## *Topic: A complete analysis report on “Insurance Claim Fraud Detection Project”*



**Introduction:**

Insurance fraud is a well-known problem that insurer companies are facing in the insurance industry. My today’s focus will be on Automobile Insurance fraud, as part of the project.

Automobile insurance fraud is a criminal act that involves deceiving an insurance company in order to receive financial gain. This can be committed by policyholders, third parties, or even the insurance companies themselves. There are two main types of fraud: hard fraud and soft fraud. Hard fraud is when there is a deliberate fabrication of an event such as a theft, accident, or injury to claim money illegally. Soft fraud, which is more common, involves exaggerating a legitimate claim or providing false information to receive a larger payout or lower premiums. For instance, a policyholder might inflate the extent of damage after an accident, or a mechanic could bill for repairs that were never performed. The consequences of committing auto insurance fraud are severe, ranging from fines and loss of insurance coverage to imprisonment. It's a serious issue that affects not only the insurance industry but also raises premiums for honest policyholders, making it a concern for the general public as well.

Automobile insurance fraud presents significant challenges to the industry, with a wide range of deceptive practices that lead to substantial financial losses and increased premiums for honest policyholders. The types of fraud can vary from exaggerated claims, where the cost of actual damages is inflated, to more complex schemes like staged accidents or false theft reports. These fraudulent activities not only impose a financial burden on insurance companies but also have broader societal impacts, such as endangering innocent people and contributing to higher overall insurance costs for consumers. To combat these challenges, it is crucial for both insurers and policyholders to be vigilant and informed about the various forms of fraud and the legal repercussions that can follow. Strategies to mitigate these risks include technological advancements in fraud detection, stricter regulations, and public awareness campaigns to educate consumers about the signs of insurance fraud and the importance of reporting suspicious activities.

Data Science and Machine Learning are revolutionizing the way automobile insurance fraud is detected and prevented. By leveraging large datasets and advanced algorithms, these technologies can identify patterns and anomalies that may indicate fraudulent activity. For instance, machine learning models can analyze claims data, policyholder details, and historical records to flag suspicious cases for further investigation. Techniques like multimodal learning, which combines different types of data such as text, images, and structured data, have shown significant improvements in detecting fraudulent behavior. Furthermore, predictive models using algorithms like AdaBoost and XGBoost have been successful in classifying claims with high accuracy, helping insurance companies to minimize losses and maintain integrity in the claims process. As these technologies continue to evolve, they offer a powerful tool for combating fraud and ensuring a fair and trustworthy insurance system.

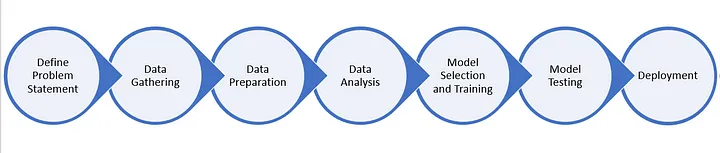
In this blog, I have created Machine Learning model to detect if claim is fraudulent or not. I have used various features & information provided in the datasets like insured personal details, incident information etc. In total, the dataset includes 40 features. Therefore, using all this available information, analysis has been done. A good model score of 92% accuracy has been achieved. I have also used various visualization techniques to understand importance of features & co-linearity.

So, let’s see what are the challenges that I faced during the making of M/L model:

**Problem Definition:**

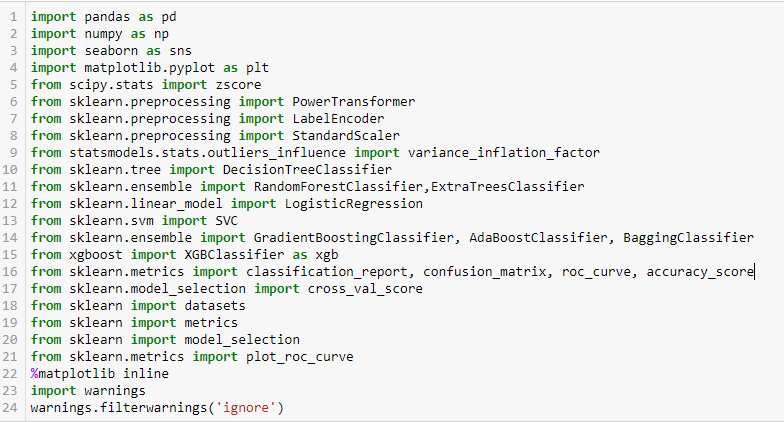
Insurance fraud is a significant issue in the industry. It is tough to recognize fraud allegations. Machine Learning is in a unique position to assist the auto insurance business with this issue.In this project, we are given a dataset that contains information on the insurance policy as well as customer information. It also includes information about the accident that served as the basis for the claims.   
In this example, you will use auto insurance data to show how to build a predictive model that can determine if an insurance claim is fraudulent or not. In this task, we will investigate the insured person's details and incidents, as well as study the sample, to determine whether or not the claim is legitimate.

**Let us go step by step through the data analysis procedure:**  
To develop a Machine Learning Model, we have a Machine Learning Life Cycle that every Machine Learning Project must follow throughout the model's lifetime. Let's take a look at the model life cycle first, and then we'll dig deeper into the actual machine learning model to better comprehend it alongside the lifecycle.



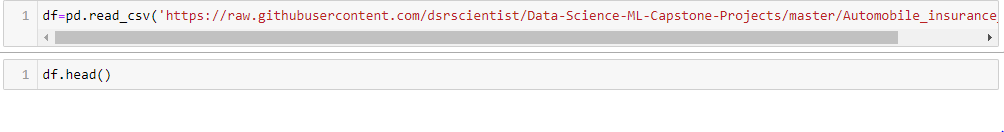
**Import necessary dependencies/libraries:**

To analyze or import the dataset, we imported all essential libraries, as shown below:  
1. Pandas was used to import the dataset and create data frames.   
2. Seaborn and Matplotlib were used for visualization.   
3. Date Time was utilized to extract the day, month, and date independently.   
4. The model was built using Sklearn.

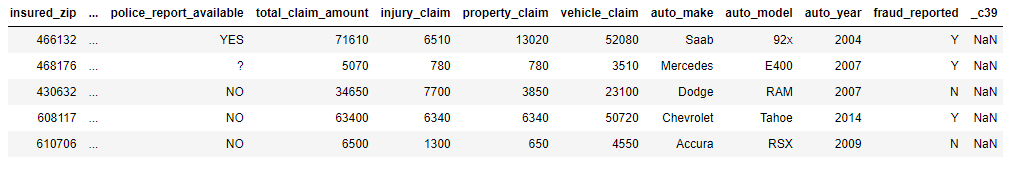


**Import the Dataset**

Now, importing the dataset.

