

|  |
| --- |
| Insurance Claims  Fraud Detection |
|  |
| November 2  Glenn Nigel Ebenezer  Batch No 1844 |



Contents

[Insurance Claims- Fraud Detection 3](#_Toc118133820)

[Problem Statement: 3](#_Toc118133821)

[I Problem Definition 4](#_Toc118133822)

[II Data Analysis 6](#_Toc118133823)

[III Exploratory Data Analysis(EDA) 7](#_Toc118133824)

[IV Pre-processing Pipeline 20](#_Toc118133826)

[V Building Machine Learning Models 23](#_Toc118133827)

[VI Hyper parameter Tuning: 25](#_Toc118133828)

[VII Metrics: 27](#_Toc118133829)

[VIII Summary: 29](#_Toc118133830)

[IX Acknowledgements 30](#_Toc118133831)

# Insurance Claims- Fraud Detection

|  |
| --- |
| Problem Statement: Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem.  In this project, you are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.  In this example, you will be working with some auto insurance data to demonstrate how you can create a predictive model that predicts if an insurance claim is fraudulent or not. |
| *“Solution: A Classic Classification model to be used to detect whether the claim is Fraudulant or Genuine”* |
| [Link to the Dataset provided:](https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/blob/master/Automobile_insurance_fraud.csv)  [Github Link to the Jupyter Notebook](https://github.com/glennnigel/GNE_Datatrained1844/blob/main/Insurance%20Claims-%20Fraud%20Detection%20(3).ipynb) |

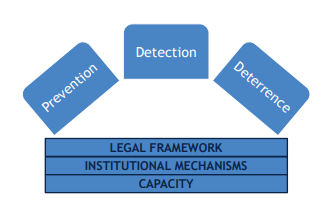
# I Problem Definition

One of the most challenging problems in the niche of insurance is fraudulent claims, and the ability to detect whether the claim is a fraud or genuine case, over the years, was and is still a herculean task. Although, in these modern times, with the advent of Machine learning (ML) in Datascience, we have the capability to detect these frauds with the help of ML by studying trends and data analysis.

The Solution for the problem statement requires us to figure out the best and most efficient way to build a model using ML to automatically, study the data and give us a fraud alert whenever it is detected. With the number of insurance policies and claims that are out there, it is very important that a fraudulent claim be flagged in time.

As we know, it is merely impossible to achieve a 100% accurate model that is efficient and not cost the insurance company all its profits. We can at least ensure we build a model that can red flag any suspicious case, even if it’s a false positive, the number of cases for manual scrutiny would drastically reduce.

In India alone, the Insurance Regulatory and Development Authority (IRDA), stated on record that Insurance companies lose over USD 6.25 billion to frauds which results in higher premiums for genuine consumers. And hence devised an “Anti-Fraud Conceptual Framework” which rests on 3 Pillars, of which Detection forms the main Cornerstone. It is only with the right, and timely detection that can anything else, like prevention or deterrence, or the other institutional and legal mechanisms come into play.



Insurance companies that haven’t implemented high-tech solutions yet should hurry up. Artificial intelligence (AI) and predictive analytics are shaping the future of the entire industry, giving significant competitive advantages to organizations that are already making use of these technologies.

In this article, we’ll dive into the newest insurance fraud prevention methods, uncover the benefits of predictive analytics, and feature the three best insurance claims fraud detection software tools.

Top 3 Insurance Claims Fraud Detection Software

* FRISS Fraud Detection
* SAS Detection and Investigation for Insurance
* SEON

Although these are more detailed and elaborate programs out there, can be used when the stakes are high, but a far less expensive with decent accuracy maybe built and used when the stakes are rather low. This is what we will consider doing today using simple Machine Learning strategies in Python.

In this prediction model, we will be using basically 4 simple classification models:

* Decision Tree Classifier
* Linear Regression
* K Nearest Classifier
* Random Forest Classifier

We will also consider Hyperparameter tuning, to increase the efficiency and otherwise overall effectivess by knowing the actual scores incase the data/model led the result to over or under fitting.

It is a given that, as much as we try and get the best success rates, it all boils down to the data collection and manipulation. As Data Scientists we can do very little when it comes to the data that is given to us, until and unless we are scraping it, of course.

But Data Manipulation and processing is the main factor attributing to success. And along with a skillset in python, it will be really helpful if we have knowledge on the topic we are working on, also known as –Domain Knowledge.

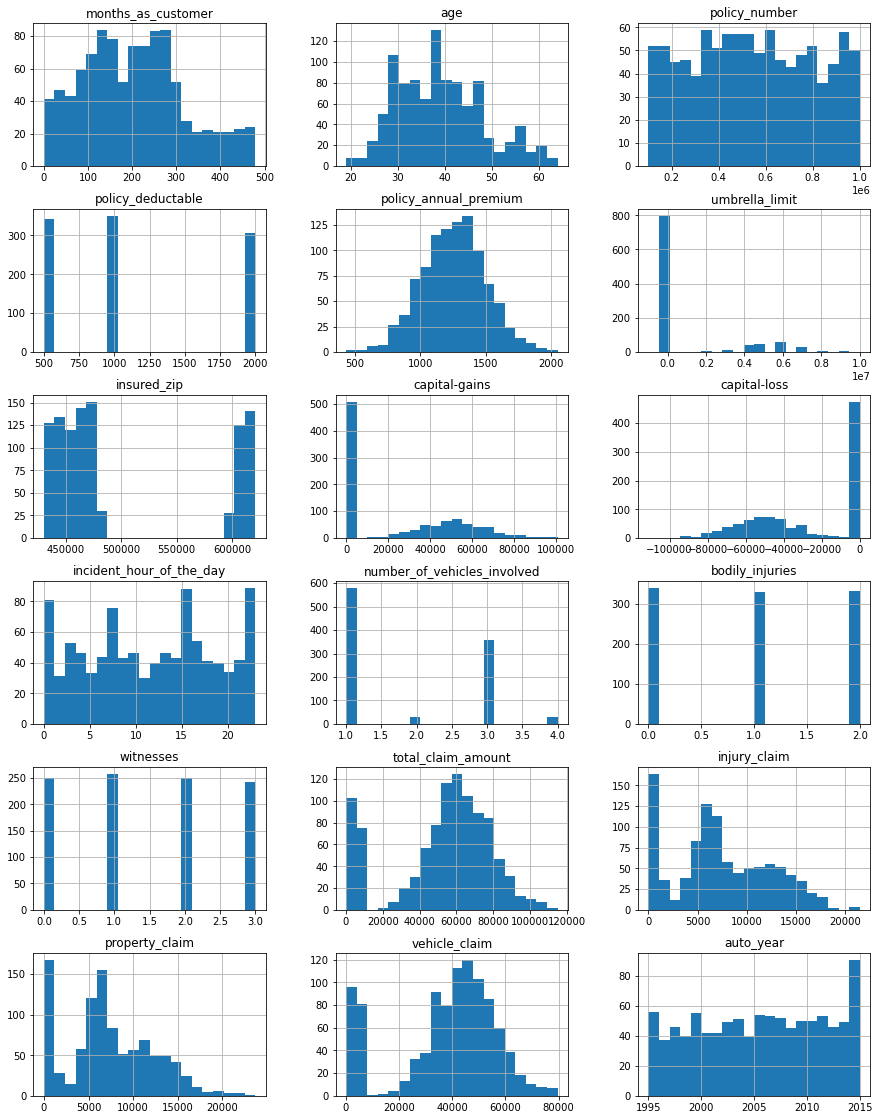
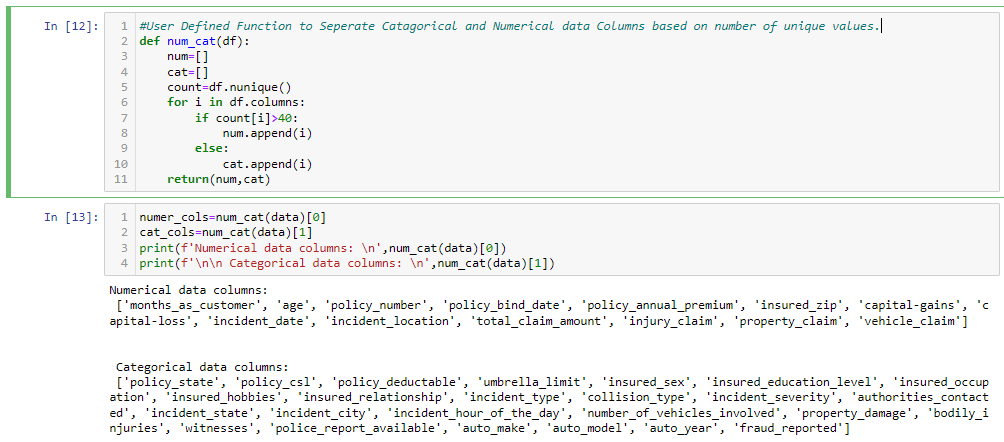
One of the reasons for having picked this project to draft this detailed report is because of the Domain knowledge I have acquired, having cleared the IRDA Entrance exam multiple times.

