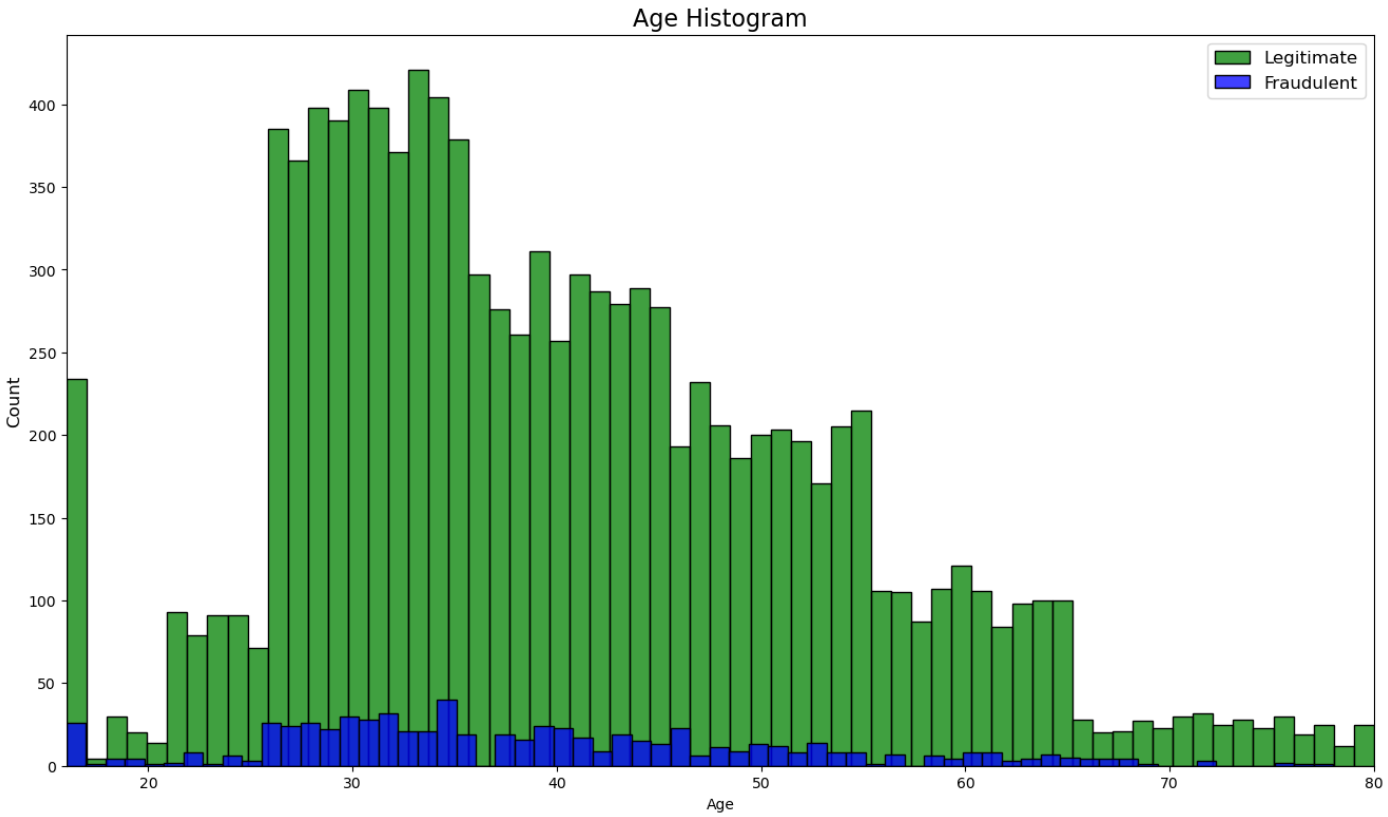
		the minimal factorised dataset (code 03)
XGB Results	xgb_full_data_test.csv	Results of tuning on the test dataset, from training on the full factorised dataset (code 04)
	xgb_full_data_validation.csv	Results of tuning on the validation dataset, from training on the full factorised dataset (code 04)
	xgb_test.csv	Results of tuning on the test dataset, from training on the redacted factorised dataset (code 04)
	xgb_validation.csv	Results of tuning on the validation dataset, from training on the redacted factorised dataset (code 04)
	xgb_minimal_data_test.csv	Results of tuning on the test dataset, from training on the minimal factorised dataset (code 04)
	xgb_minimal_data_validation.csv	Results of tuning on the validation dataset, from training on the minimal factorised dataset (code 04)

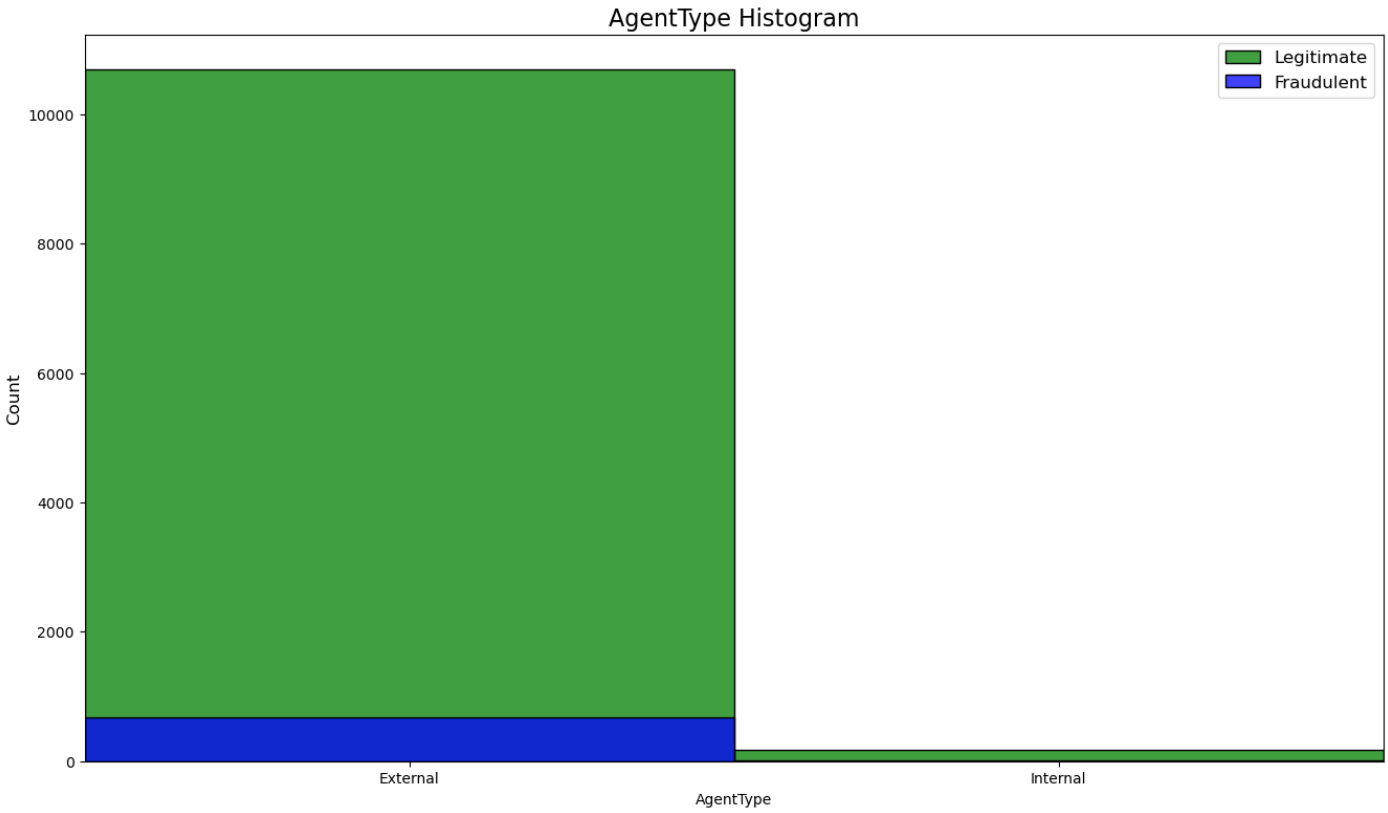
Appendix B

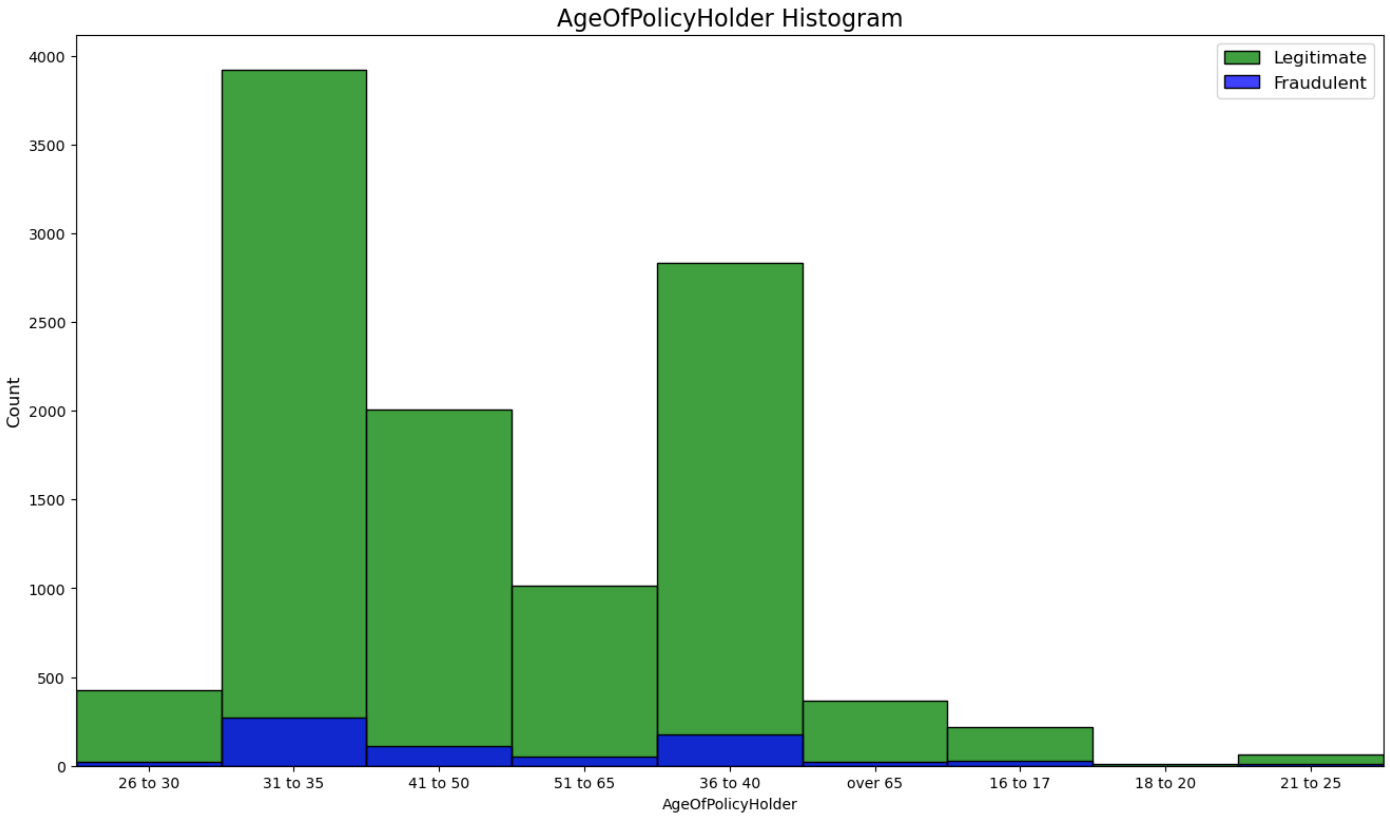
Feature histogram. Please note that string features were factorised for the creation of the histogram and each number correlates to a unique value.

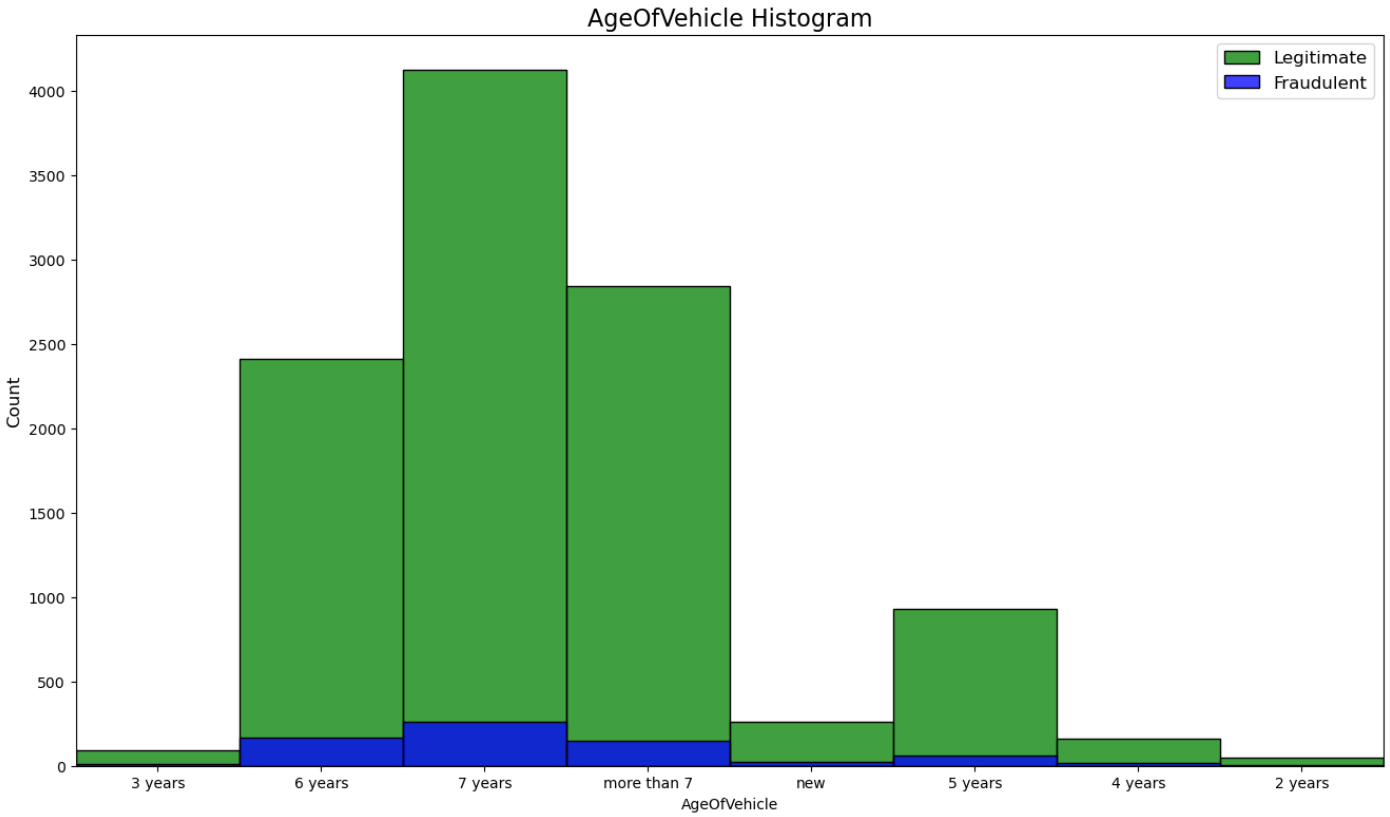

AccidentArea Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
Urban
Rural
AccidentArea