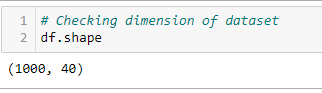


I have imported dataset that was in “csv” format as “df”. Check dataset below:



By inspecting the dataset, we can see that it comprises both categorical and numerical columns. "fraud\_reported" is our goal column, and because it includes two categories, it is referred to as a "Classification Problem" in which we must forecast whether an insurance claim is fraudulent or not. Because it is a classification problem, we will employ all classification techniques when developing the model, which we will see as the blog progresses.

We also determined the number of rows and columns using the simple code 'df.shape'. We got 1000 rows and 40 columns. PCA can be performed, but I decided not to lose any data at this moment because the dataset is very small, and the first lesson of a data scientist is 'Data is Crucial', so I proceeded with all of the data.



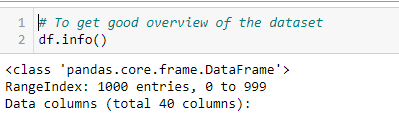
According to the machine learning model's lifecycle, we have already accomplished steps 1 and 2. Now, let us go on to points 3, 4, 5, and 6, which are the most important aspects of any machine learning model. The better the data is evaluated and cleaned, the higher the model accuracy will be, or the model will stay overfitting or underfitting. We'll go over why all of the steps are used.

**Exploratory Data Analysis & Data Preparation:**

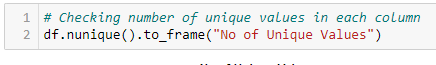
In this part we will firstly be exploring the data with some basis steps and then further proceed with some crucial analysis, like feature extraction, imputing and encoding.

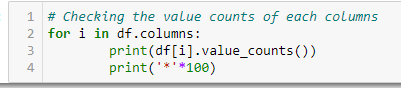
Let’s start with checking shape, unique values, value counts, info etc…..

After doing the analysis if we find any unnecessary columns in the dataset we can drop those columns.

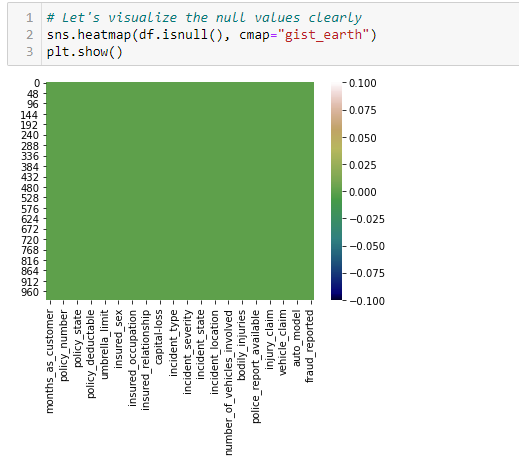








After doing this basis analysis, now we are checking for the null values and further will mention all the observations.



**Observations :**

* First we can see that we donot have any null values in the dataset.
* Second, the dataset contains 3 different types of data namely integer data type, float data type and object data type.
* Third, after analyzing it is seen that c39 column has only entries those are all NaN. Keeping all entries NaN is useless hence dropping that column



* Fourth, we can observe the columns policy\_number and incident\_location have 1000 unique values which means they have only one value count. So it not required for the prediction so we can drop it.



* Fifth, by looking at the value counts of each column we can realize that the columns umbrella\_limit, capital-gains and capital-loss contains more zero values around 79.8%, 50.8% and 47.5%. I am keeping the zero values in capital\_gains and capital\_loss columns as it is. Since the umbrella\_limit columns has more that 70% of zero values, let's drop that column.

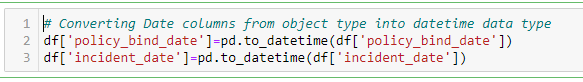


* Sixth, the column insured\_zip is the zip code given to each person. If we take a look at the value count and unique values of the column insured\_zip, it contains 995 unique values that mean the 5 entries are repeating. Since it is giving some information about the person, either we can drop this or we can convert its data type from integer to object for better processing.

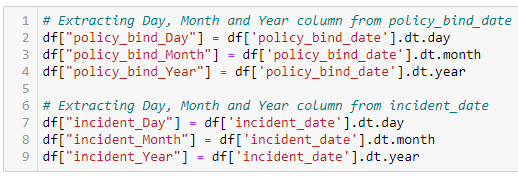


**Proceeding to Feature Extraction:**

The policy\_bind\_date and incident\_date have object data type which should be in datetime data type that means the python is not able to understand the type of this column and giving default data type. We will convert this object data type to datetime data type and we will extract the values from these columns.



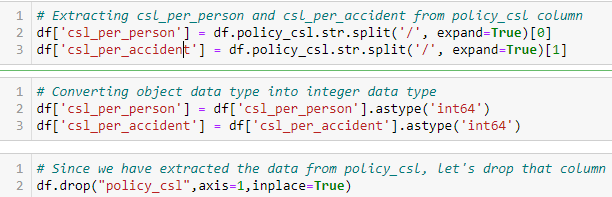
Now that we have converted object data type into datetime data type. Now let's extract Day, Month and Year from both the columns



After we have extracted Day, Month and Year columns, from both policy\_bind\_date and incident\_date columns. So we can drop these columns.



Again, from the features we can see that the policy\_csl column is showing as object data type but it contains numerical data, maybe it is because of the presence of "/" in that column. So first we will extract two columns csl\_per\_person and csl\_per\_accident from policy\_csl colums and then will convert their object data type into integer data type.



After extracting we have dropped the policy\_csl feature.

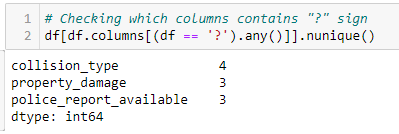
Also we have observed that the feature ‘incident-year’ has one unique value throughout the column also it is not important for our prediction so we can drop this column.



**Moving on to Imputation:**

Imputation is a technique to fill null values in the dataset using mean, median or mode. YES…. I know you might be thinking that we did not get any null values while checking for the null values, however from the value counts of the columns we have observed that some columns have "?" values, they are not NAN values but we need to fill them.