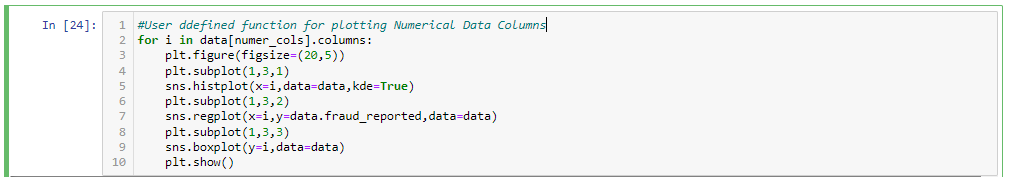
# II Data Analysis

* The Dataset was imported into the Jupyter Notebook as a CSV file, and on initial inspection, we saw that there were 40 columns and 1000 records. The Columns were a mix of Float, Int and object datatypes and there were no missing or null values.
* On studying the standard deviation from the mean in the columns with int and float dtype, it was clear that there were a few outliers as the standard deviation was disproportionate. Added to this the increments between the quartiles also were uneven in some columns. This anyway was checked and attended to at a later stage.
* There was one column that was completely empty, also a nominal data columns like “policy numbers” that can be removed.
* Comparing the categorical and Numerical Data columns, it was clear that python had on the basis of their Dtype had predicted if it was an object or non-object, which in this case would not work, as there were columns with limited numerical values, that were actually categorical columns. This too needed to be addressed.
* The 'insured\_zip’ column, although not nominal, had almost all unique values, and may not contribute well towards the model building, maybe removed after confirmation via EDA.
* Looking at the Dataset and the columns and column data, along with the number of unique values in each column, we were able to estimate a threshold at which we can categorize a column as categorical or not based on whether it contains lesser than the threshold of 40 unique values.
* Finally, although there were no null or missing values, there were a lot of ”?” and “0” in the columns that needed to be attended to as well.

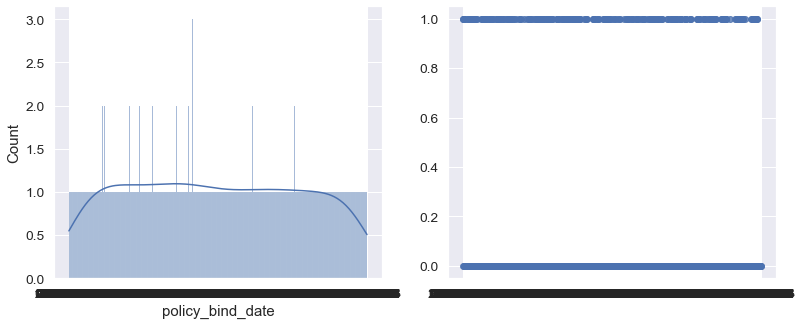
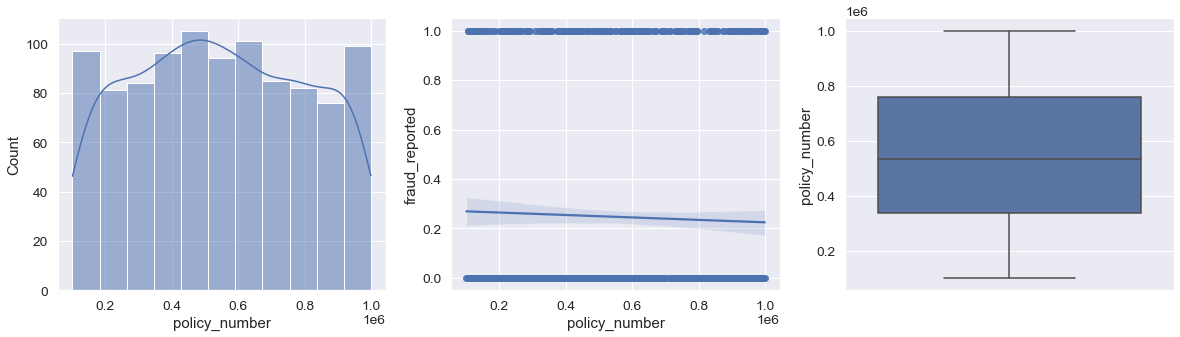
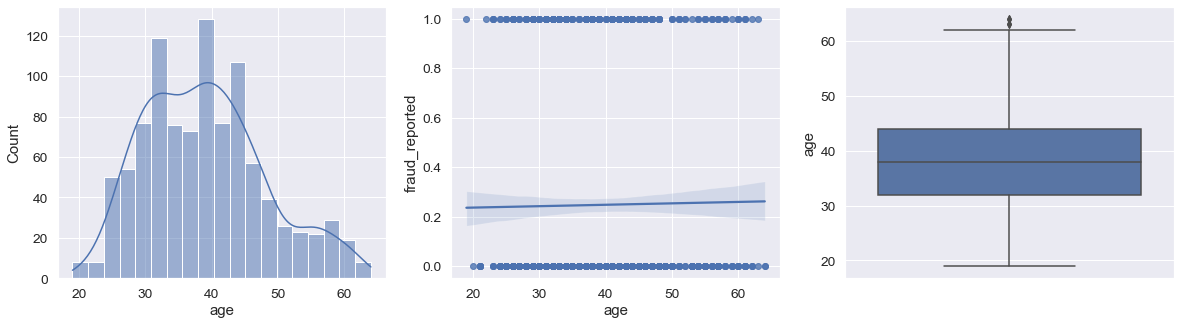
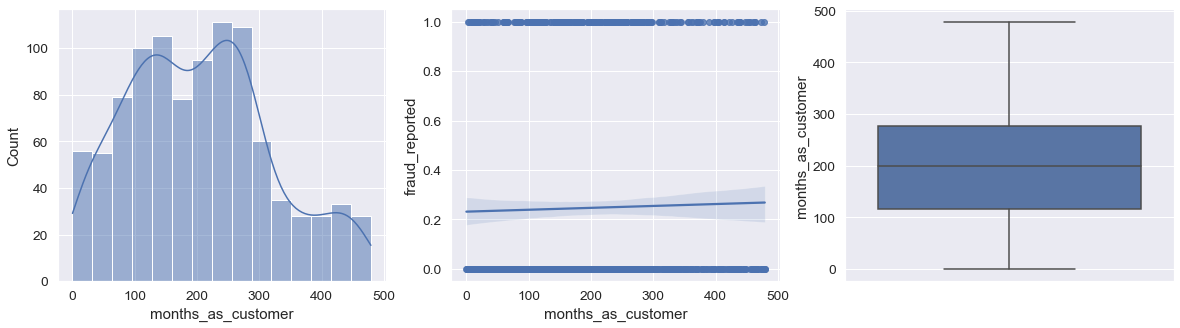
# III Exploratory Data Analysis(EDA)

# -using Visualisation Techniques

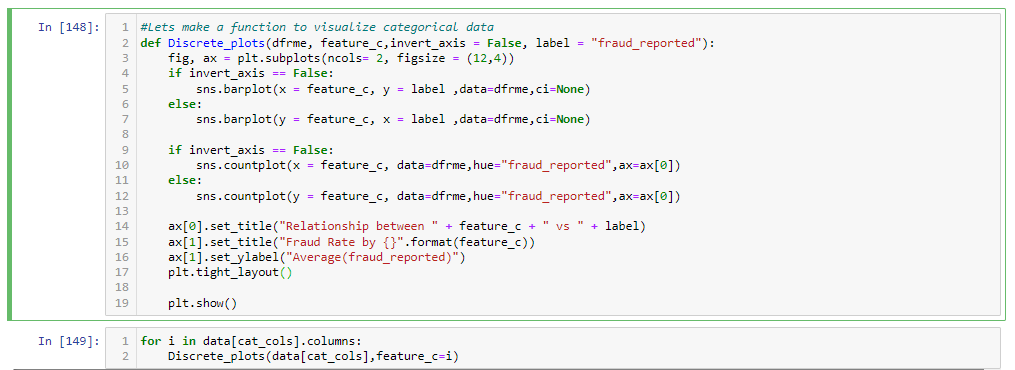
* We first plot a Histogram plot of all columns
* We then began the EDA be separating the Categorical and Numerical data columns, based on the number of unique values.
* Based on this we plotted a Regplot, Boxplot and Histplot of all the Numerical data Columns using another user defined function.