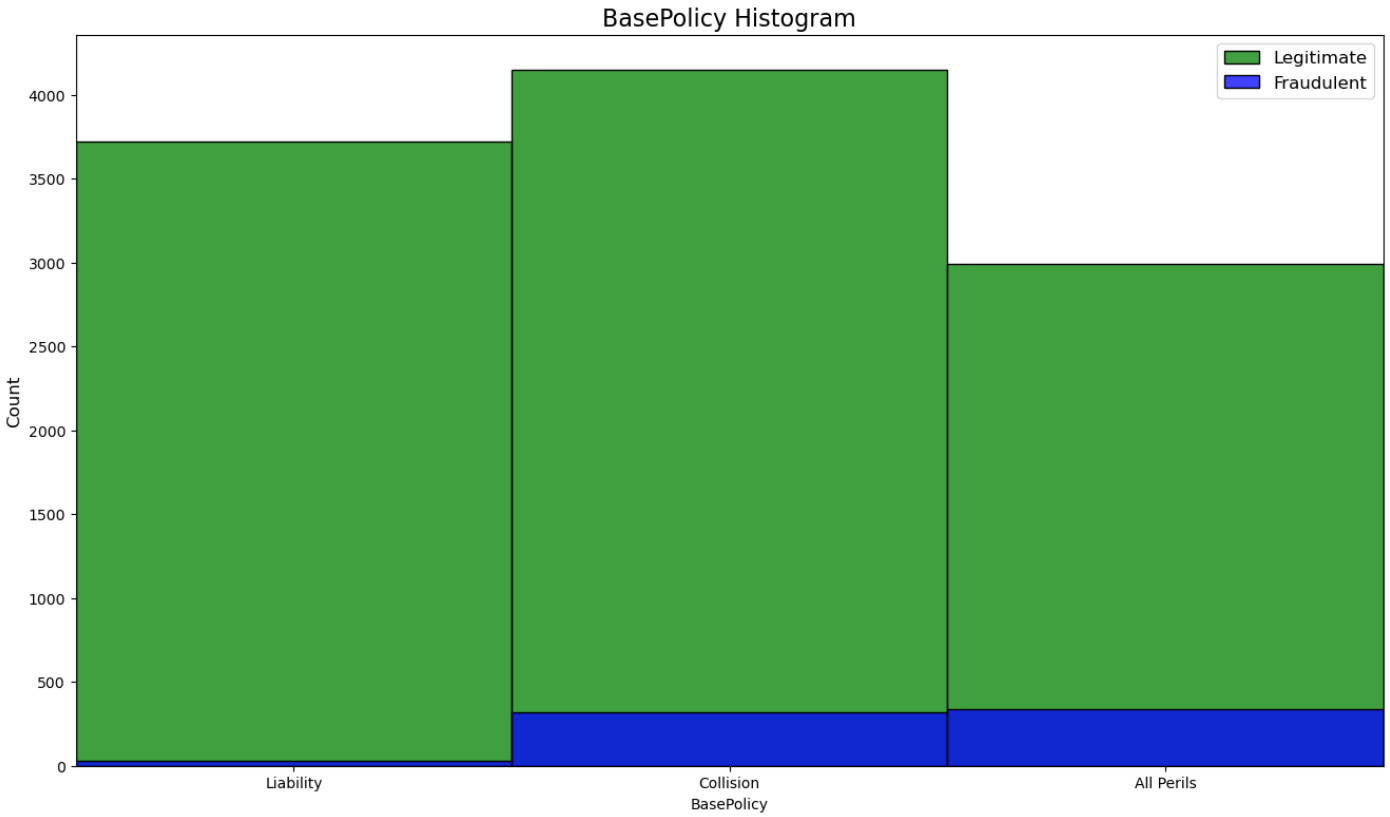

AddressChange_Claim Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
1 year
no change
4 to 8 years
2 to 3 years
under 6 months
AddressChange_Claim

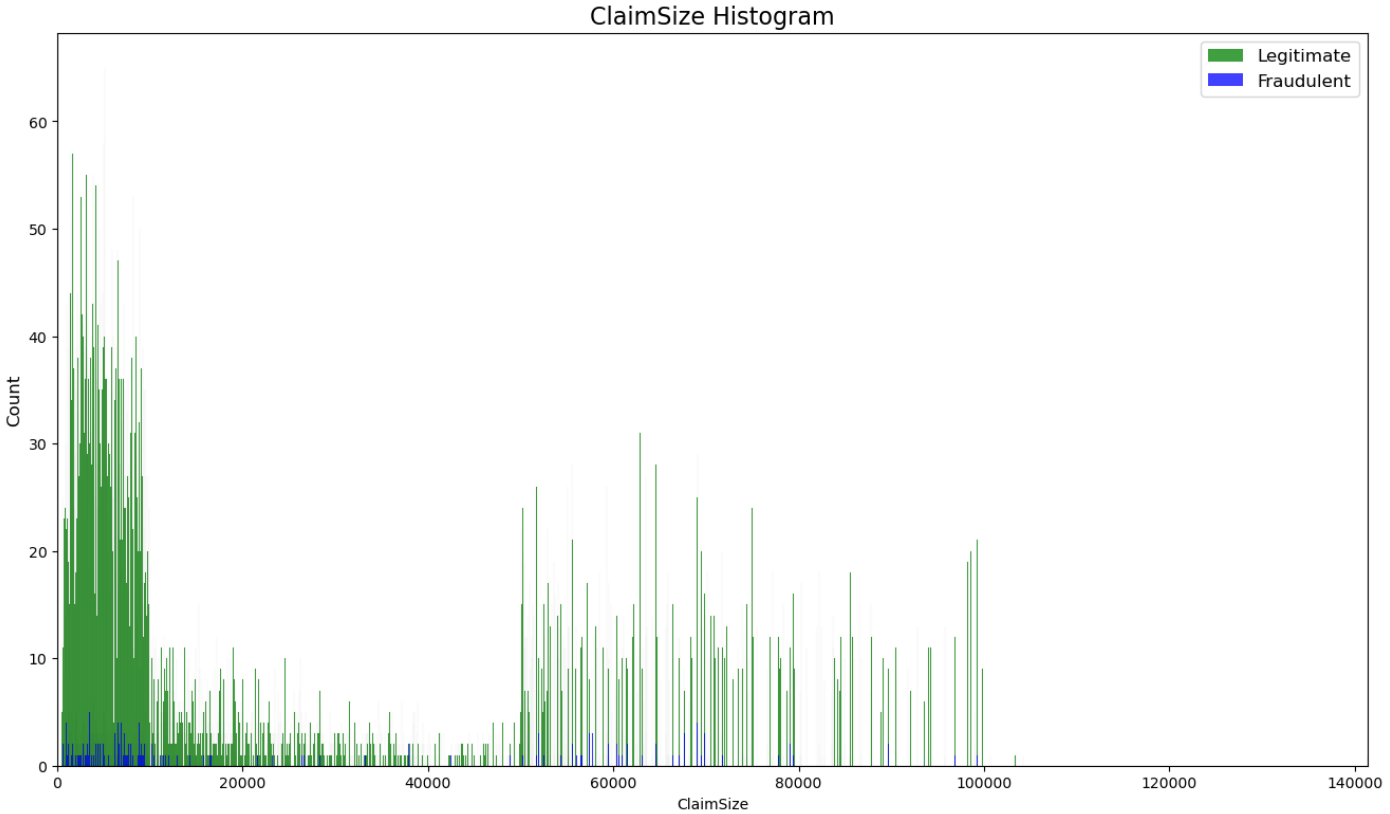

Age Histogram
Legitimate
Fraudulent
Count
Age
0
50
100
150
200
250
300
350
400
20
30
40
50
60
70
80

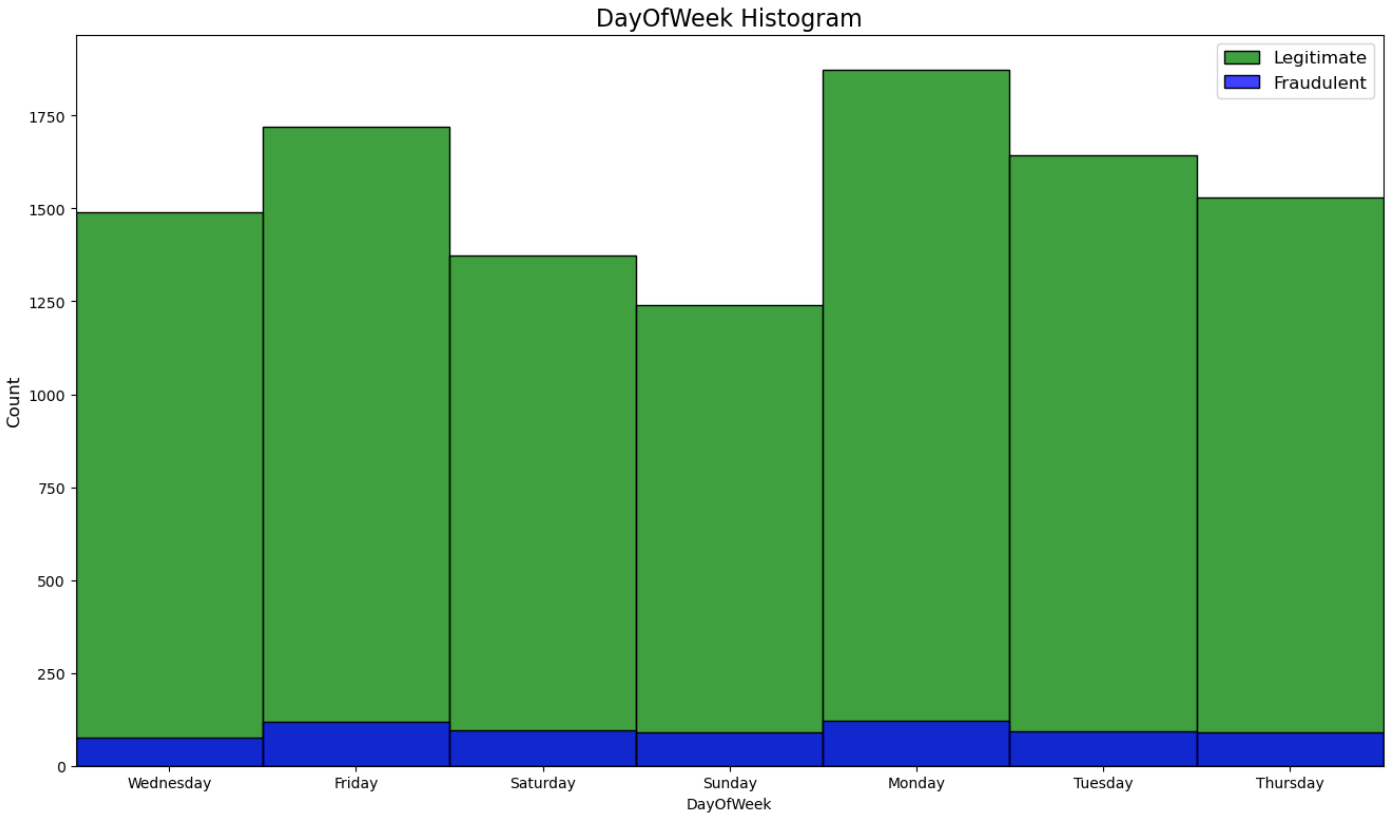

AgentType Histogram
Legitimate
Fraudulent
Count
AgentType
0
2000
4000
6000
8000
10000
External
Internal

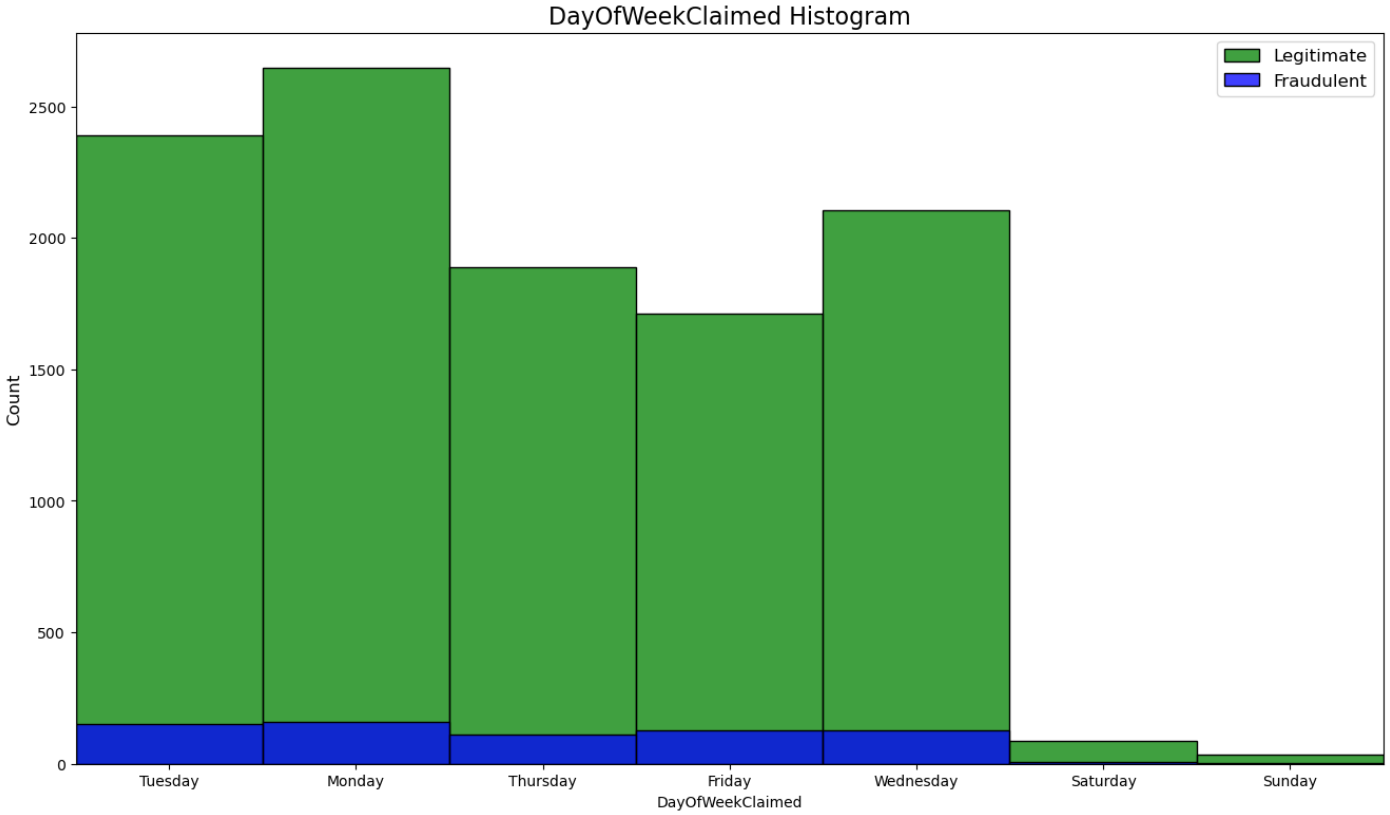

AgeOfPolicyHolder Histogram
Legitimate
Fraudulent
Count
0
500
1000
1500
2000
2500
3000
3500
4000
26 to 30
31 to 35
41 to 50
51 to 65
36 to 40
over 65
16 to 17
18 to 20
21 to 25
AgeOfPolicyHolder