So, let’s begin……



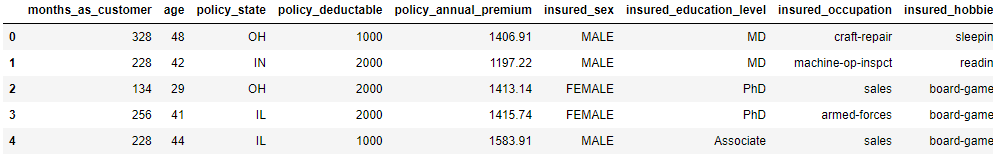
These are the columns which contains "?" sign. Since these column seems to be categorical so we will replace "?" values with most frequently occurring values of the respective columns that is their mode values.



The mode of property\_damage and police\_report\_available is "?", which means the data is almost covered by "?" sign. So we will fill them by the second highest count of the respective column.

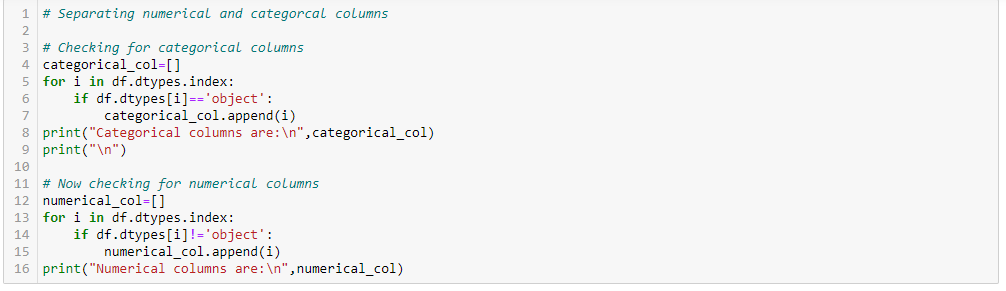


Now after all the data cleaning until now, the dataset looks like this…..



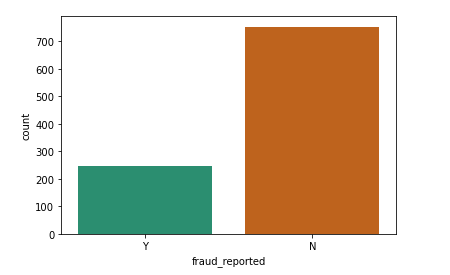
**Preparing for Visualization**

First we will look into the categorical and numerical columns so that we can visualize the features accordingly.