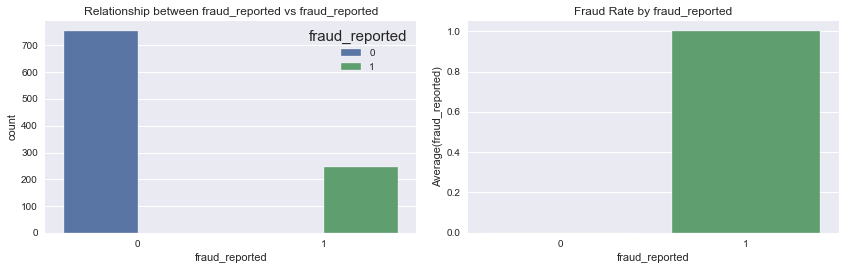
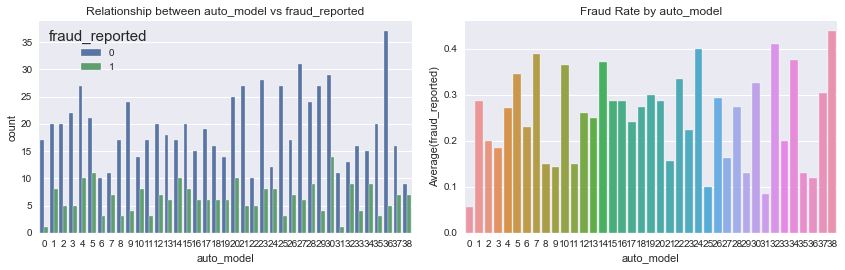
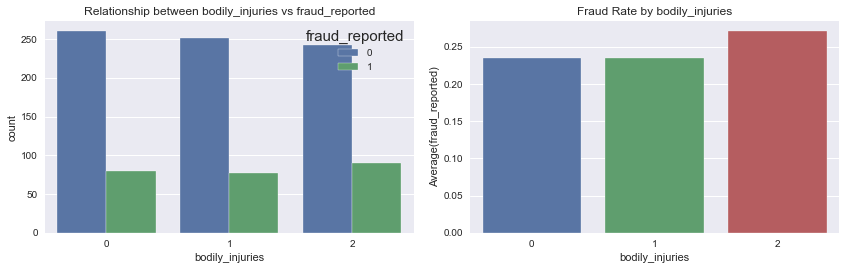
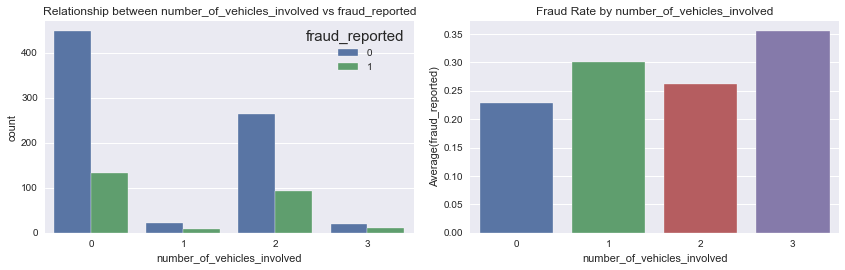
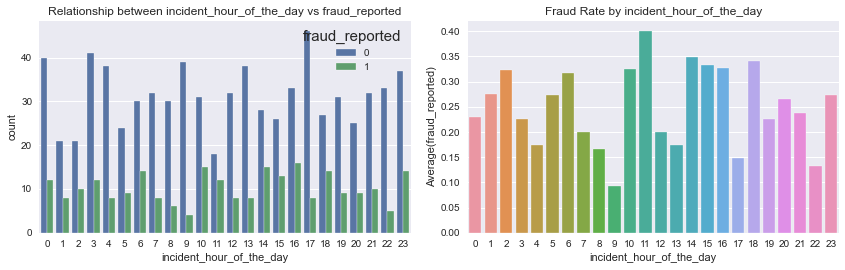
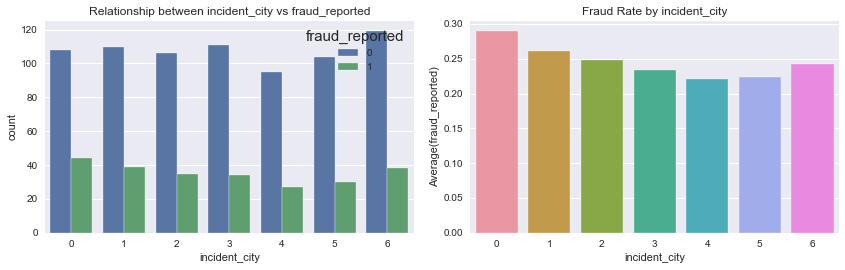
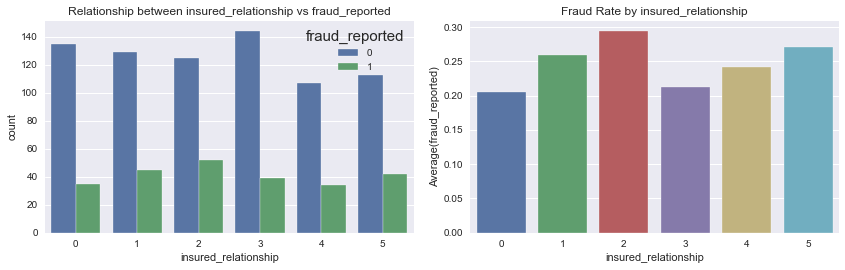
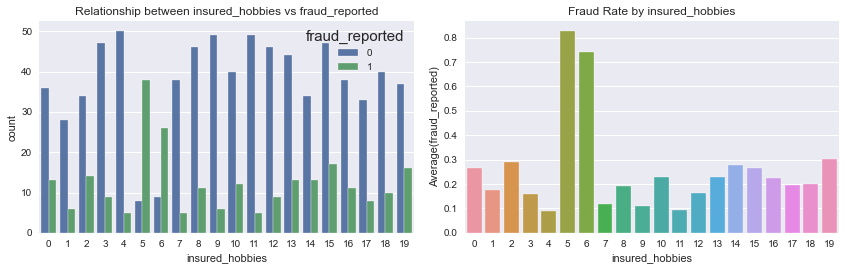
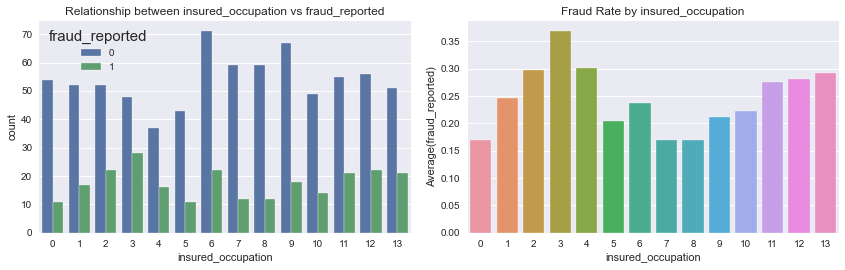
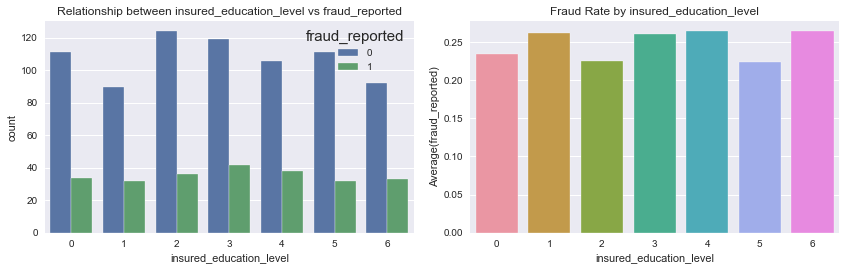
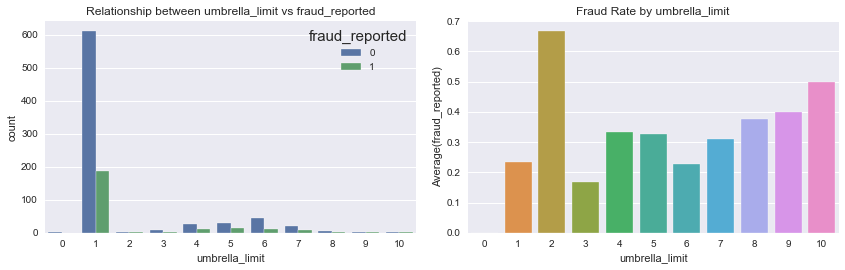
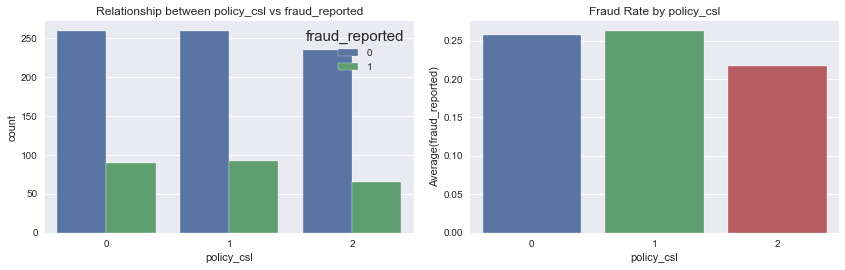
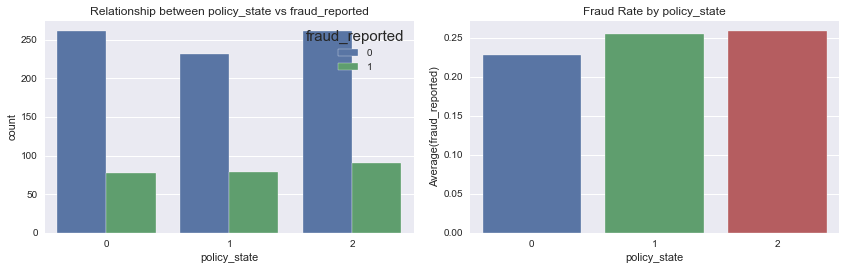
Results shows below