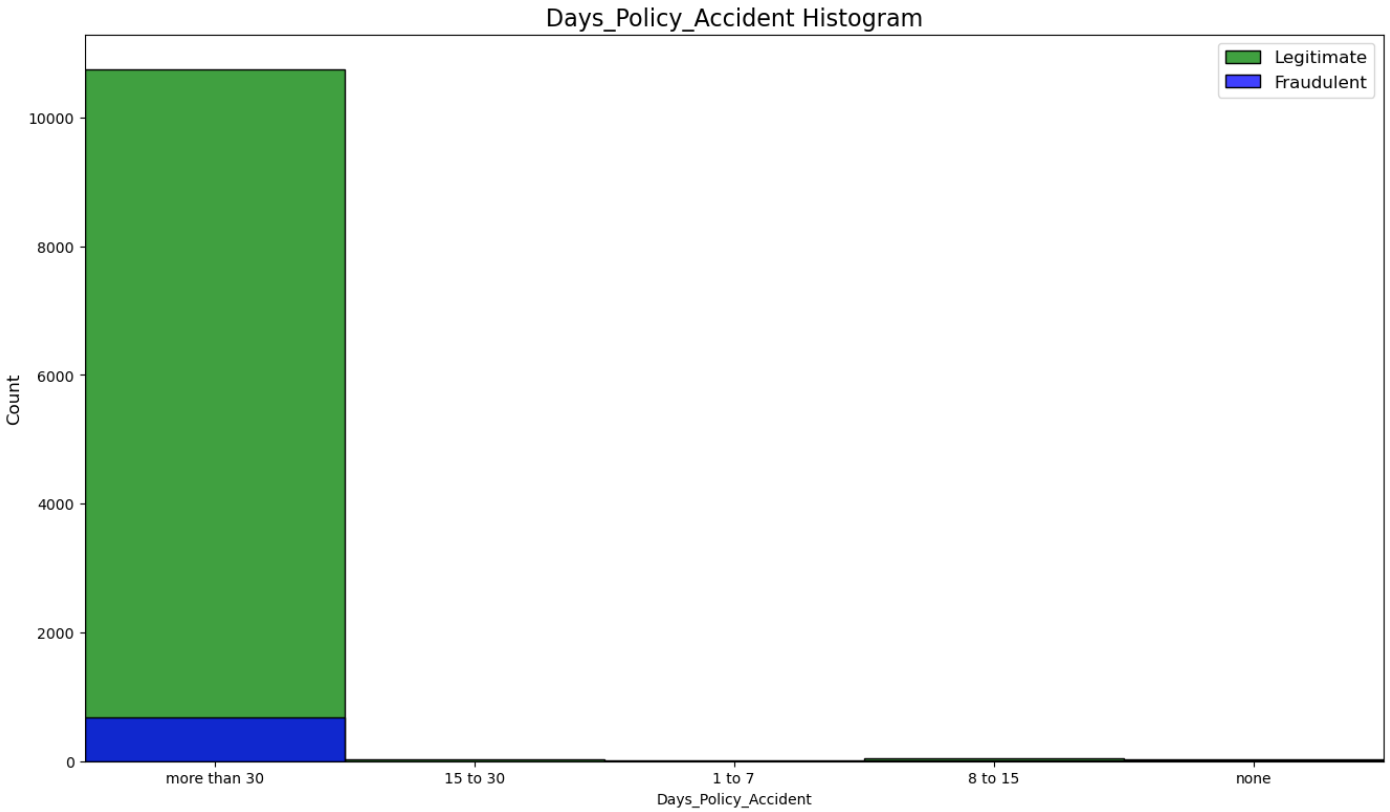

AgeOfVehicle Histogram
Legitimate
Fraudulent
Count
0
500
1000
1500
2000
2500
3000
3500
4000
3 years
6 years
7 years
more than 7
new
5 years
4 years
2 years
AgeOfVehicle

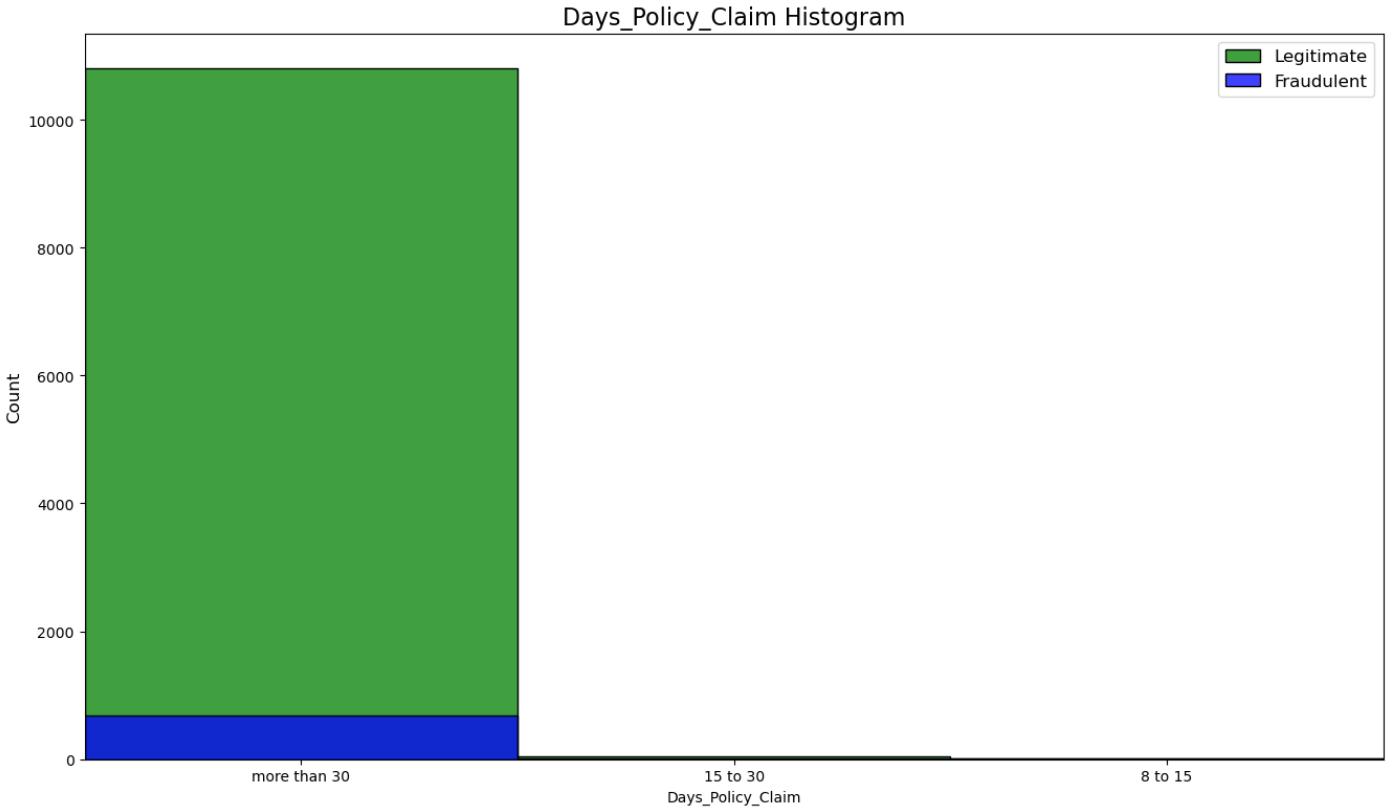

BasePolicy Histogram
Legitimate
Fraudulent
Count
0
500
1000
1500
2000
2500
3000
3500
4000
Liability
Collision
All Perils
BasePolicy

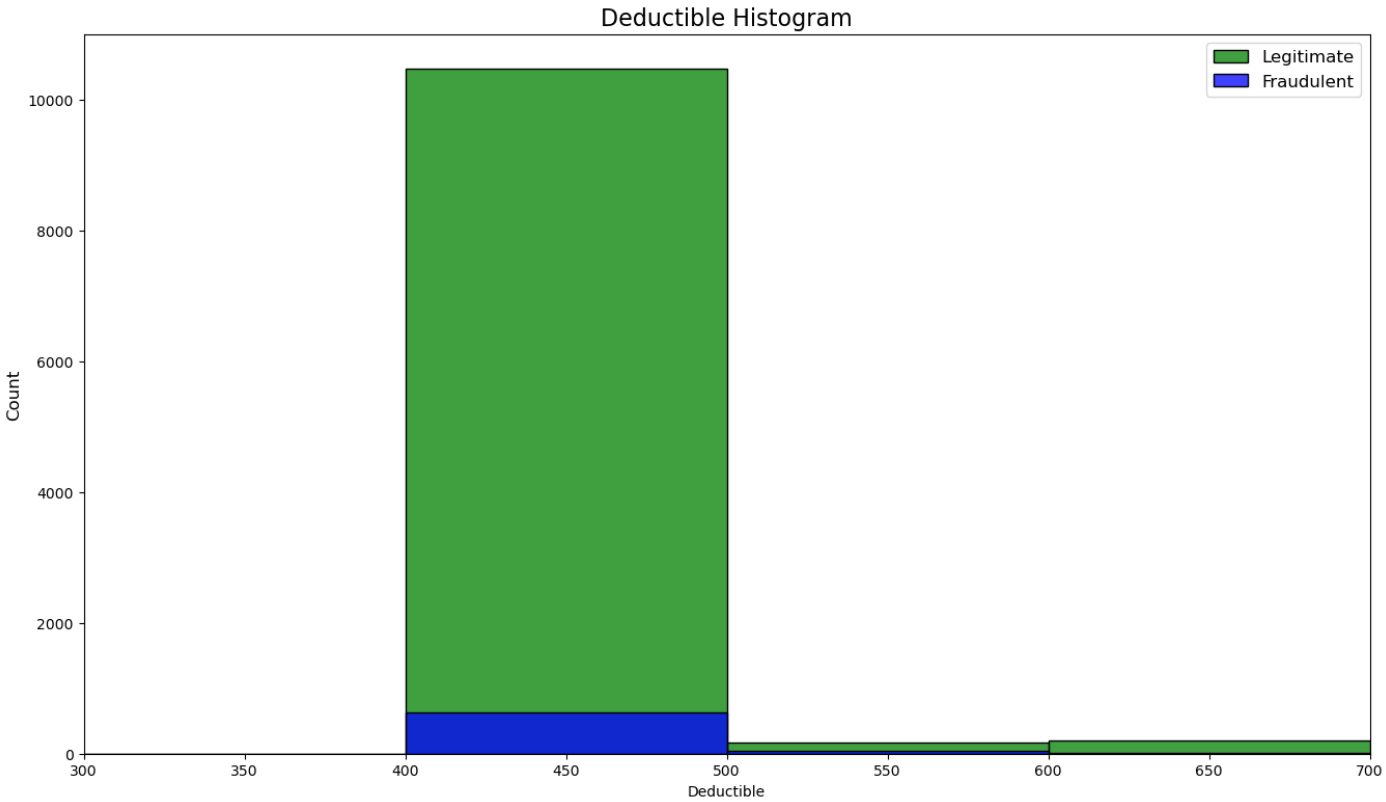

ClaimSize Histogram
Legitimate
Fraudulent
Count
0
10
20
30
40
50
60
0
20000
40000
60000
80000
100000
120000
140000
ClaimSize

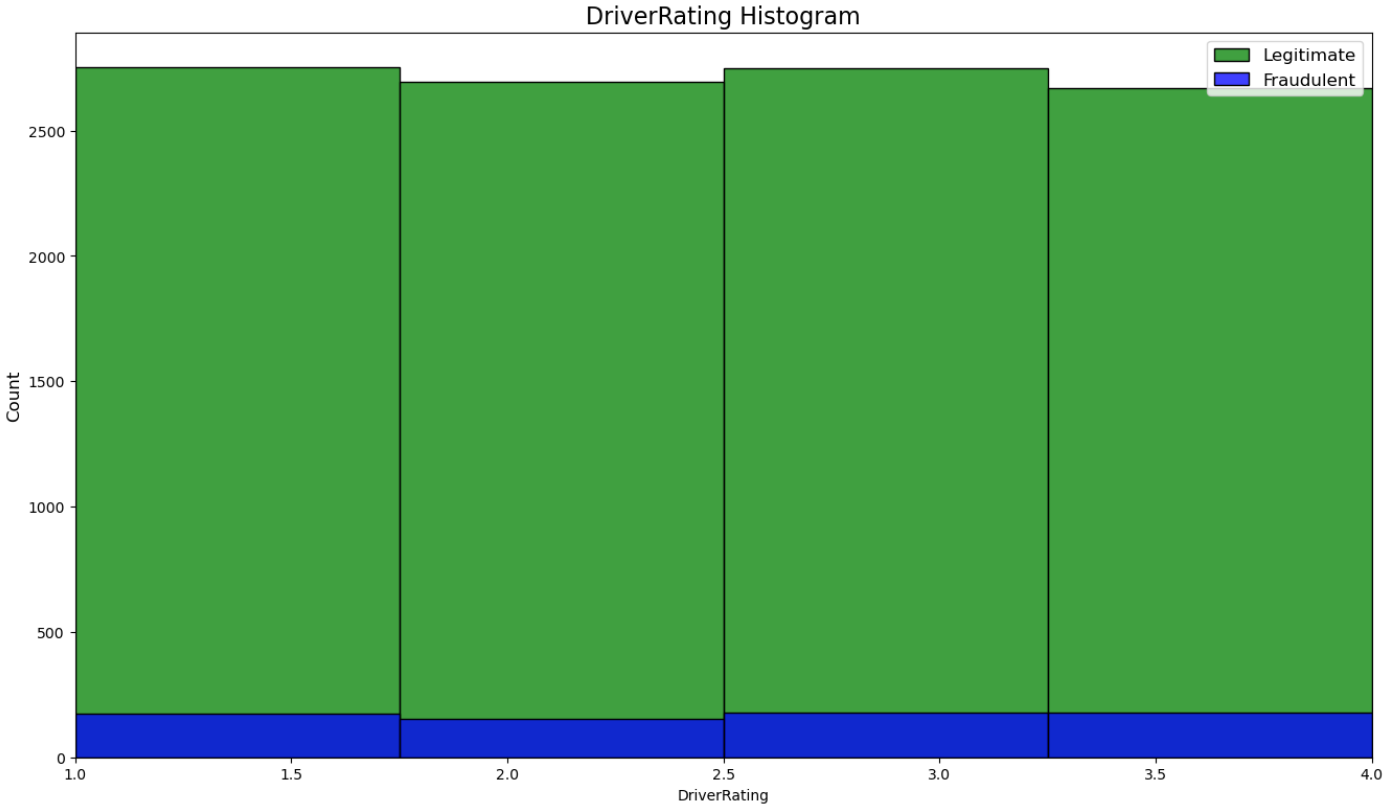

DayOfWeek Histogram
Legitimate
Fraudulent
Count
0
250
500
750
1000
1250
1500
1750
Wednesday
Friday
Saturday
Sunday
Monday
Tuesday
Thursday
DayOfWeek