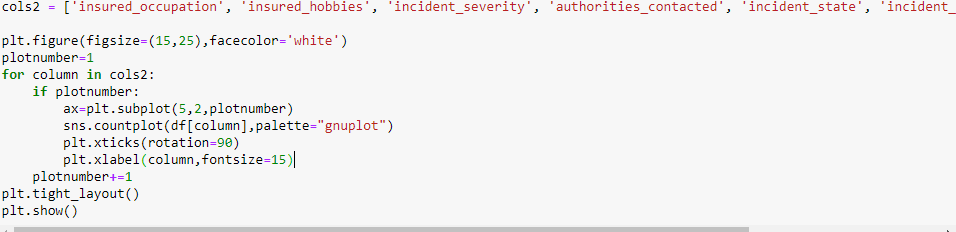


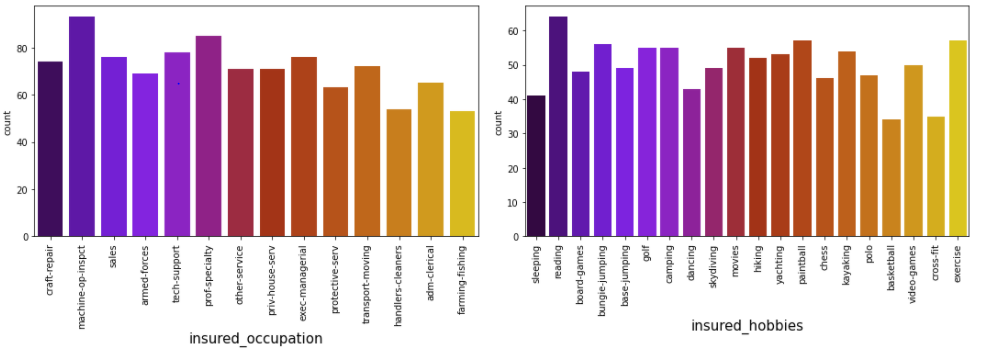
**Visualization**

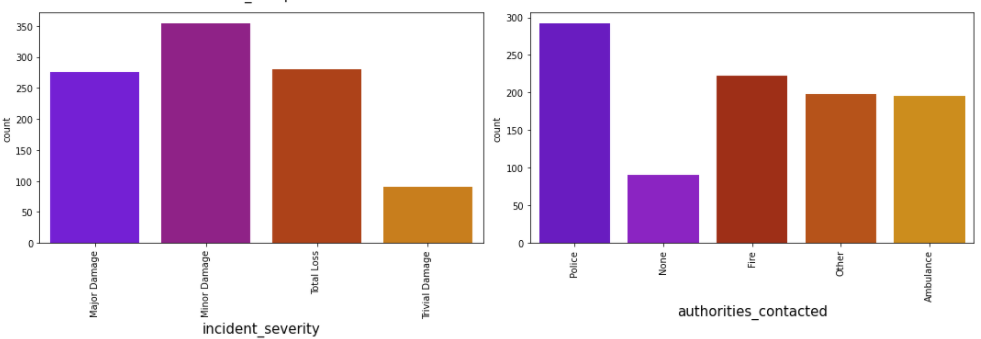


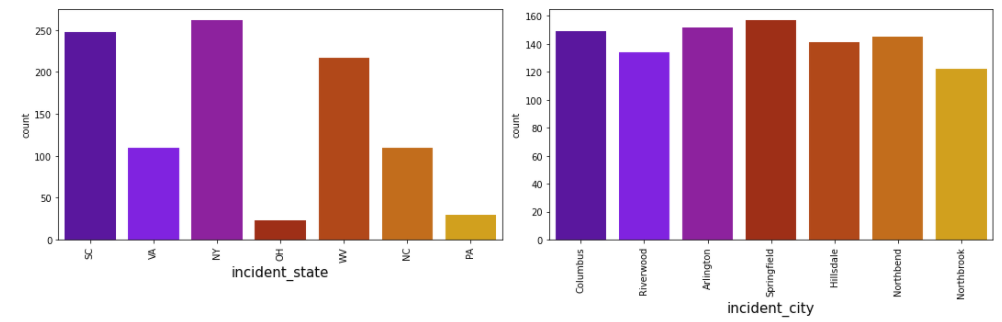


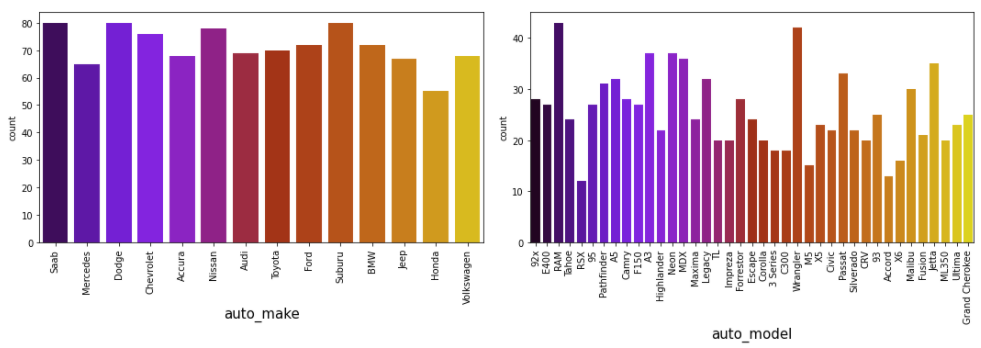
By looking into the plot we can observe that the count of "N" is high compared to "N". Which means here we can assume that "Y" stands for "Yes" that is the insurance is fraudulent and "N" stands for "No" means the insurance claim is not fraudulent. Here most of the insurance claims have not reported as fraudulent. Since it is our target column, it indicates the class imbalance issue. We will balance the data using oversampling method in later part.





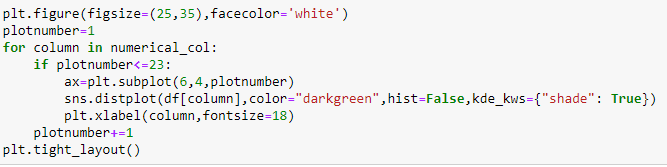


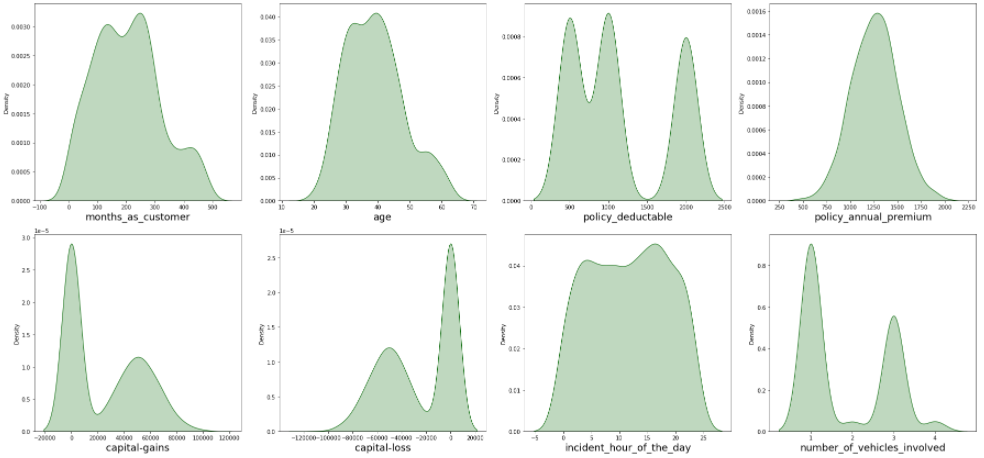


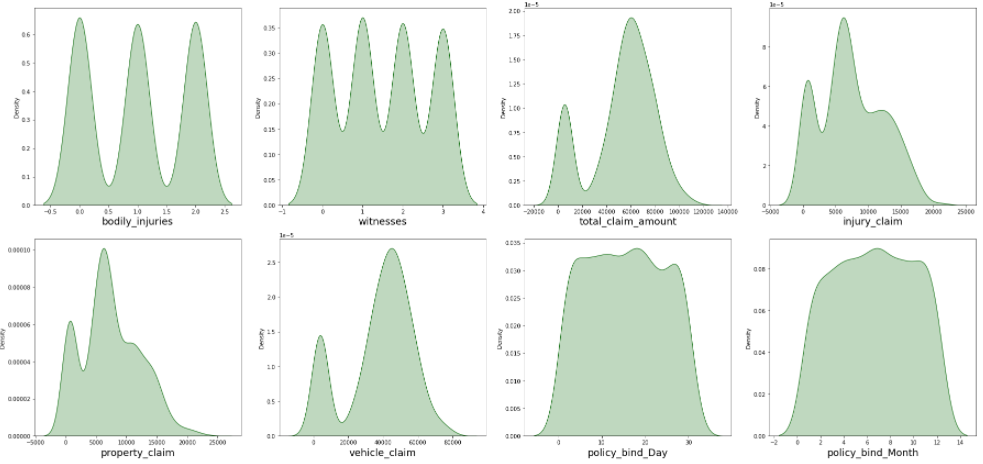


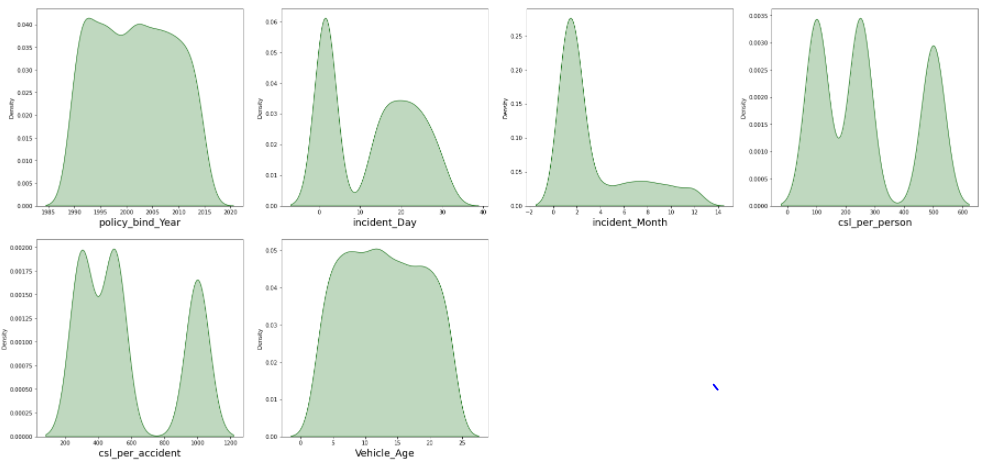
 By looking into the count plots we can observe the following things:

* In the insured occupation we can observe most of the data is covered by machine operation inspector followed by professional speciality.
* With respect to insured hobbies, we can notice reading covered the highest data followed by exercise. And other categories have the average counts.
* The incident severity count is high for Minor damages and trivial damage data has very less count compared to others.
* When the accidents occurs then most of the authorities contacts the police, here the category police covers highest data and Fire having the second highest count. But Ambulance and Others have almost same counts and the count is very less for none compared to all.
* With respect to the incident state, New York, South Carolina and West Virginia states have highest counts. In incident city, almost all the columns have equal counts.
* When we look at the vehicle manufactured companies, the categories Saab, Suburu, Dodge, Nissan and Volkswagen have highest counts.
* When we take a look at the vehicle models then RAM and Wrangler automobile models have highest counts and also RSX and Accord have very less count.

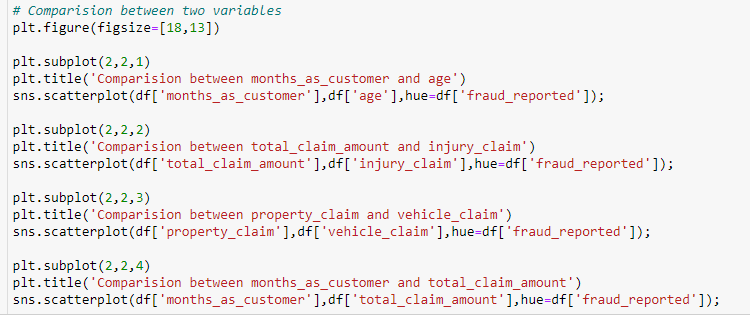


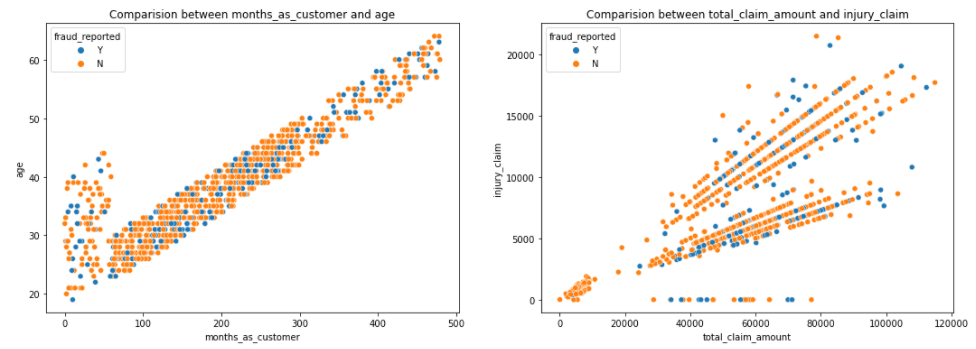


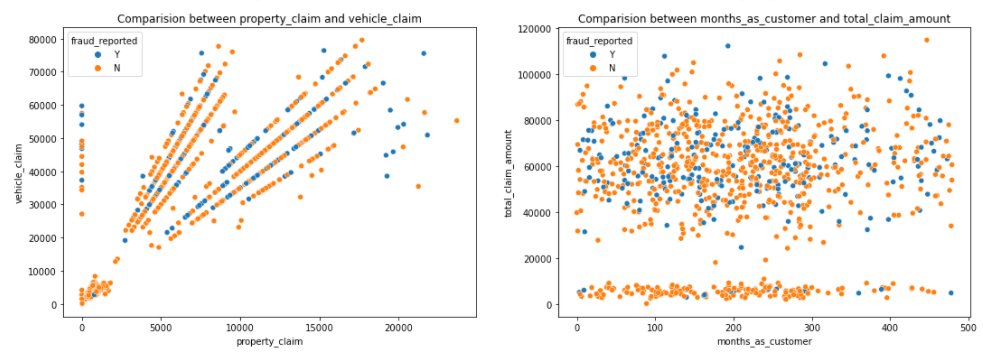




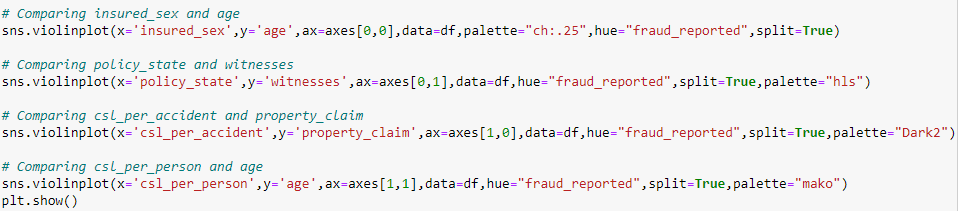
The data is normally distributed in most of the columns. Some of the columns like capital gains and incident months have mean value greater than the median, hence they are skewed to right. The data in the column capital loss is skewed to left since the median is greater than the mean. We will remove the skewness using appropriate methods in the later part.