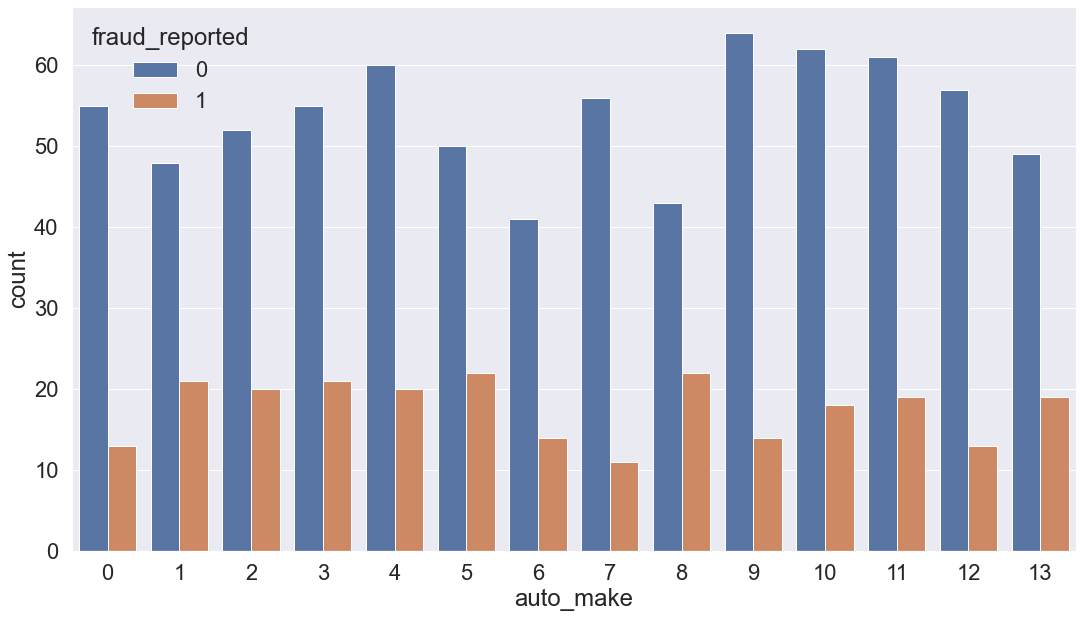
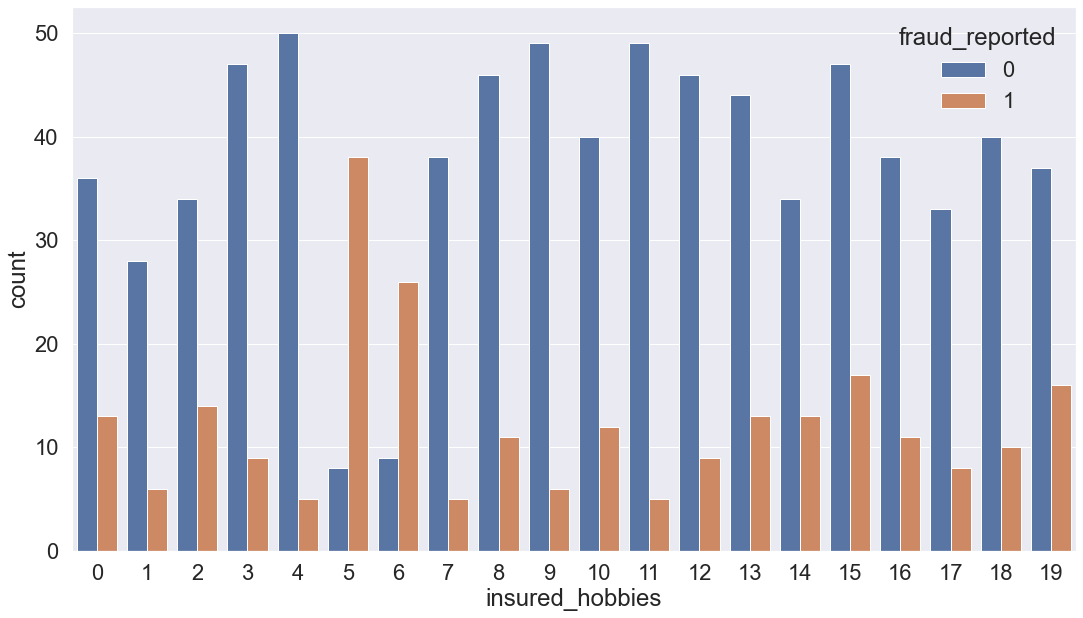
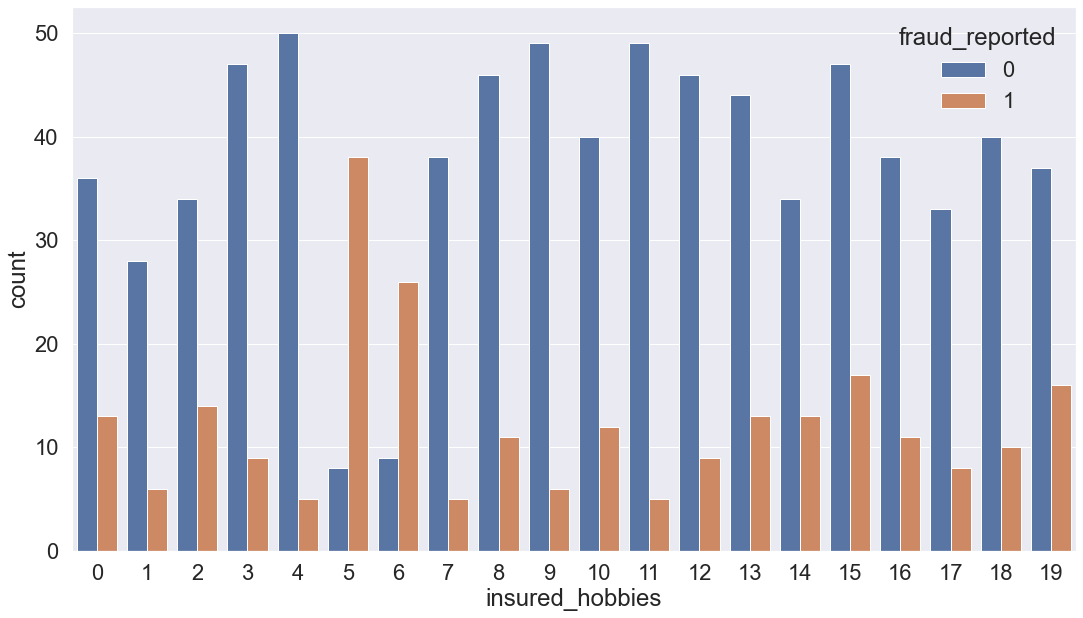
from the above it is clear that it is clear that none of the numerical columns, except 'vehicle\_claim', 'property\_claim','injust claim','total\_claim\_amount' have any relationship with the target.