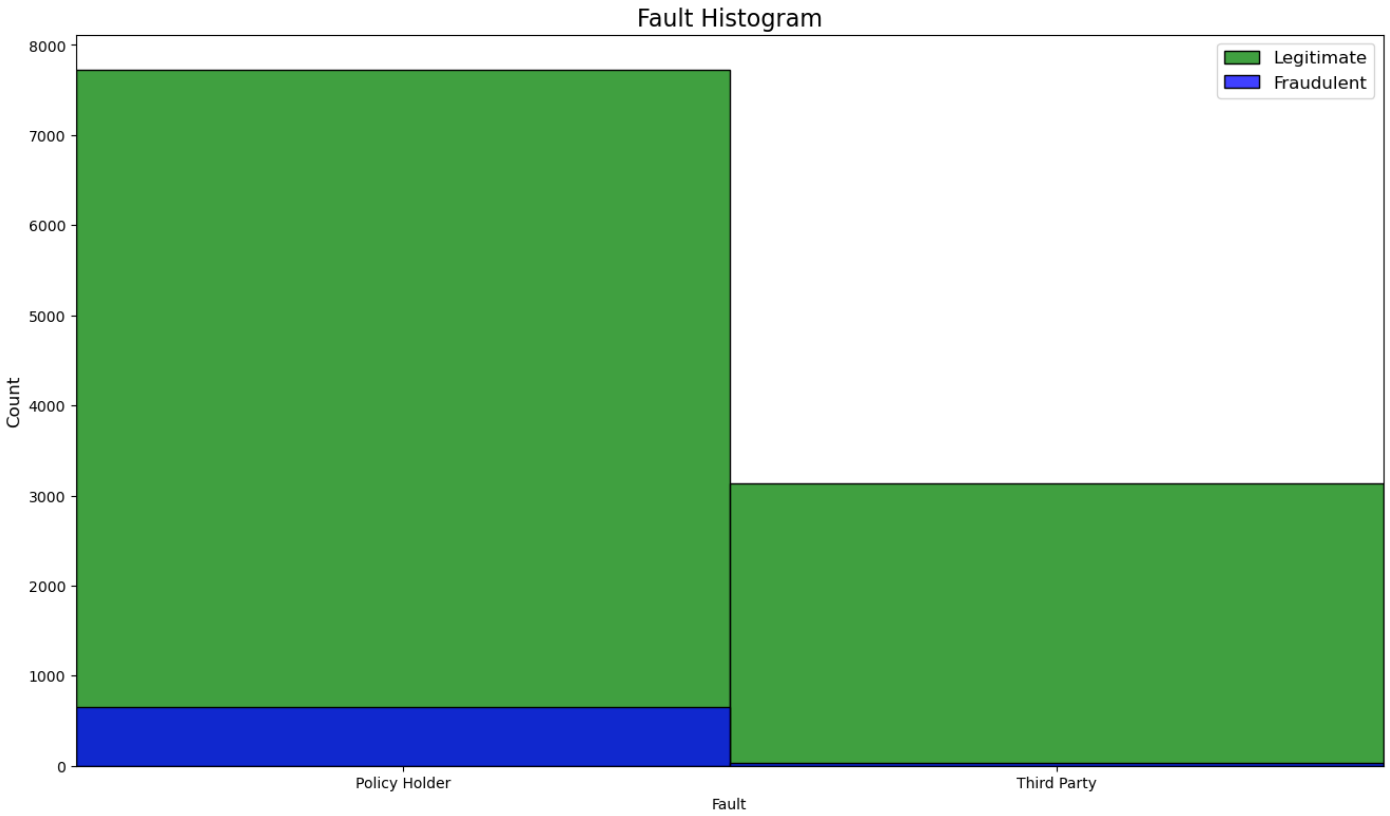

DayOfWeekClaimed Histogram
Legitimate
Fraudulent
Count
0
500
1000
1500
2000
2500
Tuesday
Monday
Thursday
Friday
Wednesday
Saturday
Sunday
DayOfWeekClaimed

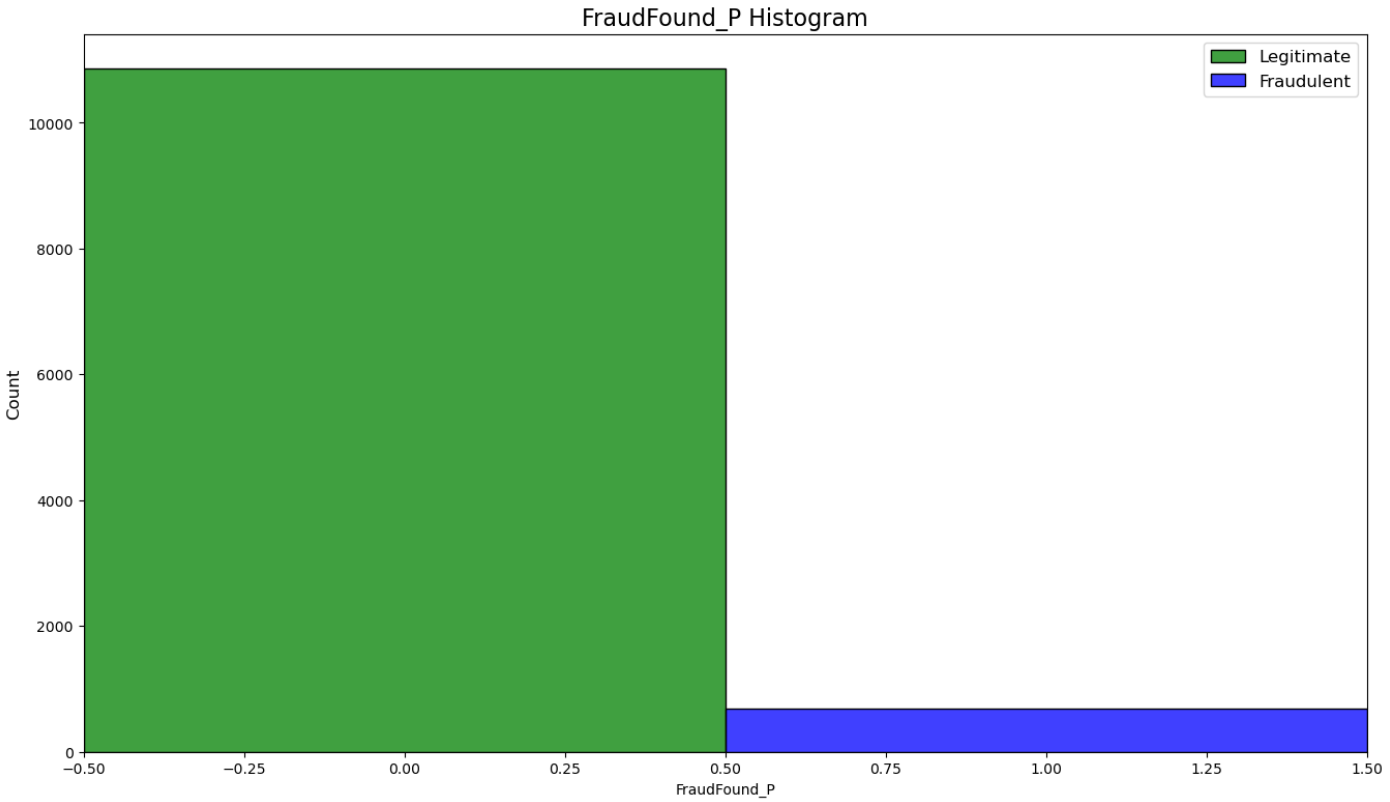

Days_Policy_Accident Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
more than 30
15 to 30
1 to 7
8 to 15
none
Days_Policy_Accident

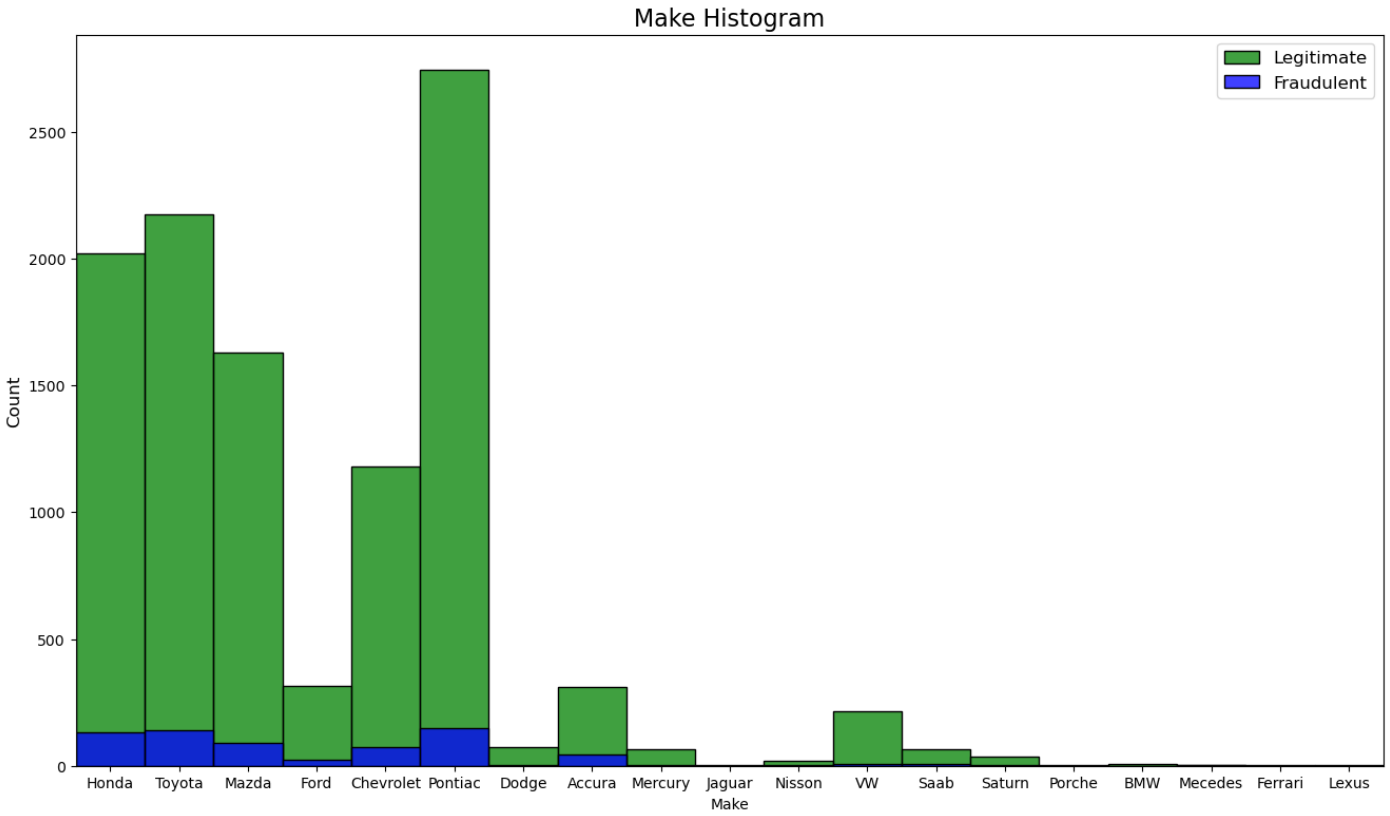

Days_Policy_Claim Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
more than 30
15 to 30
8 to 15
Days_Policy_Claim

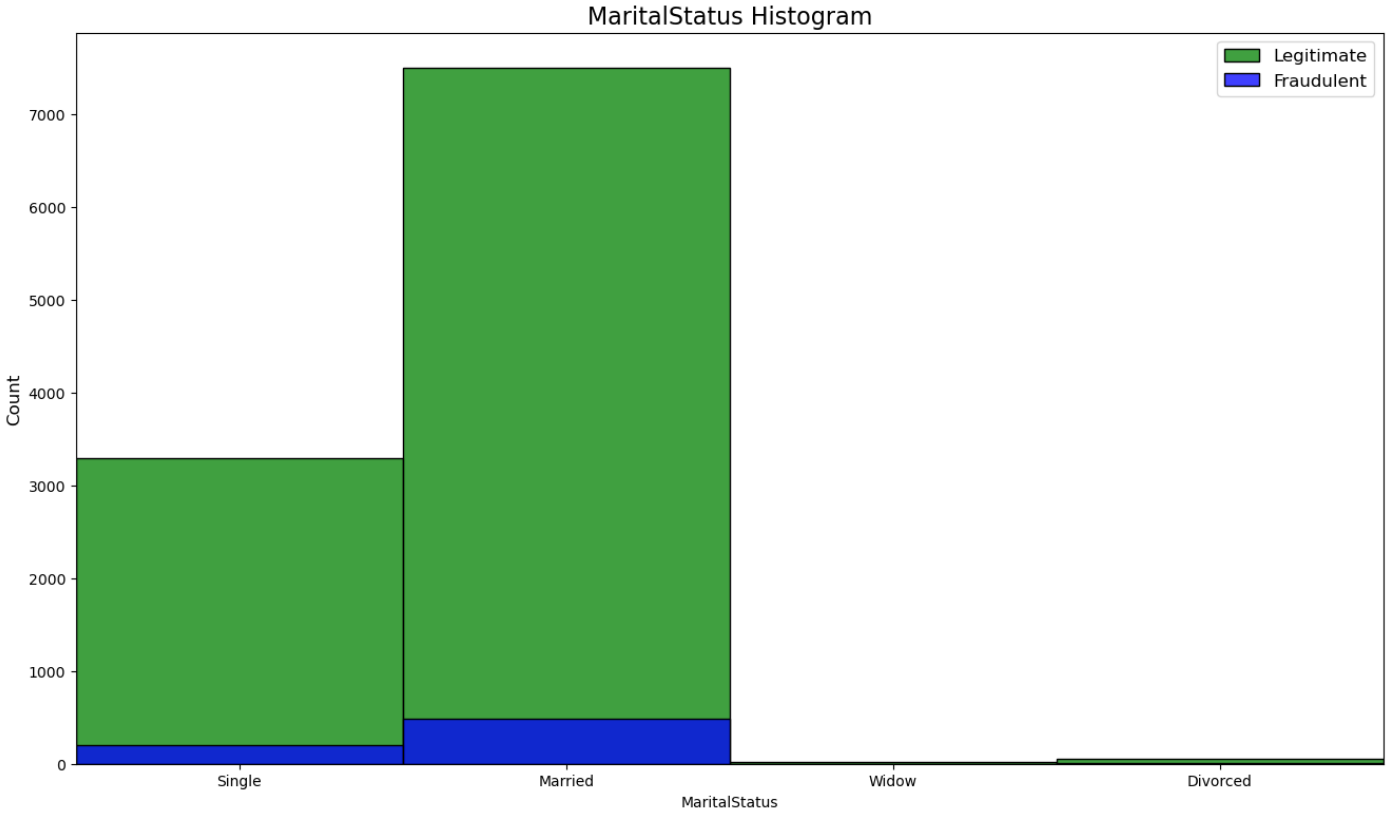

Deductible Histogram
Legitimate
Fraudulent
Count
Deductible
0
2000
4000
6000
8000
10000
300
350
400
450
500
550
600
650
700