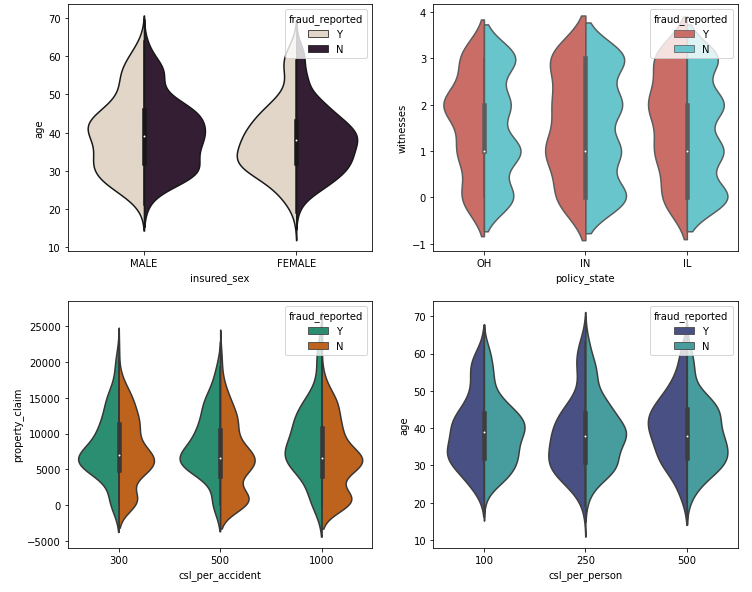


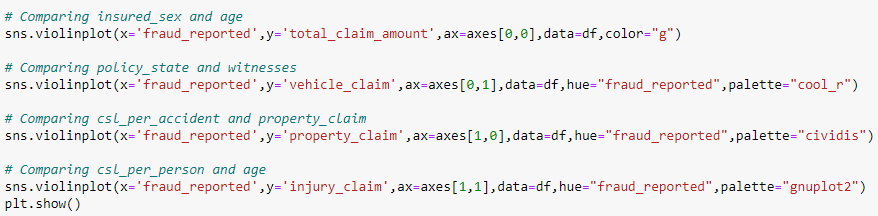


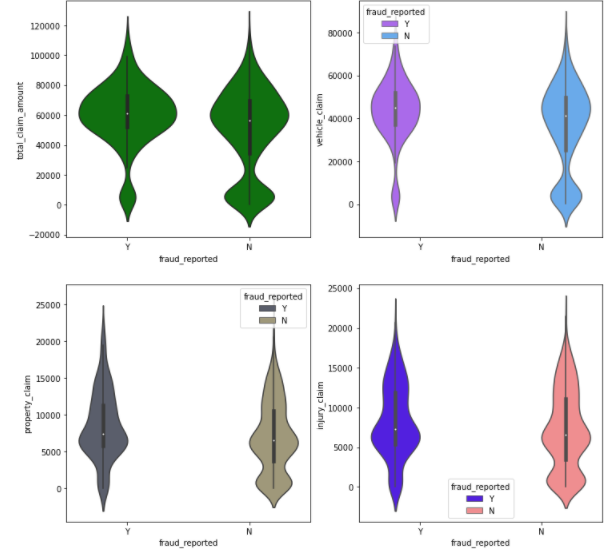


* There is a positive linear relation between age and month\_as\_customer column. As age increases the month\_as customers also increases, also the fraud reported is very less in htis case.
* In the second graph we cna observe the positive linear relation, as total cliam amount increases, injury claim is also increases.
* Third plot is also same as second one that is as the property claim increases, vehicle claim is also increases.
* In the fourth plot we can observe the data is scattered and there is no much relation between the features.



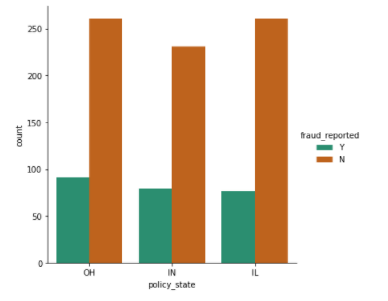




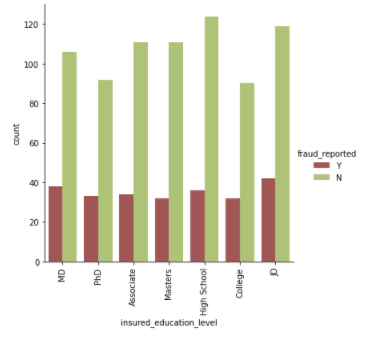


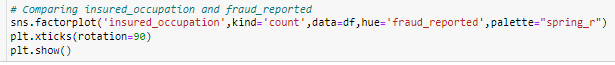
Visualization is a technique where by comparison and plotting the data becomes self explanatory, which we have seen until now. Moving ahead with some more visualization plots before we can proceed to model building.