* Next we will plot the Categorical data columns using Countplot to understand the relationship between the feature and label and barplot to understand the attributes in the feature.



Below are the generated plots:

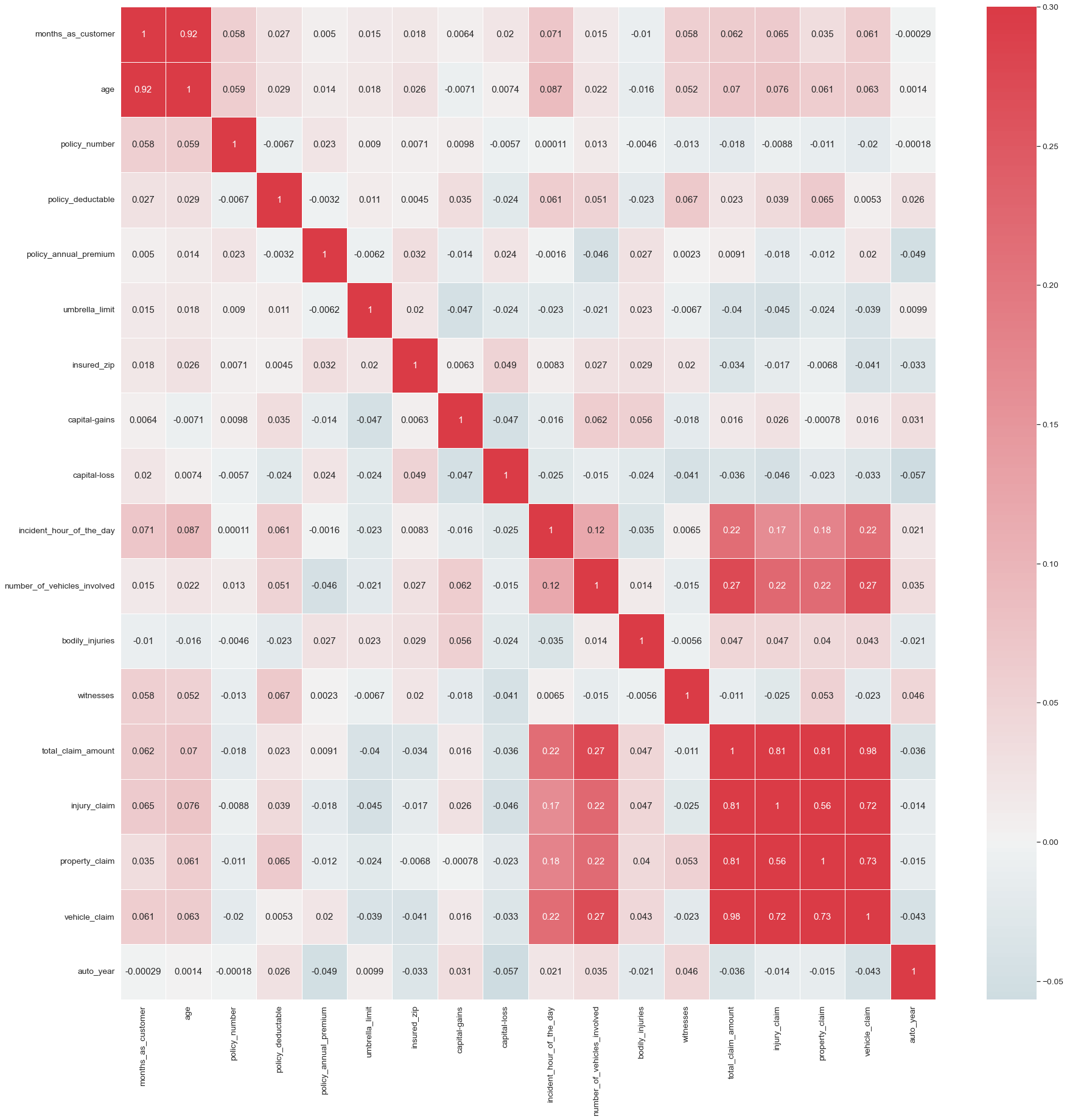


* Next, a few more comparisons were made to see how these features affect the target Label.

It is found that these features aren’t exactly related to the target label.

* Next we can look at the Correlation Plot: (Displayed on the next page)

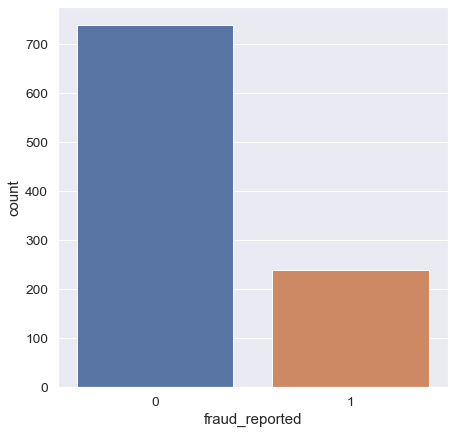
1. We can see that total\_claim\_amount','injury\_claim', 'property\_claim', 'vehicle\_claim' are correlated not only with the target, but also with each other.
2. Same with 'incident\_type', 'collision\_type', 'incident\_severity', 'authorities\_contacted', 'incident\_state', 'incident\_city', 'incident\_location', 'incident\_hour\_of\_the\_day','number\_of\_vehicles\_involved. they are correlated with each other
3. The 'collision\_type', 'incident\_severity' is correlated with the target.
4. Rest of the features have low correlation.
5. These findings were also suggestive in the scatterplot. which means we can consider dropping the other numerical data columns.



In the above correlation heatmap, we can see all the features that have a relationship with the target label. In case the correlation coefficient is found negative, it means that there is a negative correlation between the feature and the Target Label.

* Next, we check to see if the Dataset has a balanced set of records with regard to the Target Label. (Attaching the plot below)

We checked to see if SMOTE would remove the Imbalances, which it did, via Oversampling. We then used this technique while building model.



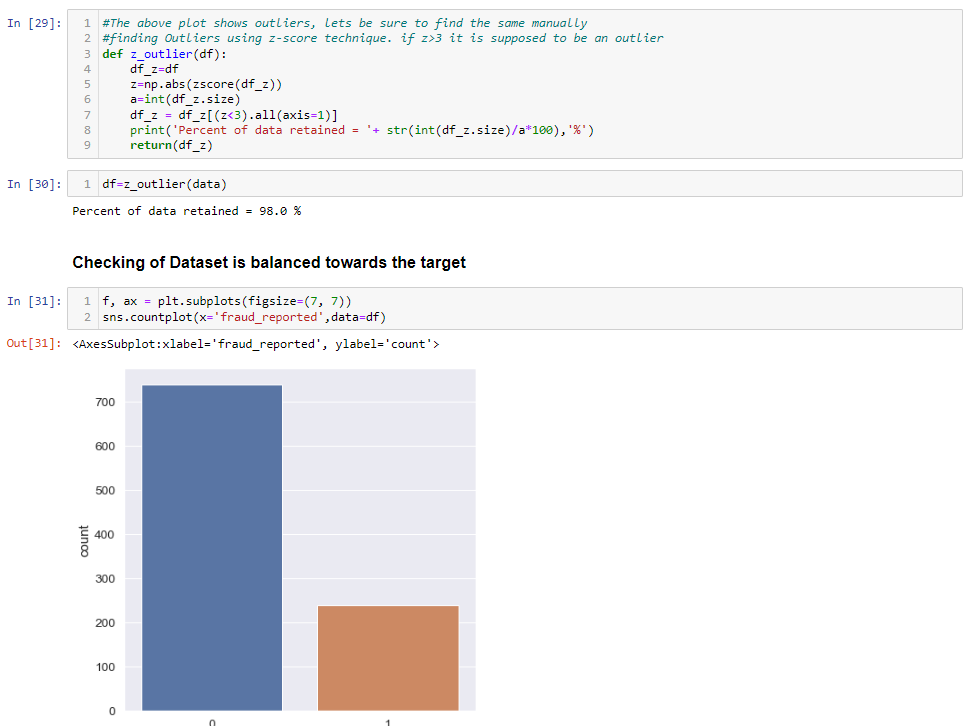
End of EDA.