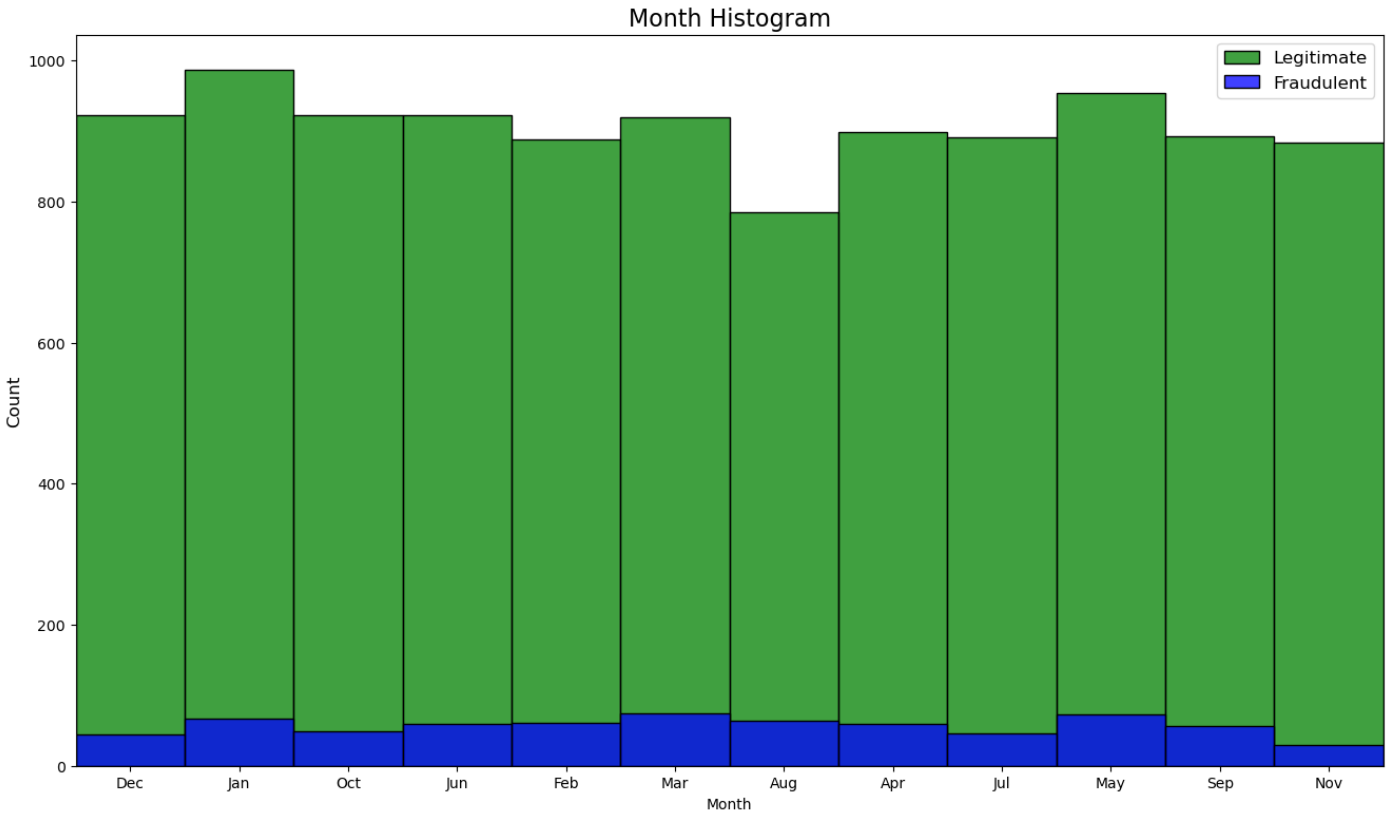

DriverRating Histogram
Legitimate
Fraudulent
Count
DriverRating
0
500
1000
1500
2000
2500
1.0
1.5
2.0
2.5
3.0
3.5
4.0

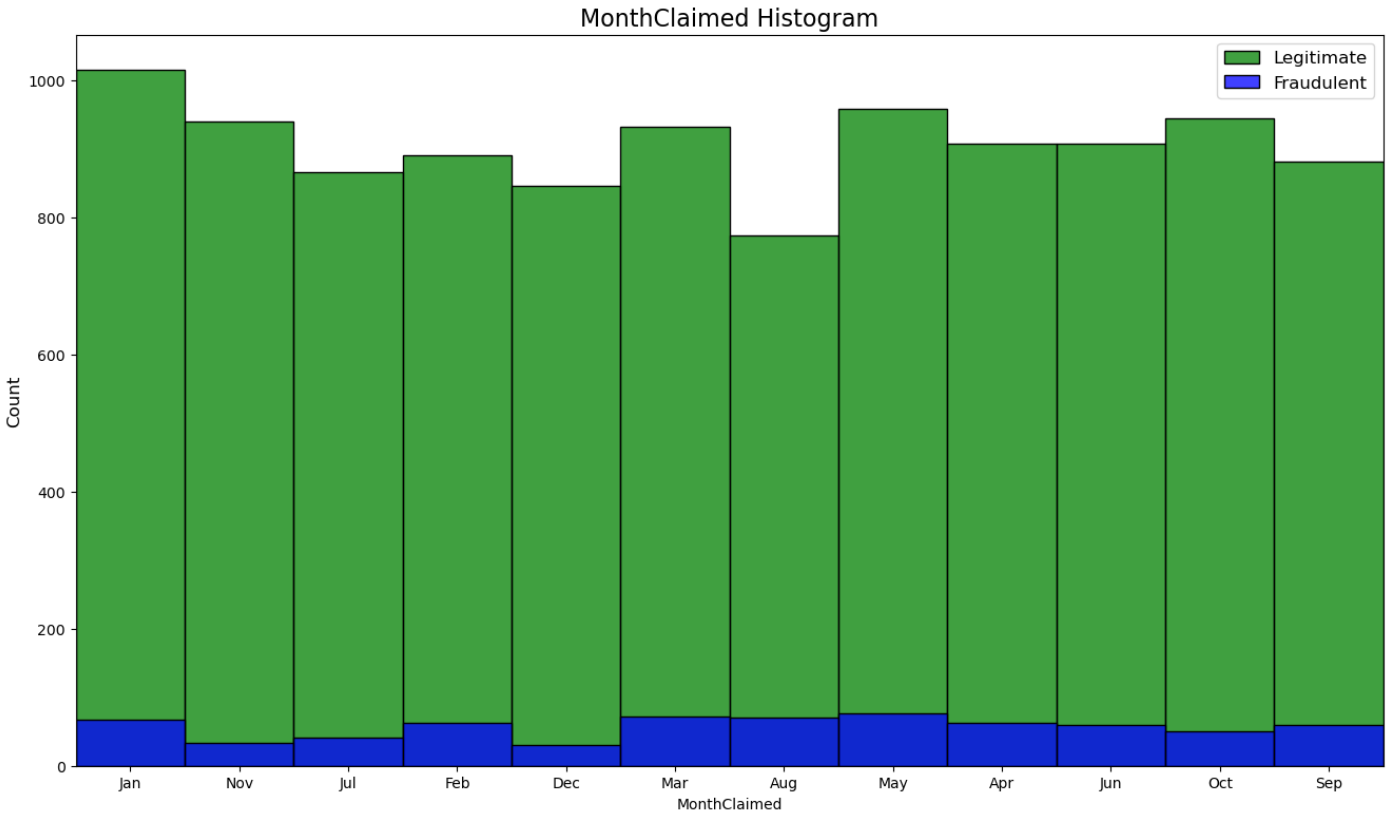

Fault Histogram
Legitimate
Fraudulent
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
Policy Holder
Third Party
Fault

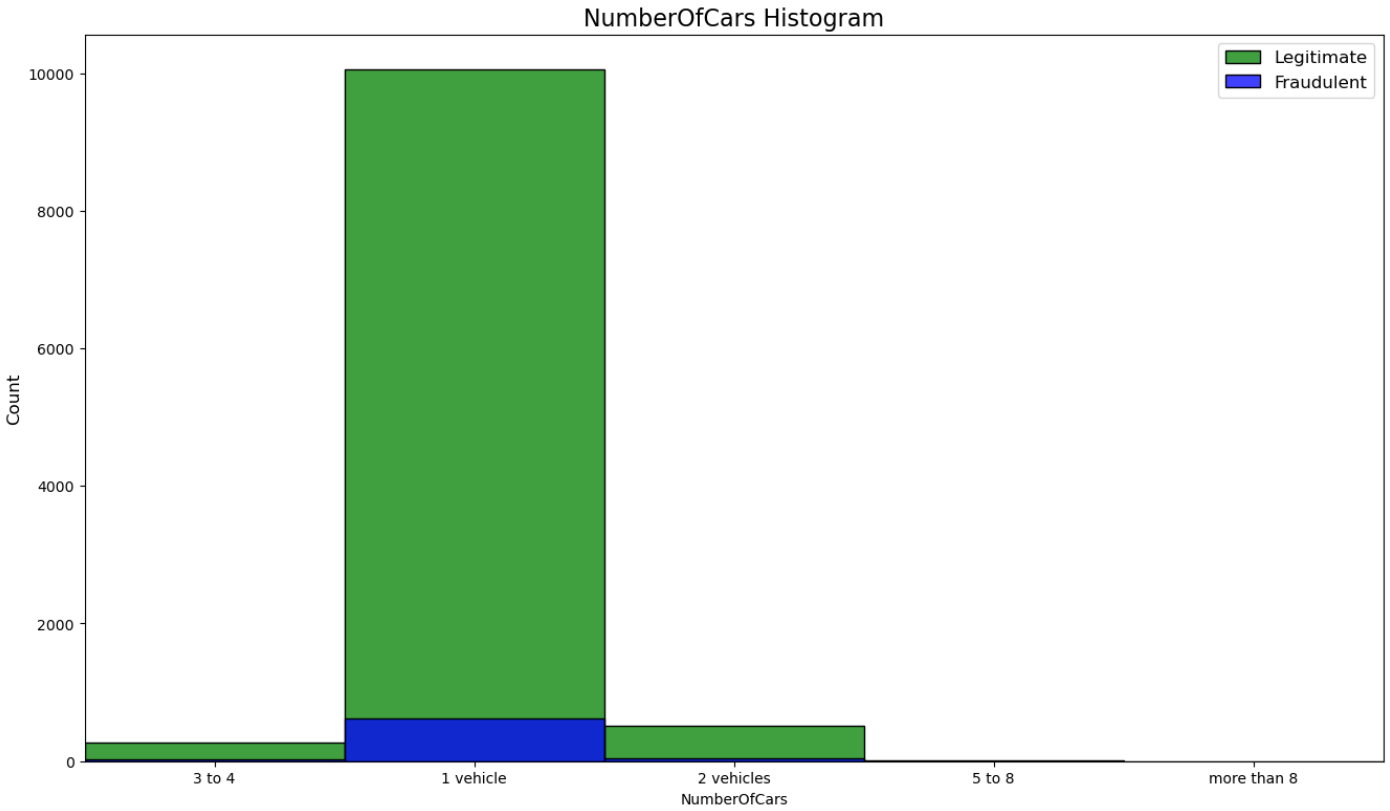

FraudFound_P Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
FraudFound_P

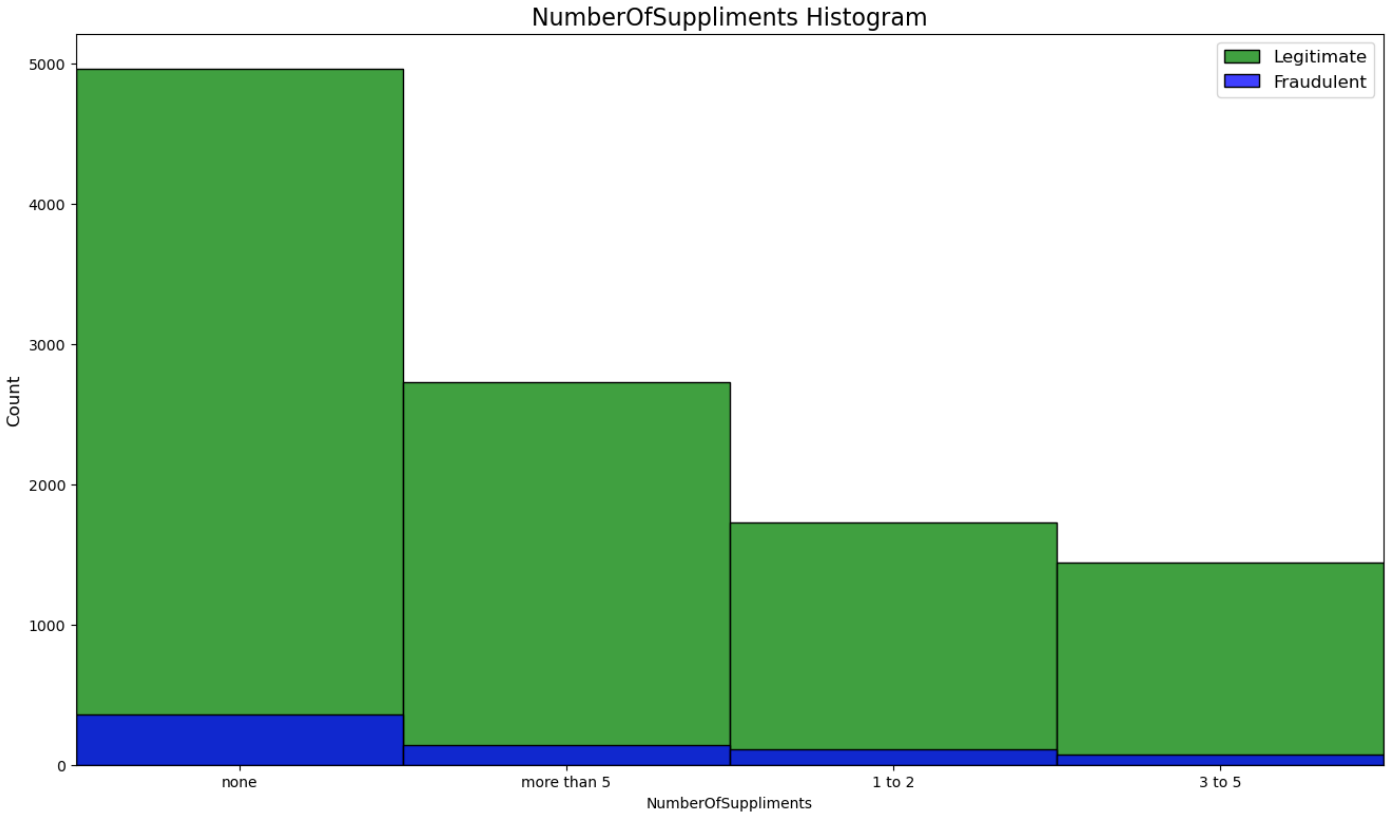

Make Histogram
Legitimate
Fraudulent
Count
0
500
1000
1500
2000
2500
Honda
Toyota
Mazda
Ford
Chevrolet
Pontiac
Dodge
Accura
Mercury
Jaguar
Nisson
VW
Saab
Saturn
Porche
BMW
Mecedes
Ferrari
Lexus
Make