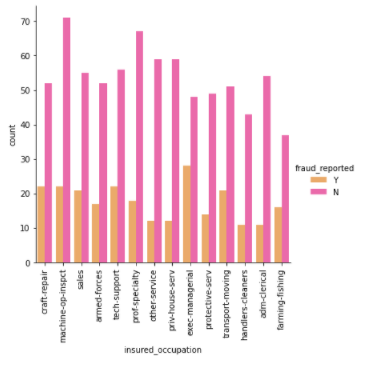




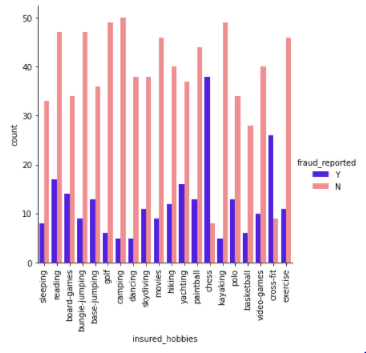


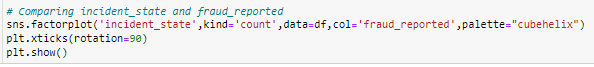


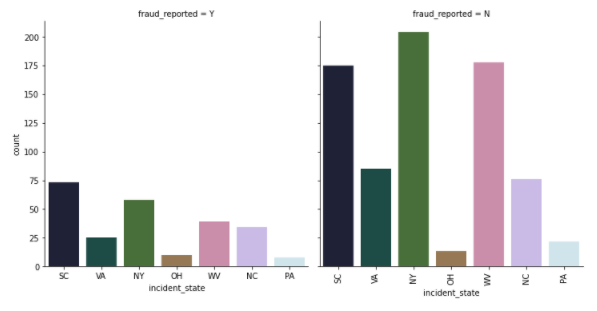




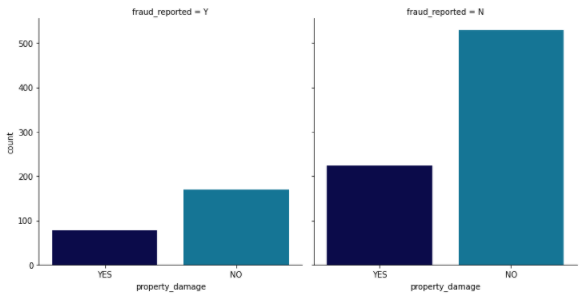






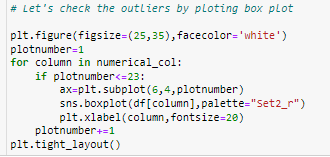


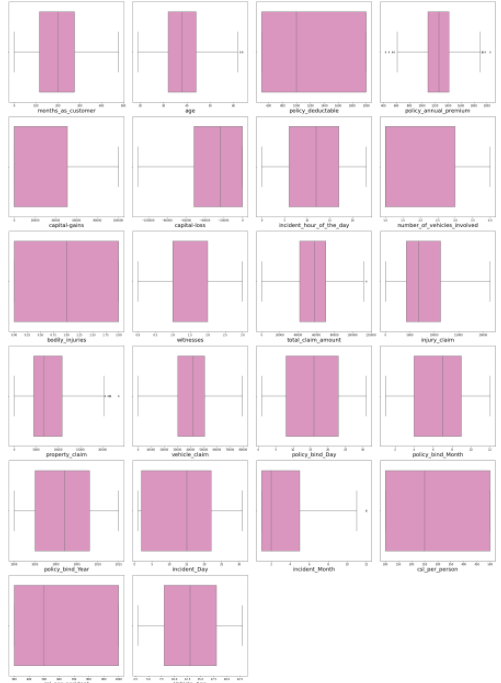




Now we have done with the visualization in order to analyze and understand the data. So in this EDA part, we have looked into various aspect of the dataset, like looking for the null values and imputing, extracting date time, observing the value counts and doing the feature extraction etc. Now we will be performing another analysis by identifying the outliers and removing them. Along with it we will also look for the skewness of the dataset and remove the skewness.

**Identifying the Outliers and Skewness**

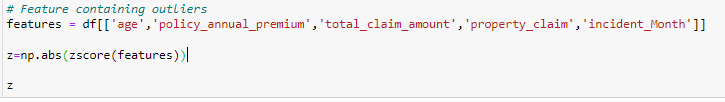




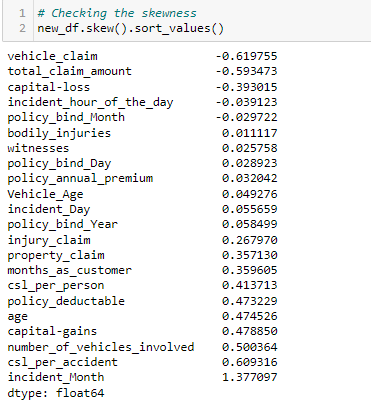
As we can see, I have used box plot to identify the outliers and we can find the outliers in the following columns:

* Age
* policy\_annual\_premium
* total\_claim\_amount
* property\_claim
* incident\_month

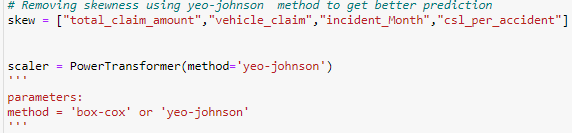
These are the numerical columns which contains outliers hence removing the outliers in these columns using Zscore method.



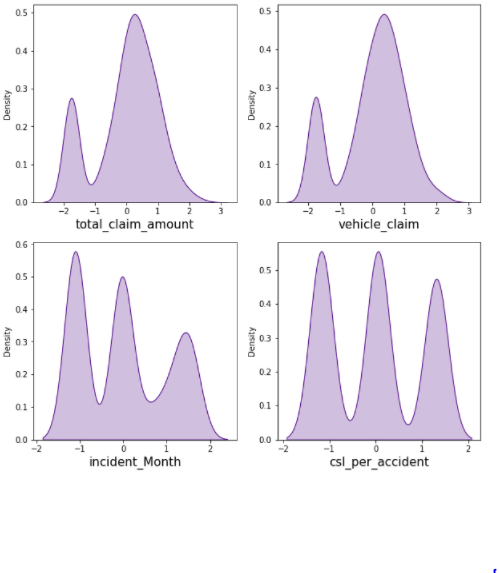
Now that we have removed the outliers, I will proceed to look into the skewness of the data and then remove it.



As we can see that skewness is present in the dataset, hence I am using the yeo-johnson method to remove the skewness.



Now we have removed the skewness and the data looks normally distributed.

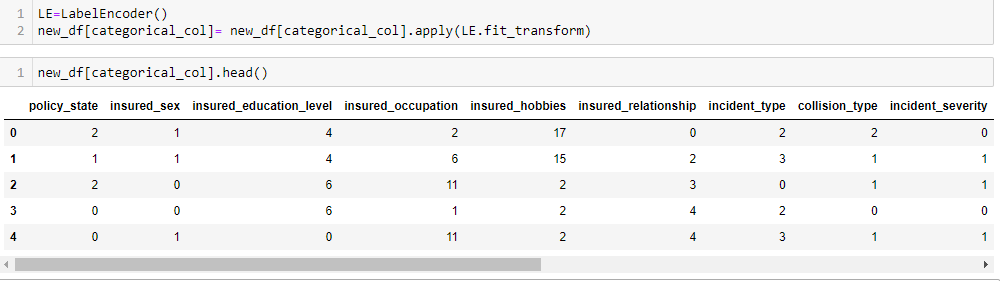


Now we have completed our analysis of the dataset and also cleaned the dataset so that we can build a good model….

However we are not yet ready. We have seen above that there are some problems that still exist in the dataset. We have seen that the dataset has both numerical and categorical data. The model only understand numerical data, hence we will encode the data. Also we have seen that there can be some multi-colinearity, that we will see through a heatmap and also further remove it. Again we have also seen that the target variable is imbalance, hence will fix it by oversampling. And finally we will scale the data so that it is ready to be trained and tested.

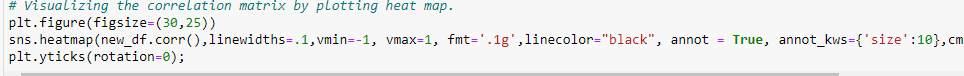
Let’s begin.

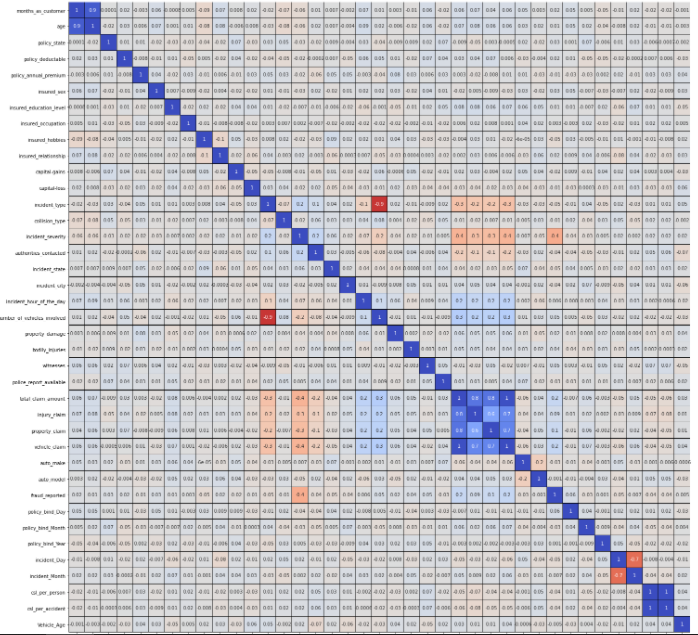
**Encoding the data**



Now we have encoded the dataset using label encoder and the dataset looks like this.