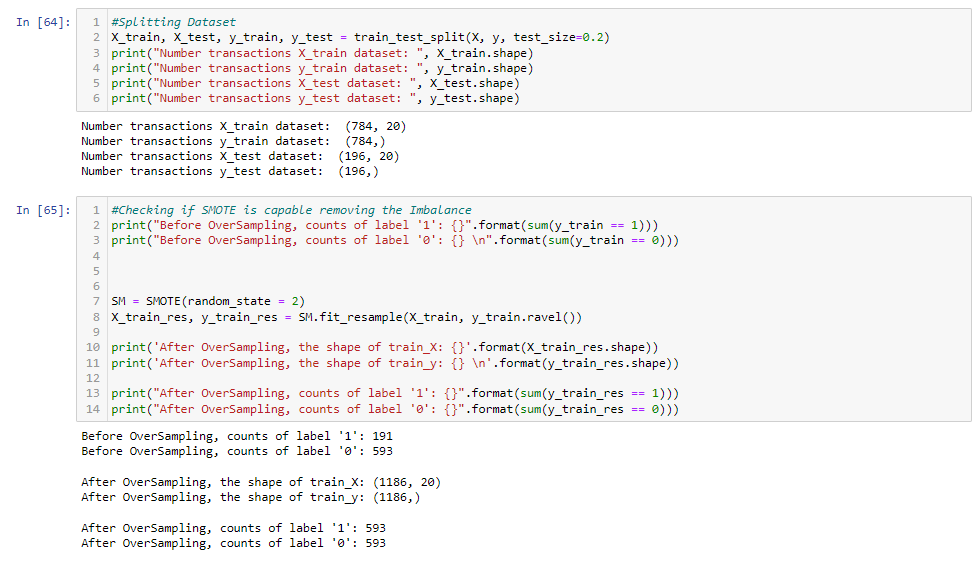
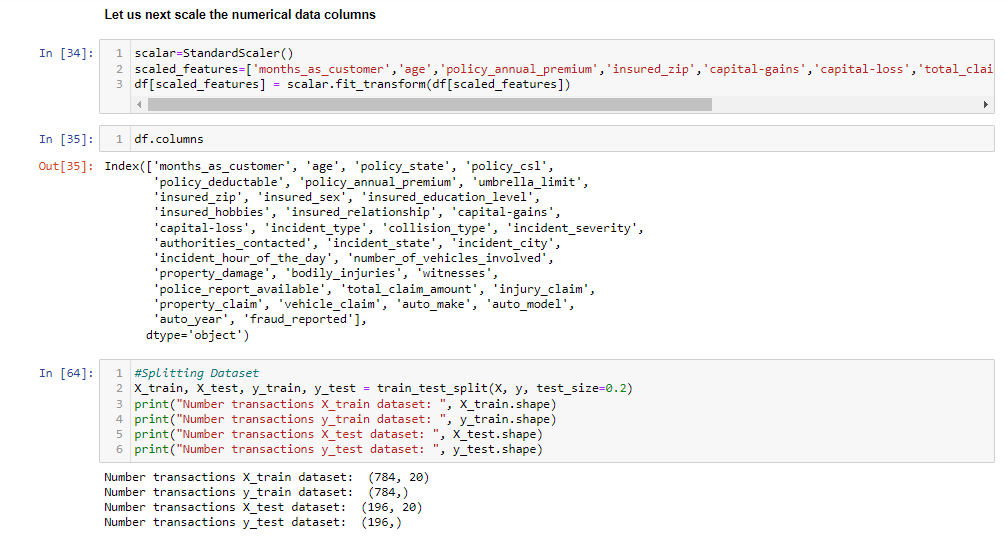
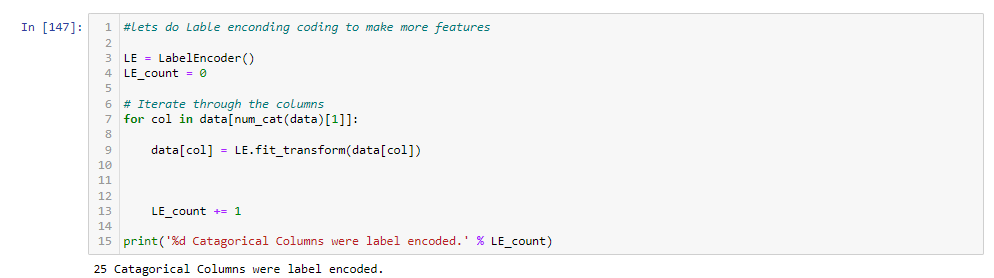
Note: During the process of plotting the categorical values, we had to also encoded the attributes of each categorical data column. This was done using Label Encoding.

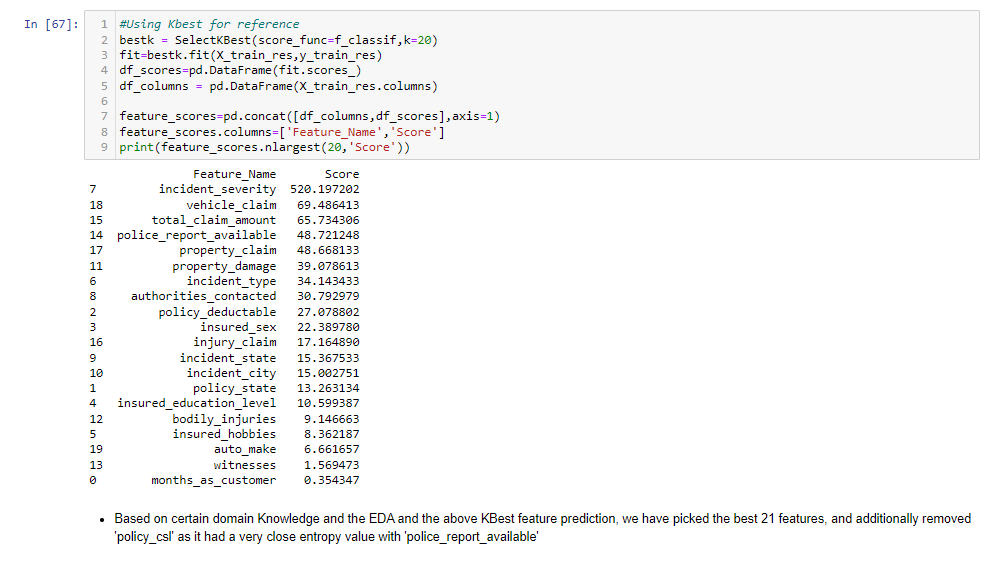
# IV Pre-processing Pipeline

* This involved dropping of the above mentioned nominal and empty columns.
* Removing all Outliers using Zscore technique.
* After separating the Categorical data columns, we chose to encode the attributes of the columns using Label Encoder.
* Used Standard Scaler to scale the numerical data columns.
* Applied SMOTE to remove imbalances in the dataset with regard to the Label.
* Split the Data into Training and Testing datsets.
* Used KBest Feature selection technique, to choose the best features to predict the target Label

Below are attached the codes for the same.







Based on the EDA, the KBest Feature Selection technique, and Domain knowledge, we removed 25 features, based on the entropy values.

* First we removed- 'insured\_zip','auto\_model','incident\_hour\_of\_the\_day','age','policy\_annual\_premium','umbrella\_limit','capital-loss','collision\_type','capital-gains','insured\_relationship','auto\_year','number\_of\_vehicles\_involved','policy\_csl'

on account of having the least entropy values.