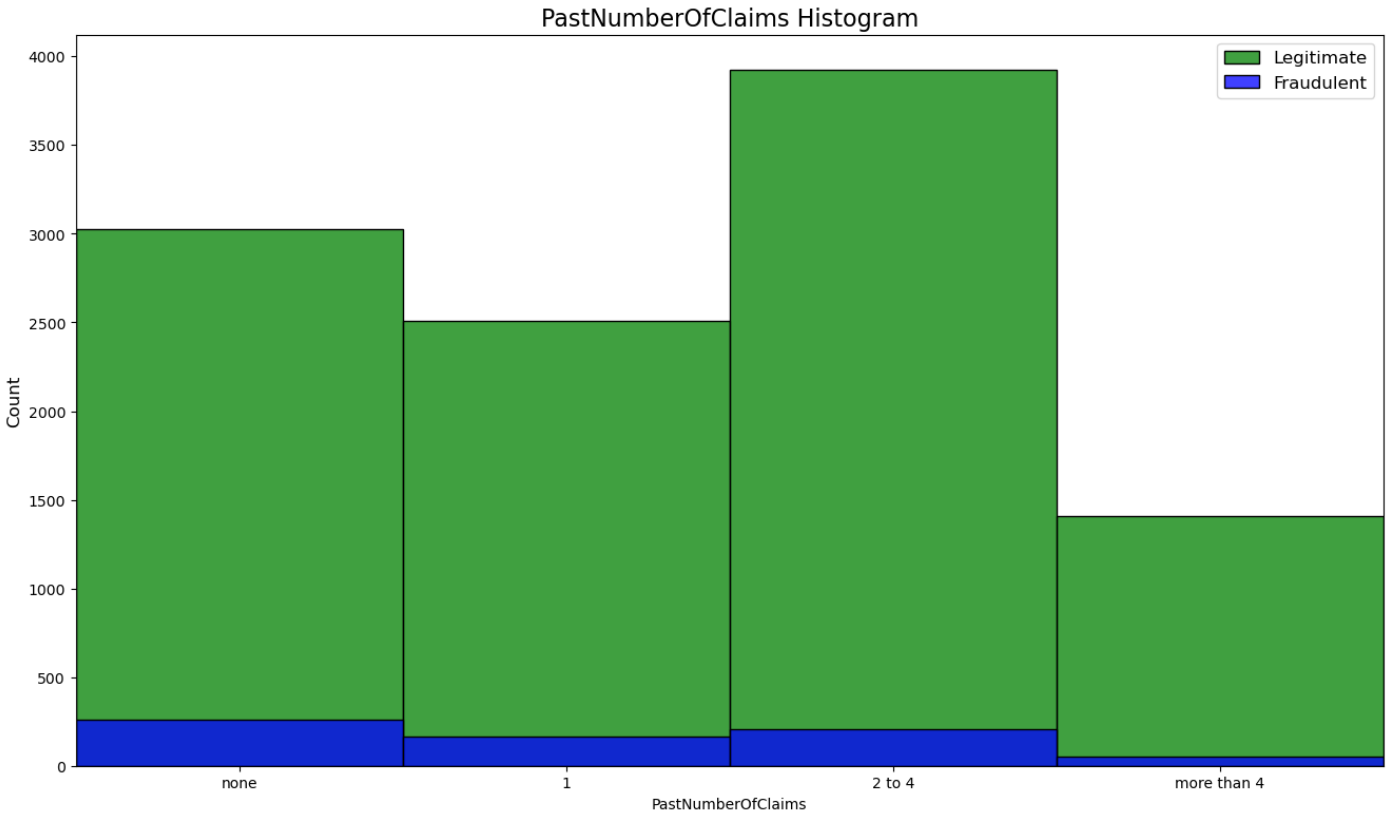

MaritalStatus Histogram
Legitimate
Fraudulent
Count
0
1000
2000
3000
4000
5000
6000
7000
Single
Married
Widow
Divorced
MaritalStatus

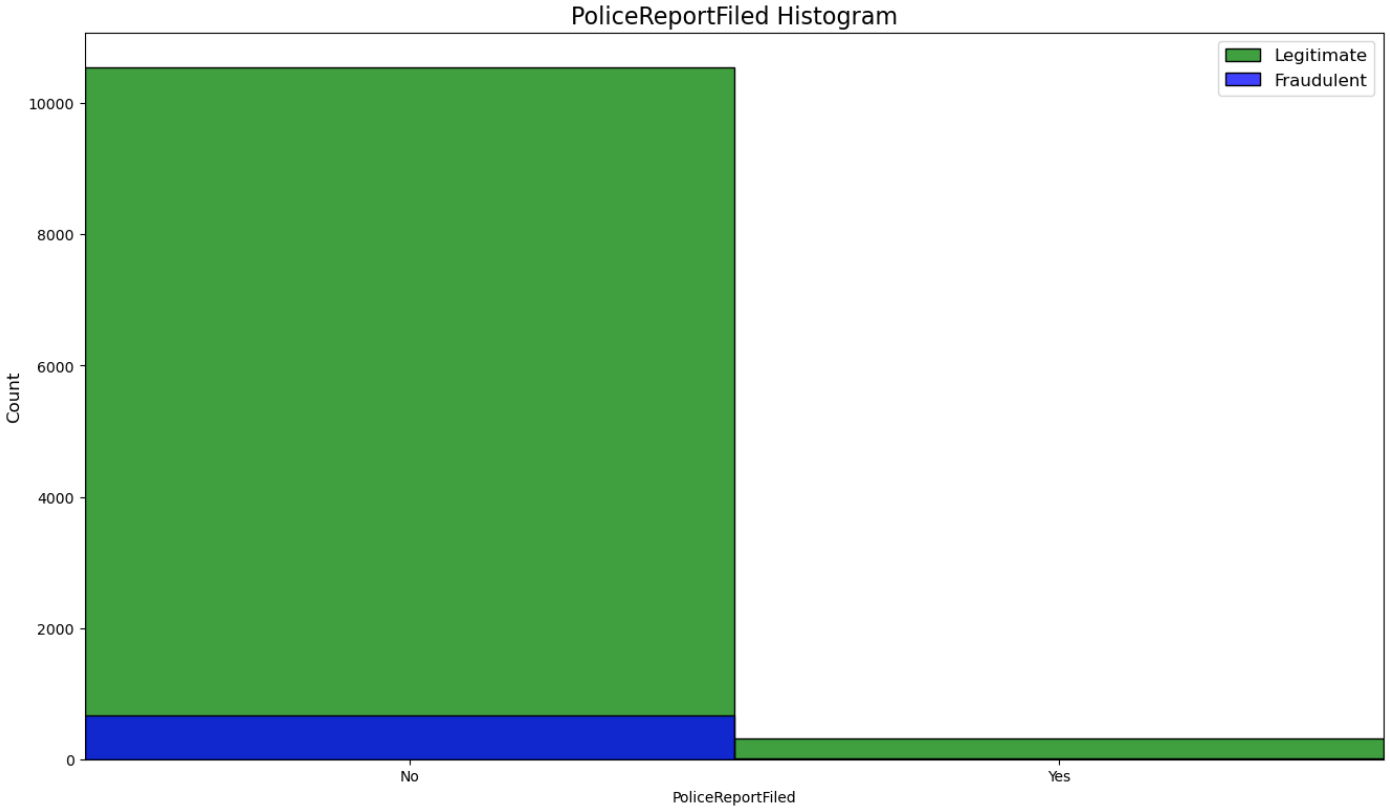

Month Histogram
Legitimate
Fraudulent
Count
1000
800
600
400
200
0
Dec
Jan
Oct
Jun
Feb
Mar
Aug
Apr
Jul
May
Sep
Nov
Month

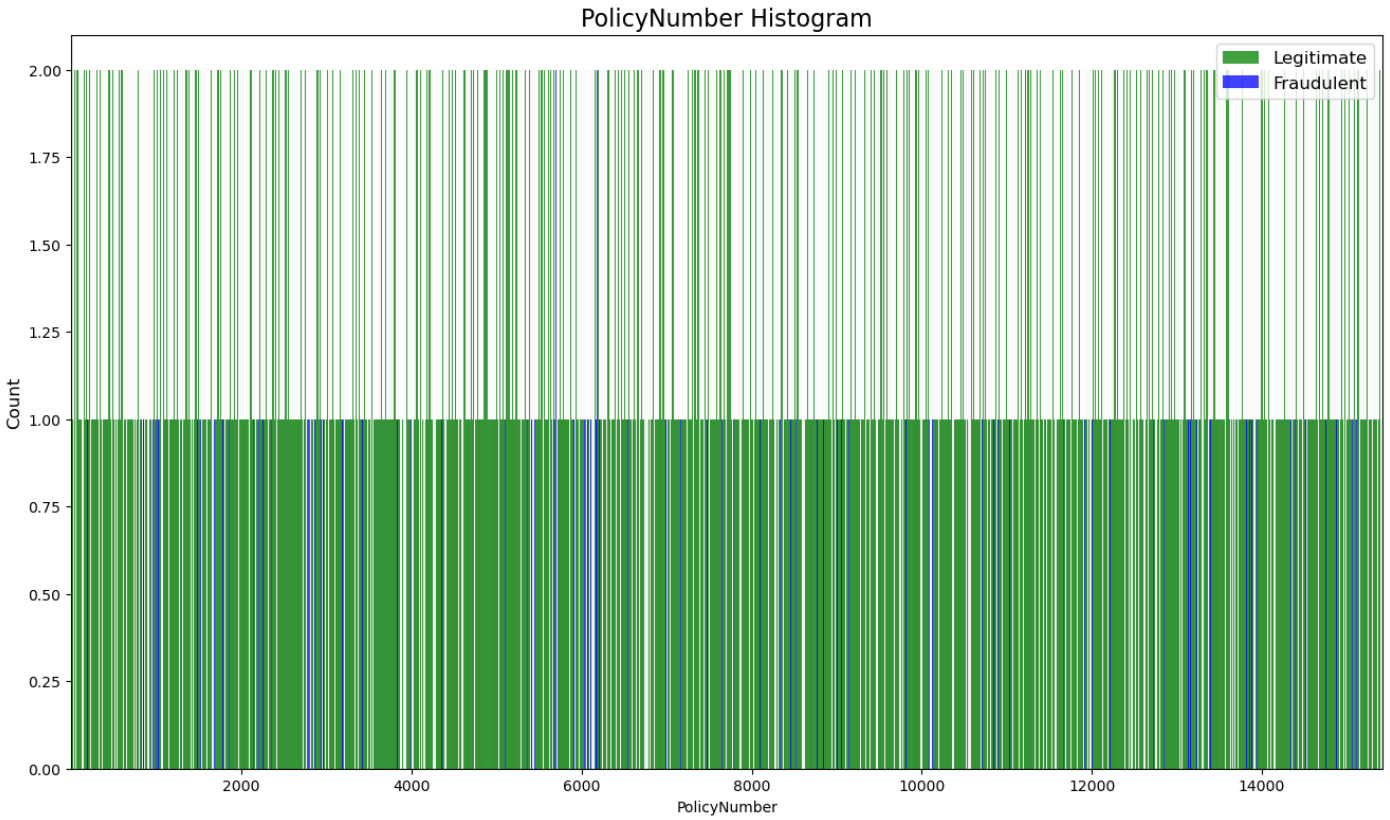

MonthClaimed Histogram
Legitimate
Fraudulent
Count
1000
800
600
400
200
0
Jan
Nov
Jul
Feb
Dec
Mar
Aug
May
Apr
Jun
Oct
Sep
MonthClaimed

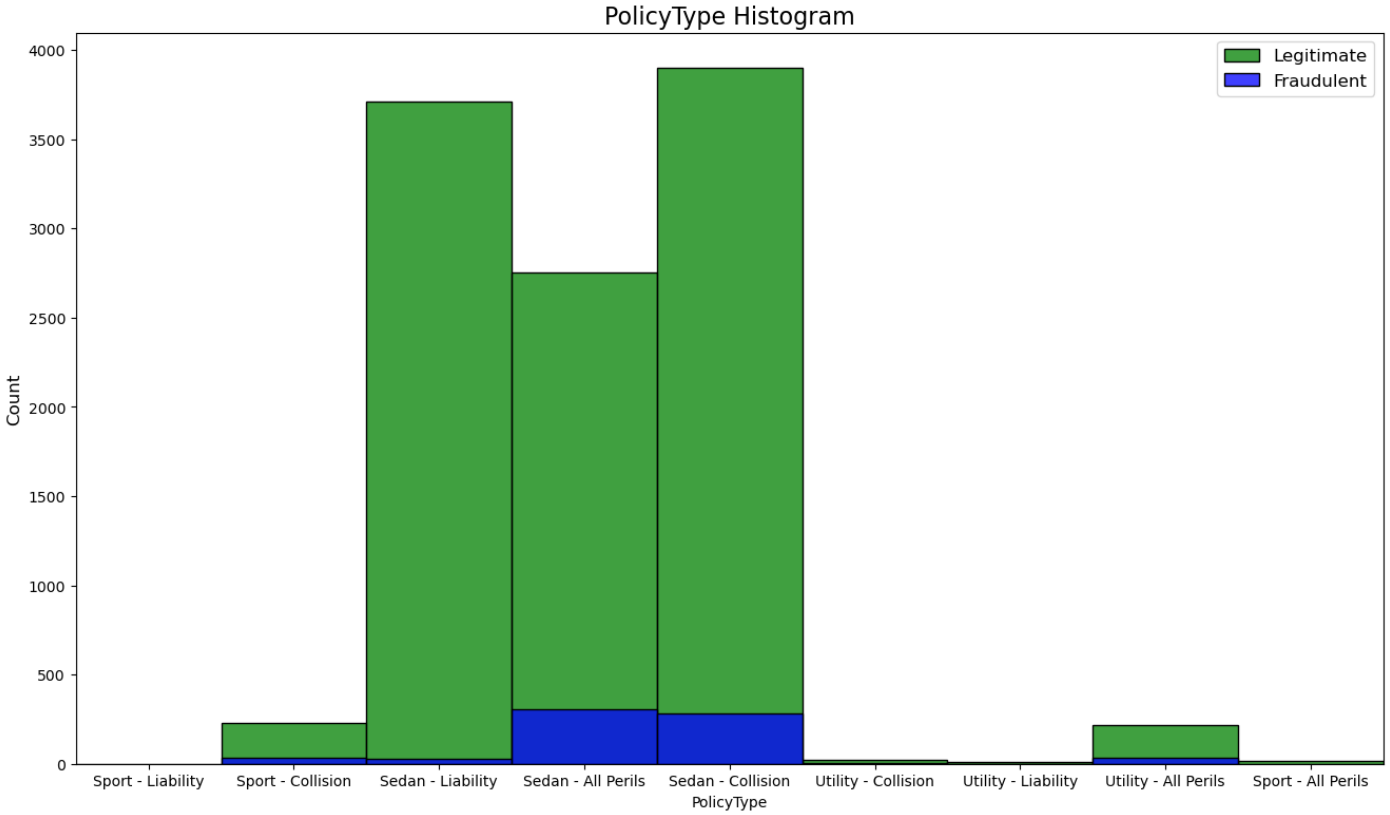

NumberOfCars Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
3 to 4
1 vehicle
2 vehicles
5 to 8
more than 8
NumberOfCars

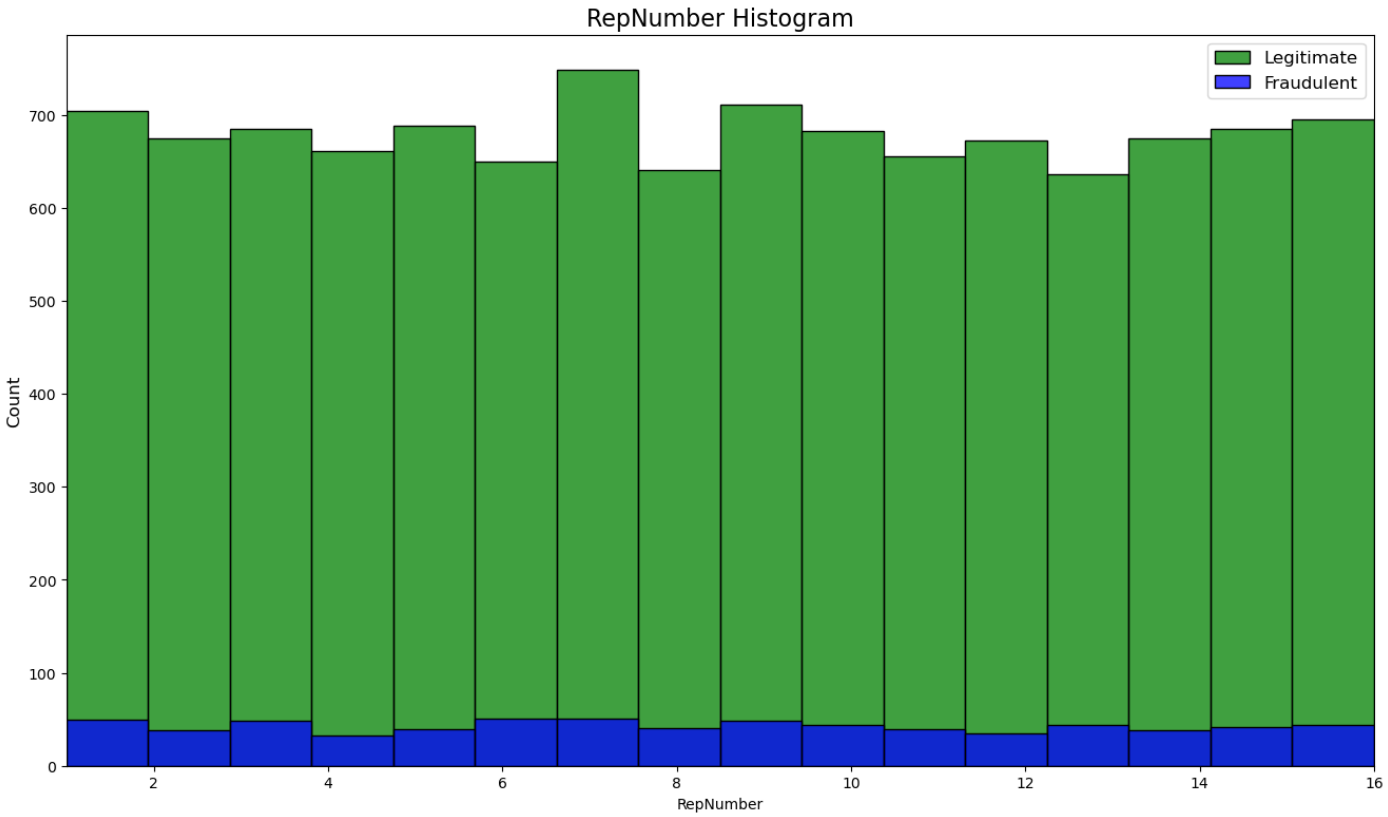

NumberOfSuppliments Histogram
Legitimate
Fraudulent
Count
5000
4000
3000
2000
1000
0
none
more than 5
1 to 2
3 to 5
NumberOfSuppliments

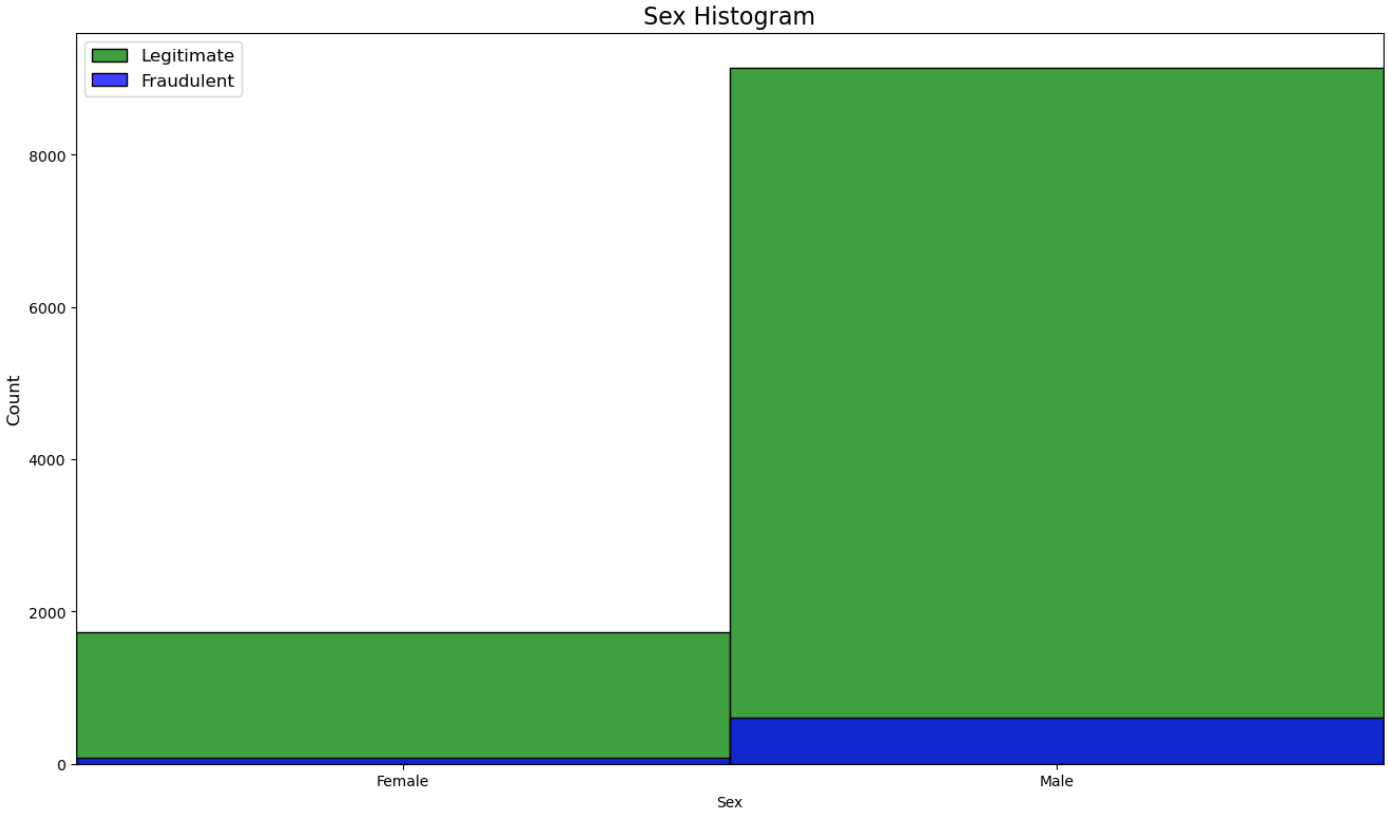

PastNumberOfClaims Histogram
Legitimate
Fraudulent
Count
4000
3500
3000
2500
2000
1500
1000
500
0
none
1
2 to 4
more than 4
PastNumberOfClaims

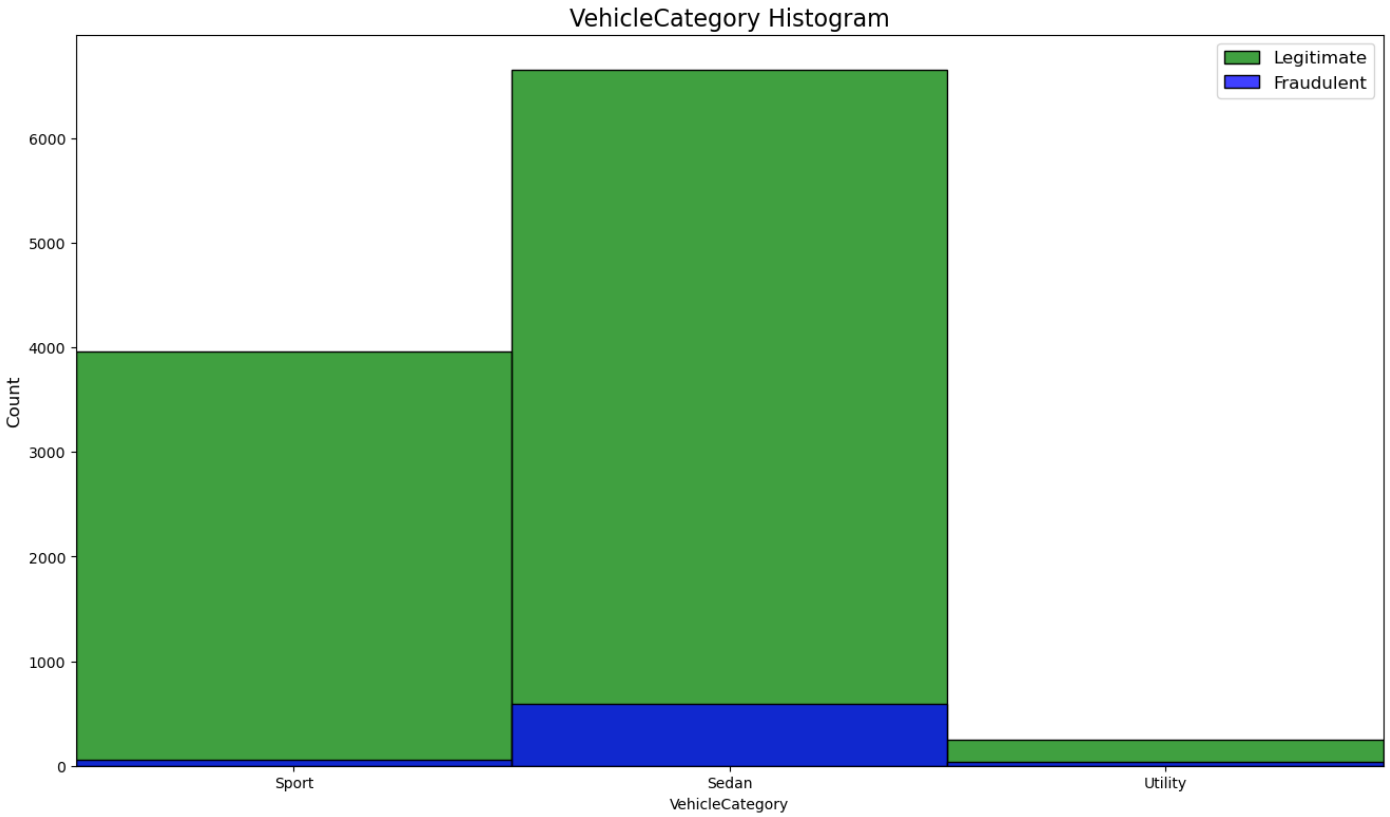

PoliceReportFiled Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
No
Yes
PoliceReportFiled

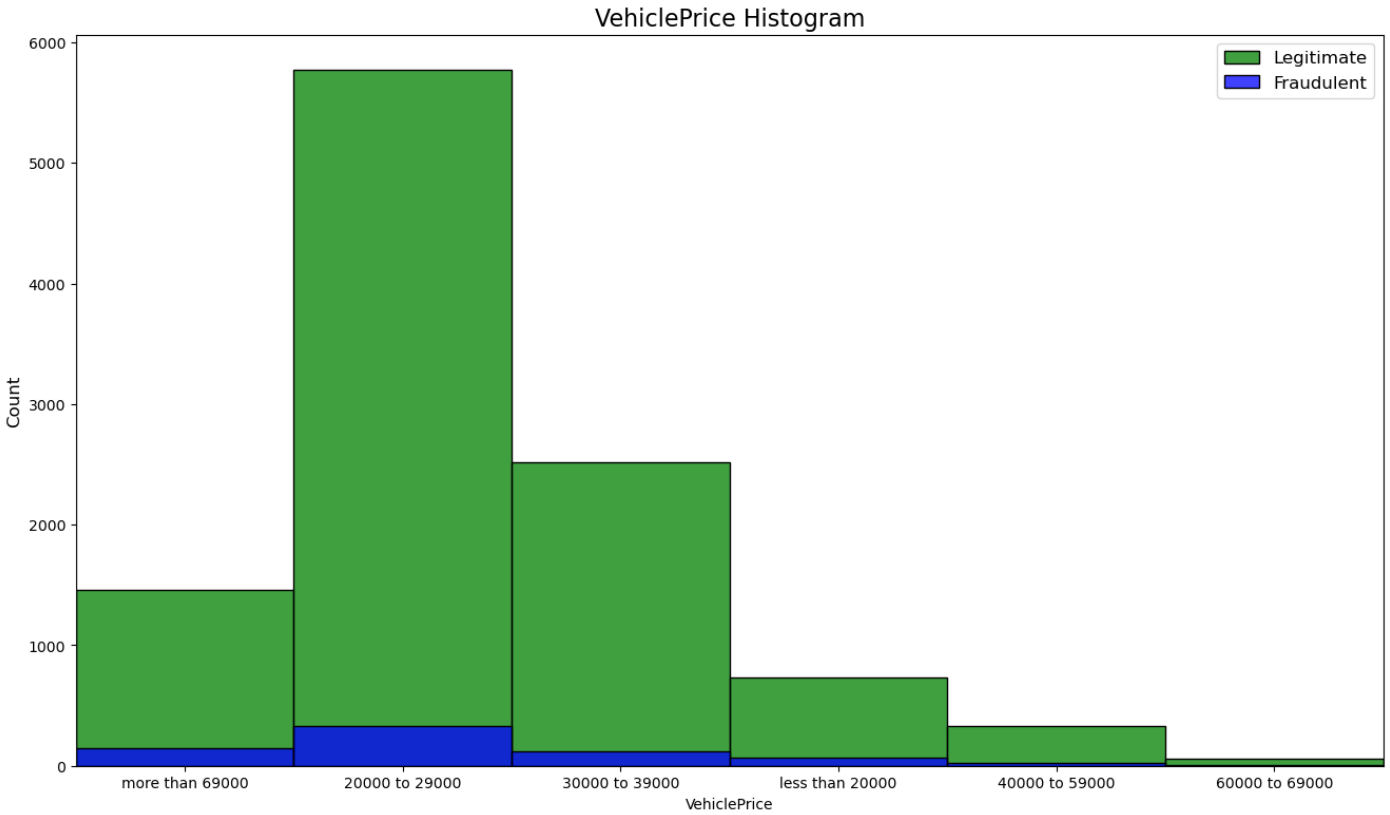

PolicyNumber Histogram
Legitimate
Fraudulent
Count
PolicyNumber
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
2000
4000
6000
8000
10000
12000
14000