* Using Domain Knowledge assisted by the common entropy values, or entropy values being too close, we decided to remove-

'months\_as\_customer','witnesses','auto\_make','incident\_city','property\_claim'

Thereby having a total of 15 features to predict the target label. They are:

'policy\_state', 'policy\_deductable', 'insured\_sex',

'insured\_education\_level', 'insured\_hobbies', 'incident\_type',

'incident\_severity', 'authorities\_contacted', 'incident\_state',

'property\_damage', 'bodily\_injuries', 'police\_report\_available',

'total\_claim\_amount', 'injury\_claim', 'vehicle\_claim'

# V Building Machine Learning Models

Using a User defined function, we automate a Model selection function that balances the data at every iteration using SMOTE Technique, and predicts the best model on the basis of accuracy score and the best random state. It also captures the incidents’ F1score.

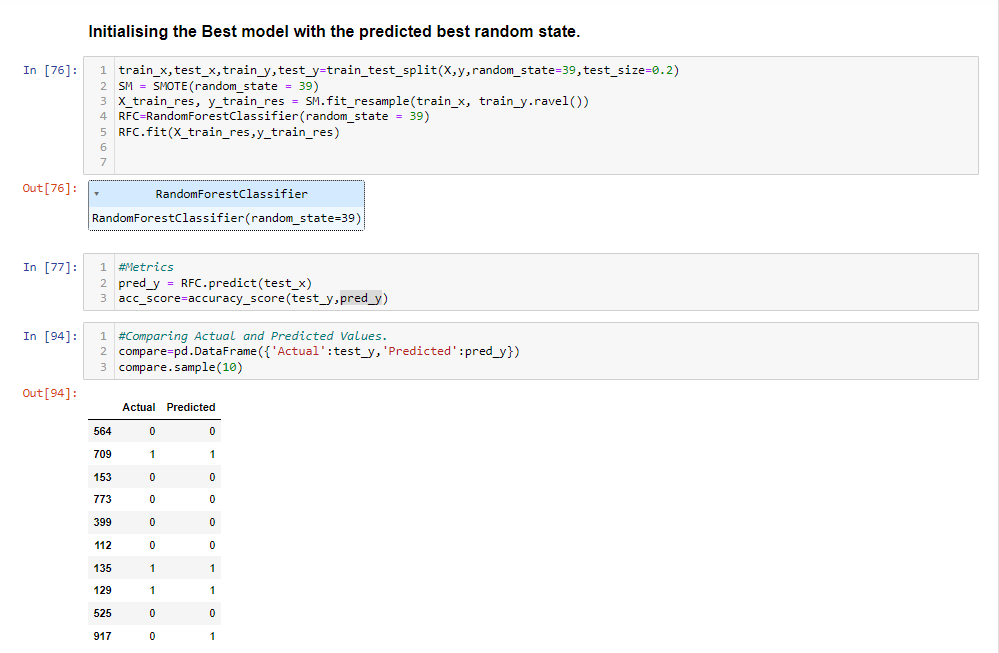
(Find the code below)



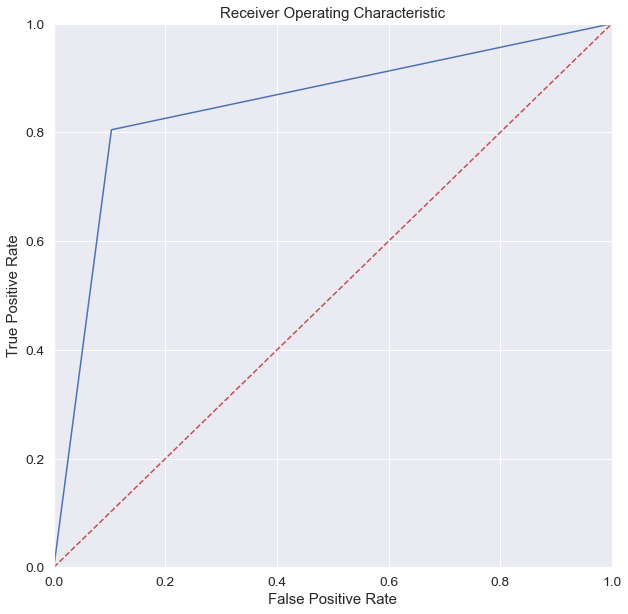
Based on the above prediction, we find that the best random\_state at 39, and Random Forest classifier giving us the best result of Accuracy score of 0.8622448979591837 and f1score of 0.6896551724137931.

Next: We build the model predicted above using the random state parameter.

The screenshot also contains a comparison between the actual and predicted Y samples.

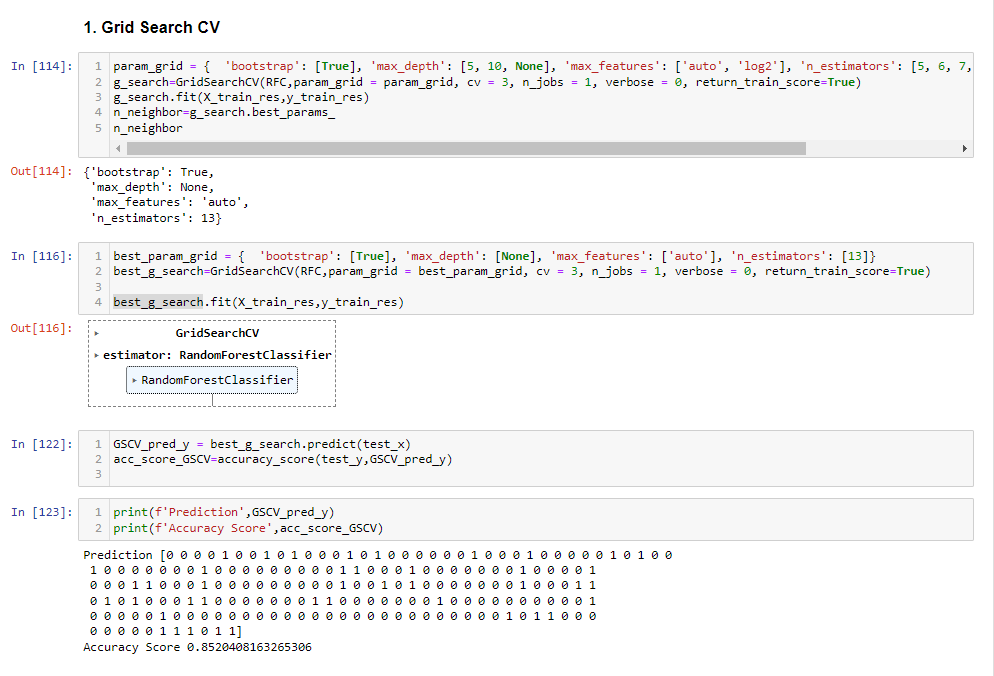


Below is the ROC plot of the above model.

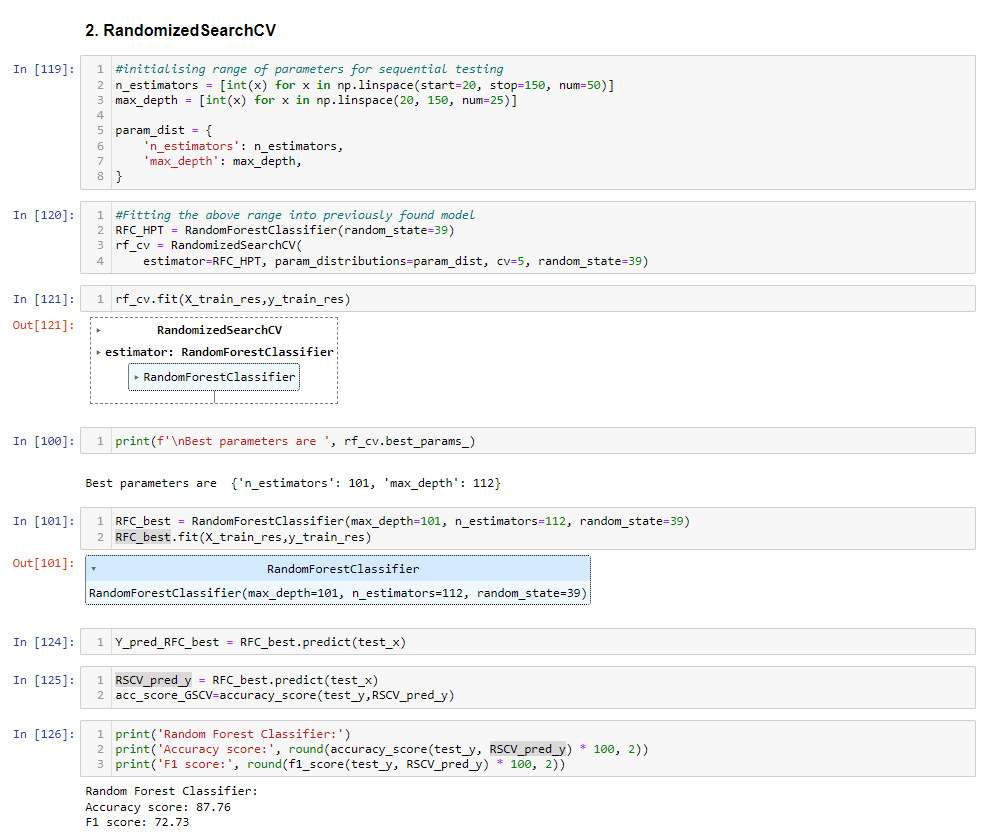


# VI Hyper parameter Tuning:

We used both GridsearchCV and Randomised SearchCV for the hyperparameter tuning, but found RandomisedSearchCV to be more productive. Gridsearch CV’s tuning wasn’t as effective. (Codes attached below)



We found no change with this method, the accuracy score was still 0.8520



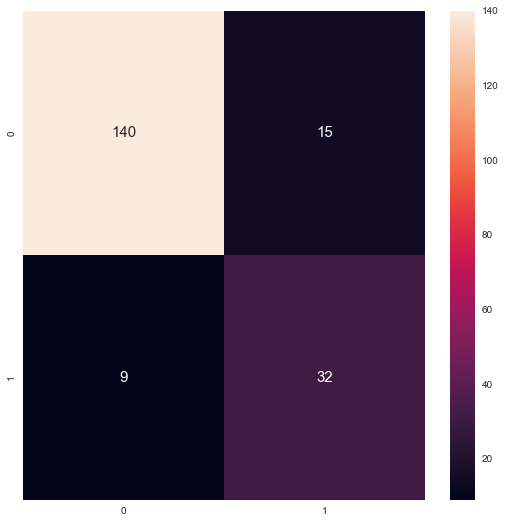
Using Randomised SearchCV found a better accuracy, of about 87.76%, which is about 1.5% more than the accuracy of the Random Forest Classifier without HPT.

Hence, we used this technique, and evaluated it using Area under ROC and classification report. (attached under the next heading)

# VII Metrics:

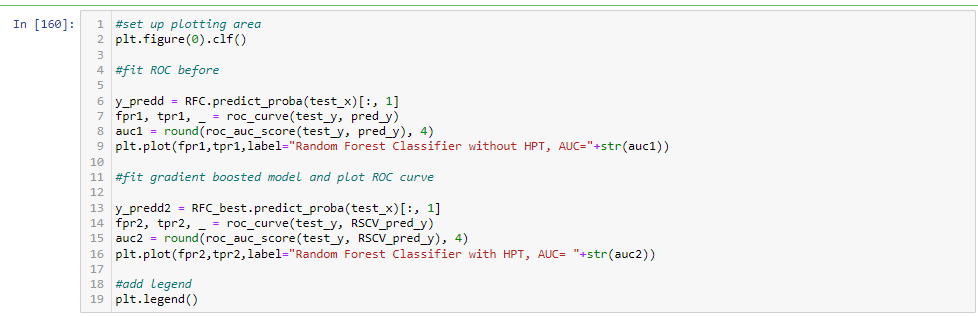
It is clear that we have fine tuned the Random Forest Classifier model to compensate for overfitting and yet give a better accuracy rate. Let us now look at other metrics.

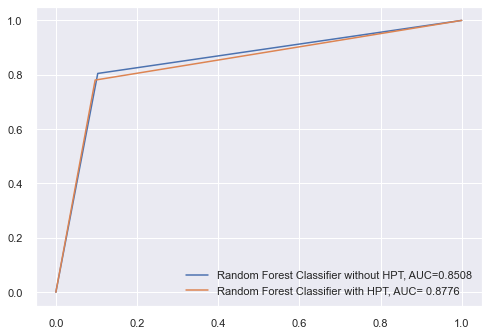
Confusion Matrix:



We find that, we have a very good true positive number, we also have a fairly small false negatives which really complements this model.

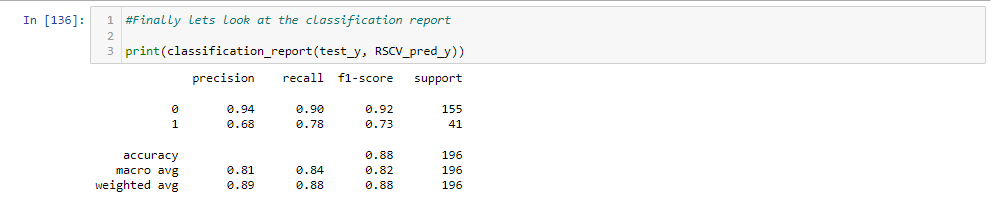
Let us look at another metric, of Area under ROC.





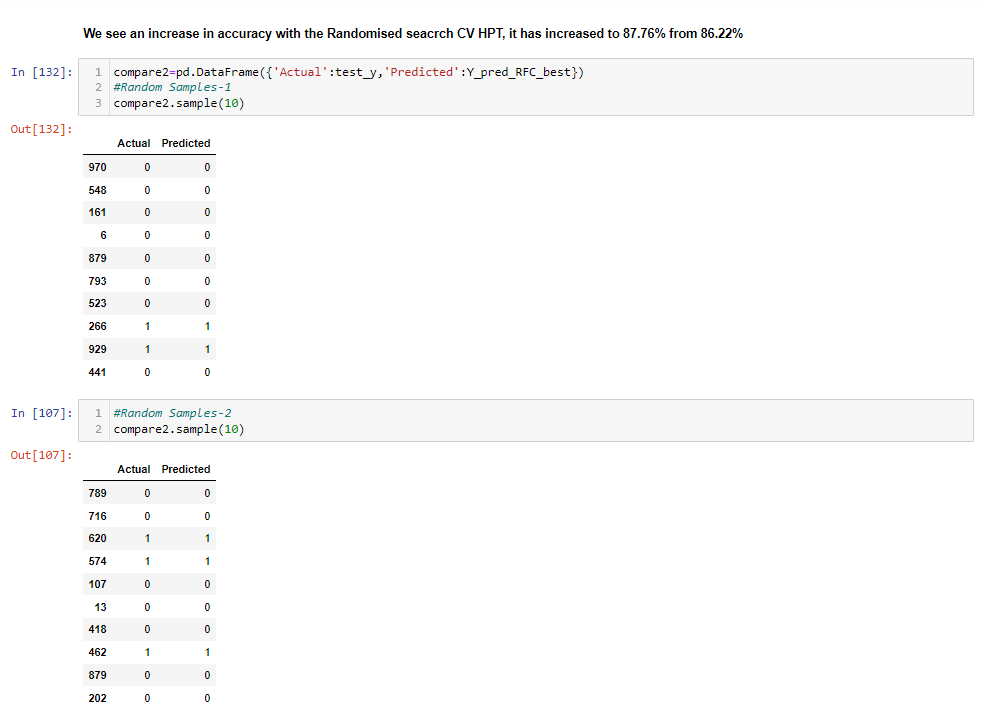
Here we see the Area under the curve as 0.8776 which is another metric to say we have the same 87.76% accuracy.

Finally let us see what the Classification Report says:



A Comparison of the the Final Predicted values after Hyper Parameter Tuning and the actual test values were compared.

Two iterations were sampled, just for our understanding of our model and its predictions.



# VIII Summary:

* We have finally achieved success in the prediction of an insurance fraud, where we have an accuracy of about 84%
* The classification report is shown above for a more detail analysis.
* From the ROC curve we have more or less achieved a right angle L, which further strengthens our finding.
* Between the Y/N in the target column, the model has better chances of predicting a "N", than calling it a fraud.
* Basically this model can basically say that 94% of the time its prediction of a "N" being a negative for fraudulent claim. Which is a good thing.

End of Project

# IX Acknowledgements

My sincere and heartfelt gratitude to the Institute-Datatrained, for giving me this learning experience and guidance thus far. My gratitude to the many authors and their content on the web, which catapult my understanding and enhanced my doubt and error resolution capacity.

Thank you.