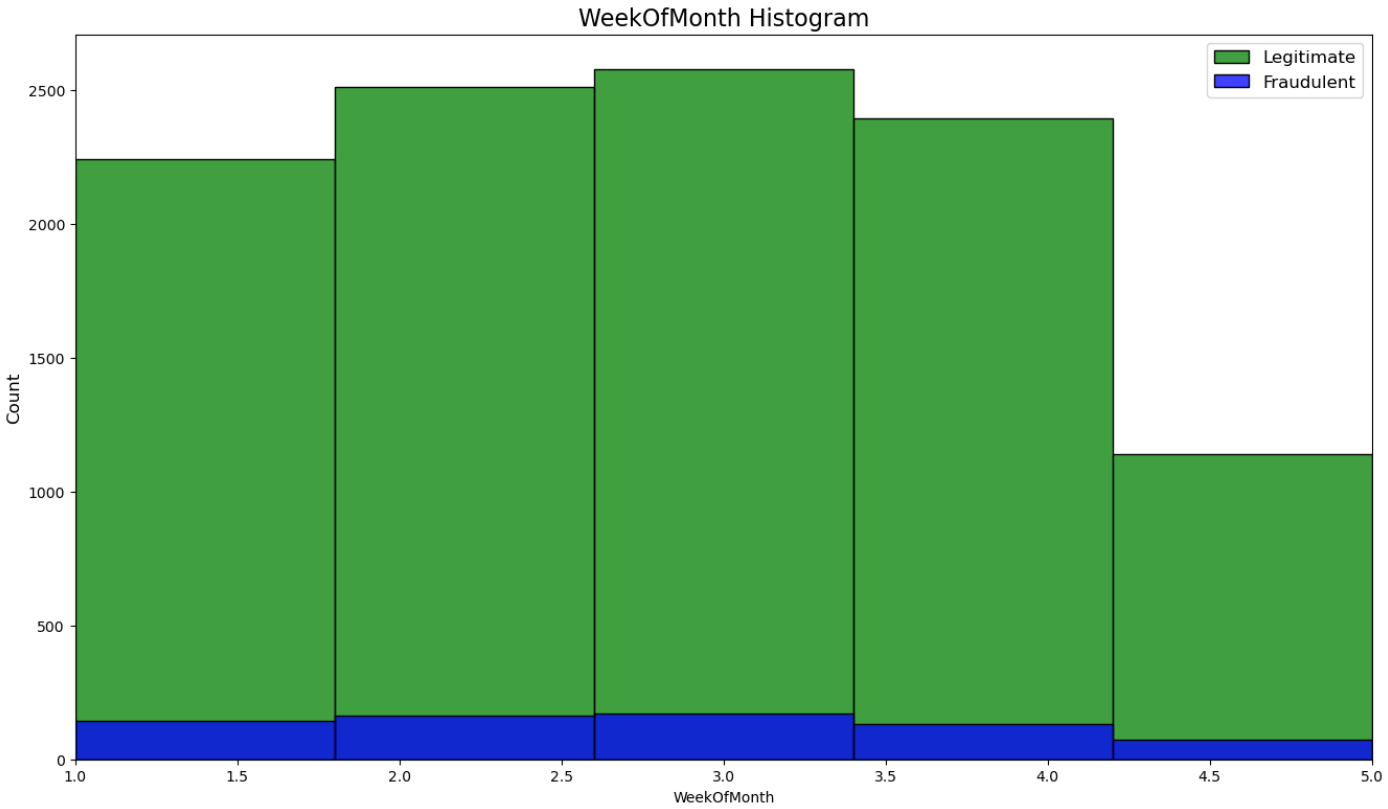

PolicyType Histogram
Legitimate
Fraudulent
Count
PolicyType
0
500
1000
1500
2000
2500
3000
3500
4000
Sport - Liability
Sport - Collision
Sedan - Liability
Sedan - All Perils
Sedan - Collision
Utility - Collision
Utility - Liability
Utility - All Perils
Sport - All Perils

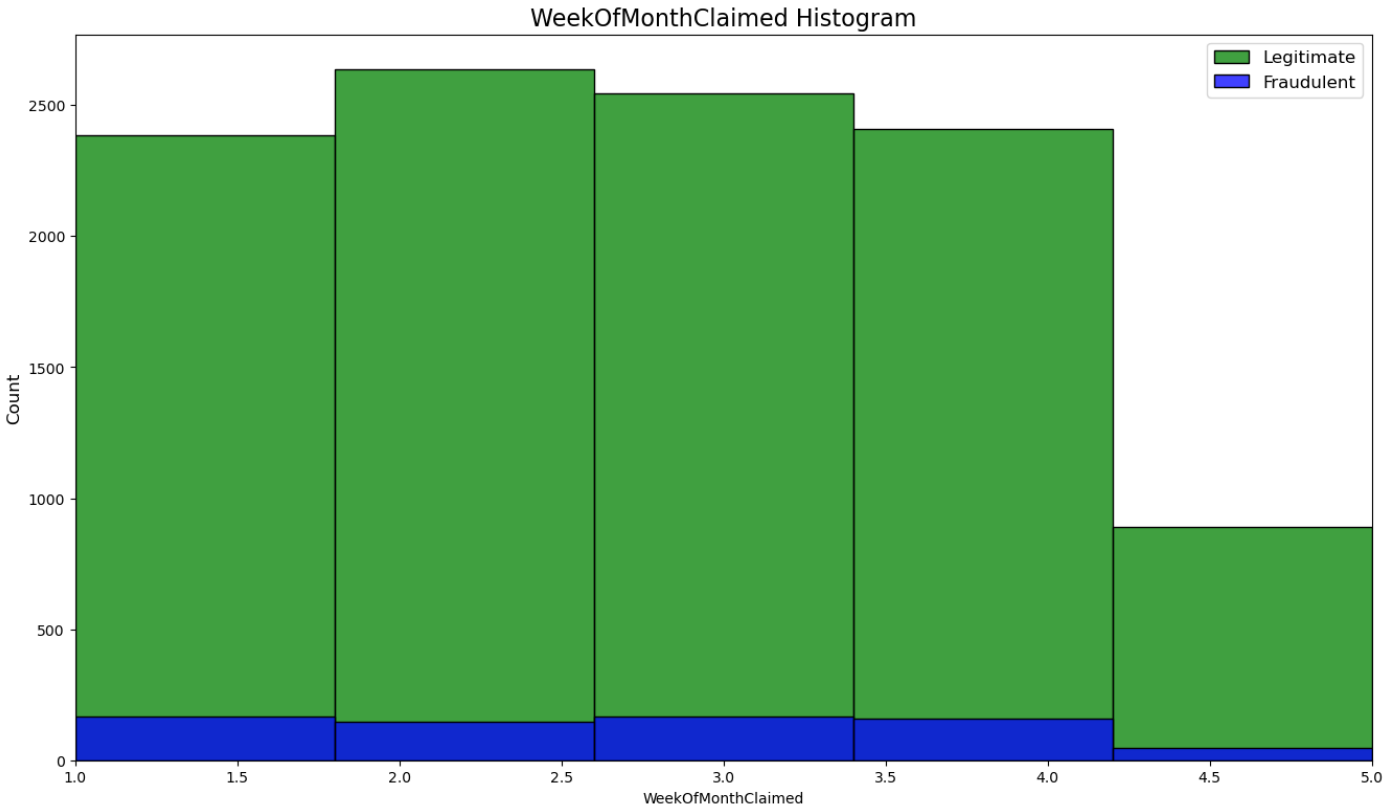

RepNumber Histogram
Legitimate
Fraudulent
Count
RepNumber
0
100
200
300
400
500
600
700
2
4
6
8
10
12
14
16

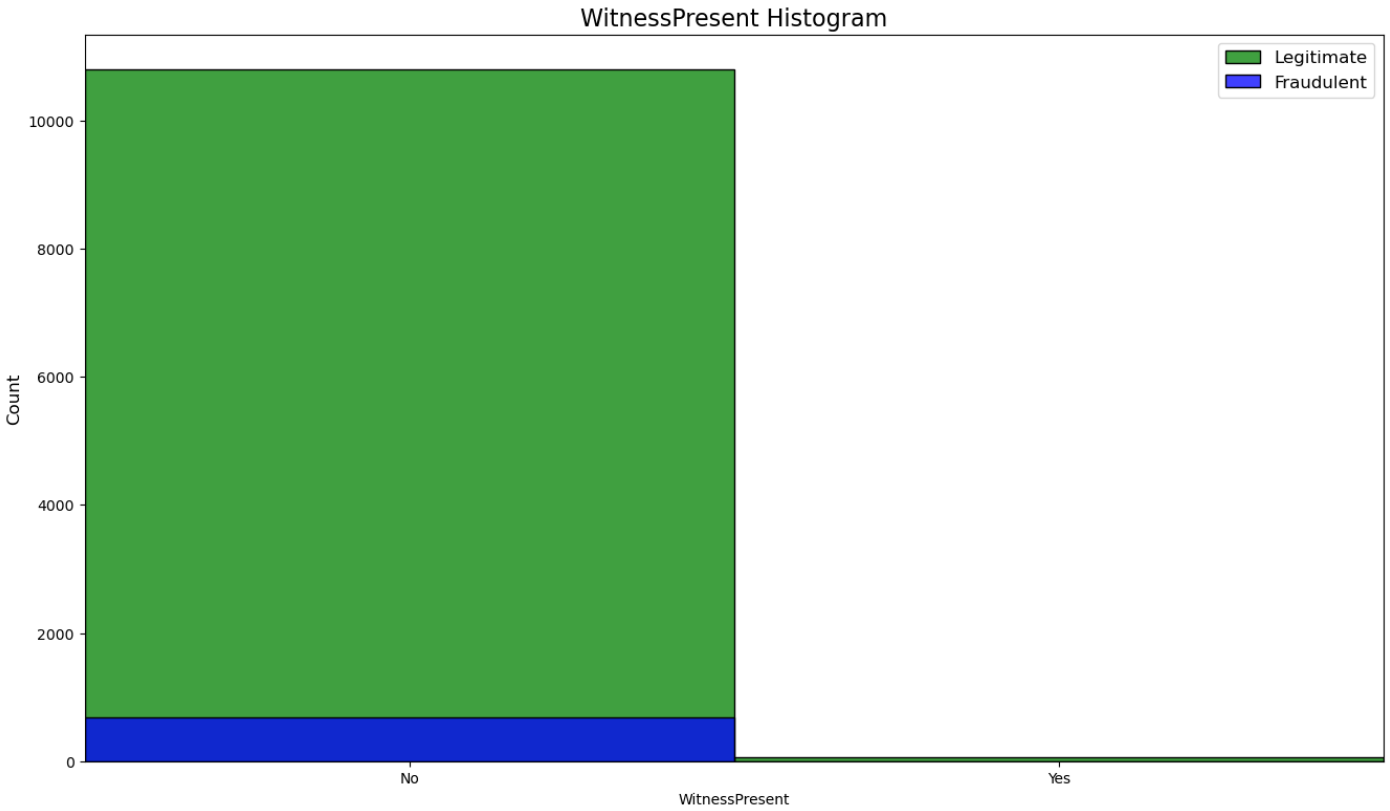

Sex Histogram
Legitimate
Fraudulent
Count
Sex
0
2000
4000
6000
8000
Female
Male

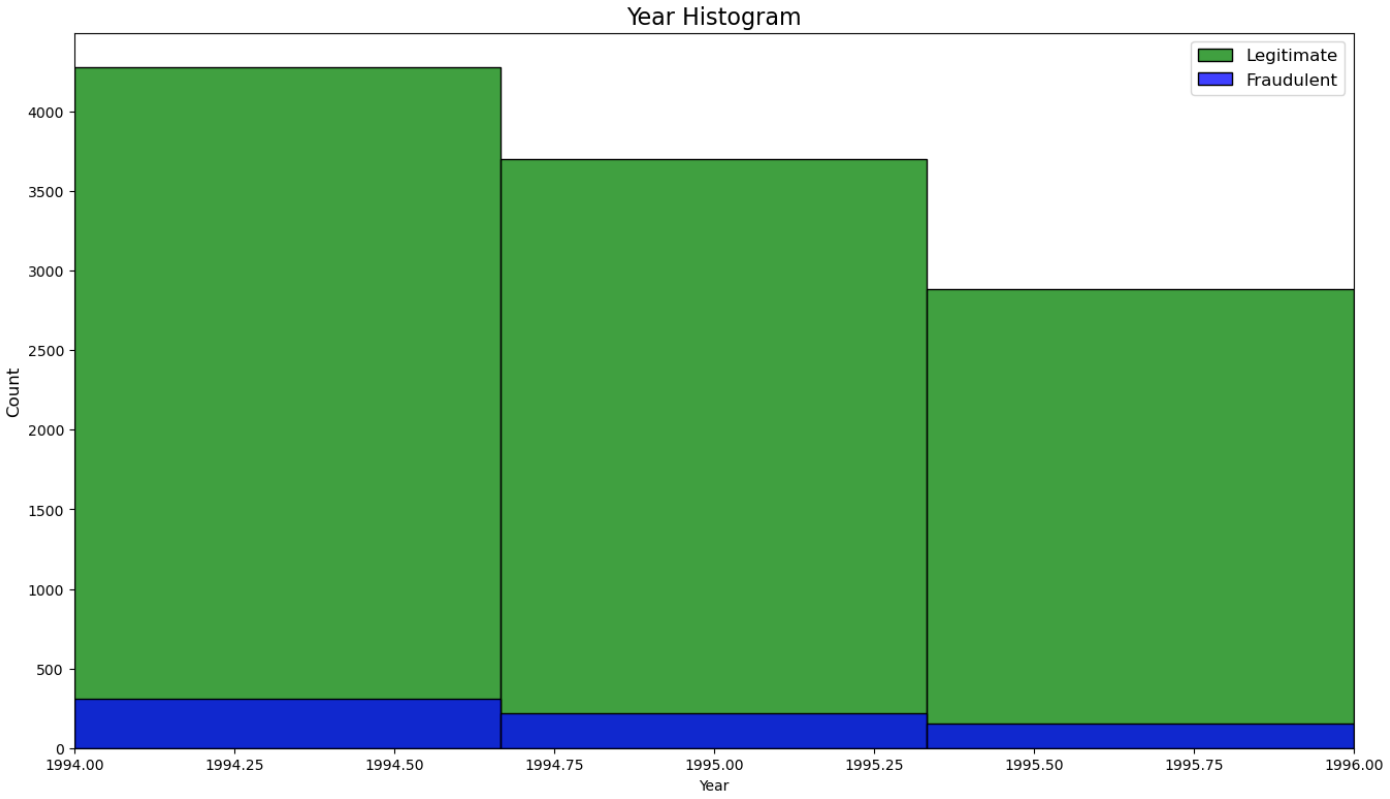

VehicleCategory Histogram
Legitimate
Fraudulent
Count
0
1000
2000
3000
4000
5000
6000
Sport
Sedan
Utility
VehicleCategory


VehiclePrice Histogram
Legitimate
Fraudulent
Count
0
1000
2000
3000
4000
5000
6000
more than 69000
20000 to 29000
30000 to 39000
less than 20000
40000 to 59000
60000 to 69000
VehiclePrice


WeekOfMonth Histogram
Legitimate
Fraudulent
Count
WeekOfMonth
0
500
1000
1500
2000
2500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


WeekOfMonthClaimed Histogram
Legitimate
Fraudulent
Count
WeekOfMonthClaimed
0
500
1000
1500
2000
2500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


WitnessPresent Histogram
Legitimate
Fraudulent
Count
10000
8000
6000
4000
2000
0
No
Yes
WitnessPresent


Year Histogram
Legitimate
Fraudulent
Count
4000
3500
3000
2500
2000
1500
1000
500
0
1994.00
1994.25
1994.50
1994.75
1995.00
1995.25
1995.50
1995.75
1996.00
Year