Moving forward, to check the co relation between the feature and target and also the relation between the features using the heatmap.





This heatmap shows the correlation matrix by visualizing the data. We can observe the relation between one feature to another.

There is very less correlation between the target and the label. We can observe the most of the columns are highly correlated with each other which lead to the multicollinearity problem. We will check the VIF value to overcome with this multicollinearity problem.

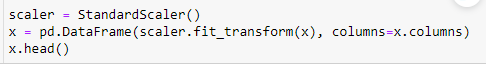
**Preprocessing Pipelines**

## Separating the features and label variables into x and y



**Scaling the DataSet**

## Feature Scaling using Standard Scalarization



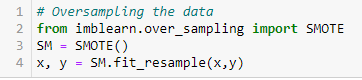
**Checking Multi-colinearity using VIF**



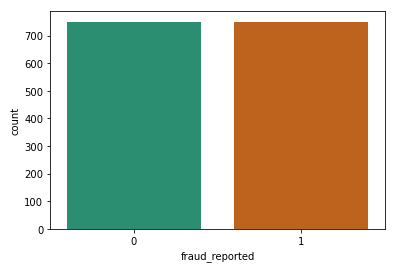
It is observed that some columns have VIF above 10 that mean they are causing multicollinearity problem. Let's drop the feature having high VIF value amongst all the columns.

I have dropped the total\_claim\_amount and csl\_per\_accident features with colinearity more than 10, and now we have removed the problem.

We had earlier identified another problem of imbalance data in the target variable, lets treat it.



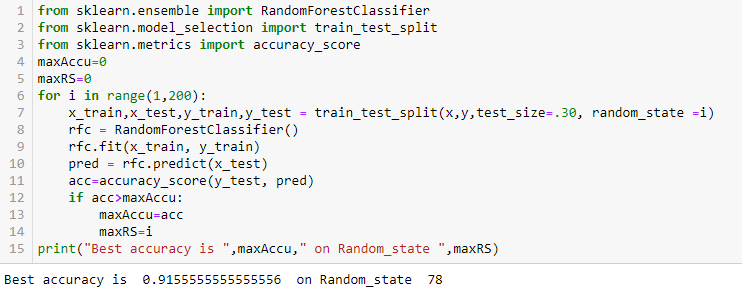
As we have treated the oversampling issue using SMOTE, now the data looks good.



Finally we have got into the position where we will start building the model.

At first let’s find the best random state in which we can build the model*.*

*(Random state ensures that the splits that you generate are reproducible. Scikit-learn use random permutations to generate the splits. The random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order.)*

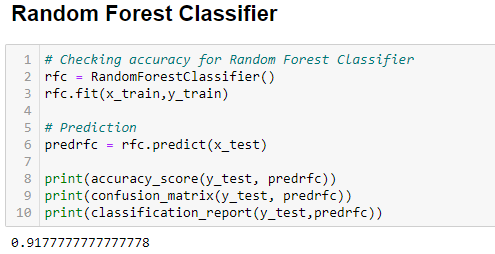


Here we have used the RandomForestClassifier to find the best random state, as we have got an accuracy score of 91% (pretty good), at the random state of 78. Let’s use this random state to build our models.

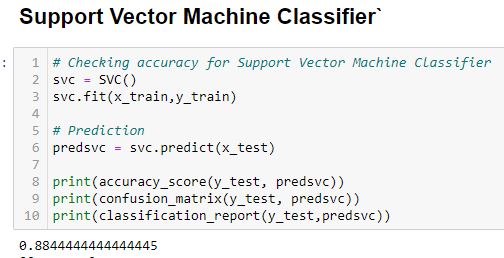
Before doing that, let us split the dataset into train and test using train\_test\_split.



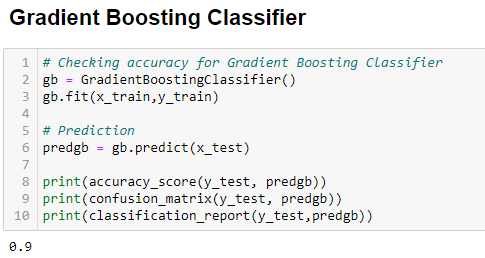
**Model Building**



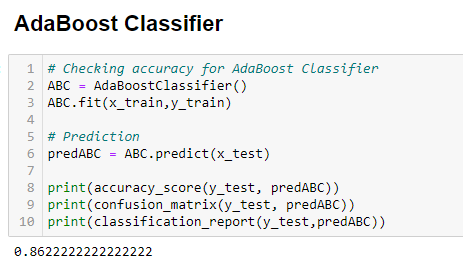
The first model was built using RandomForestClassifier, which gave an accuracy score of 91%, however we are hungry data scientist and will not be satisfied with only one model. We will try various models and see what accuracy score we get.



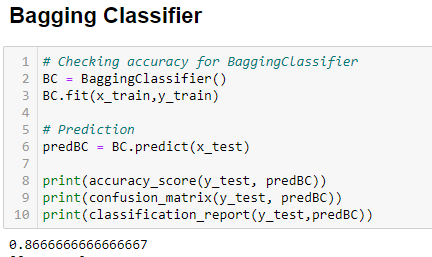
With SupportVectorClassifier we got an accuracy score of 88%.



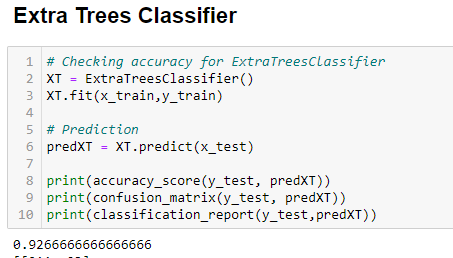
With GradientBostingClassifier we got an accuracy score of 90%.



With AdaBoostClassifier we got an accuracy score of 86%.

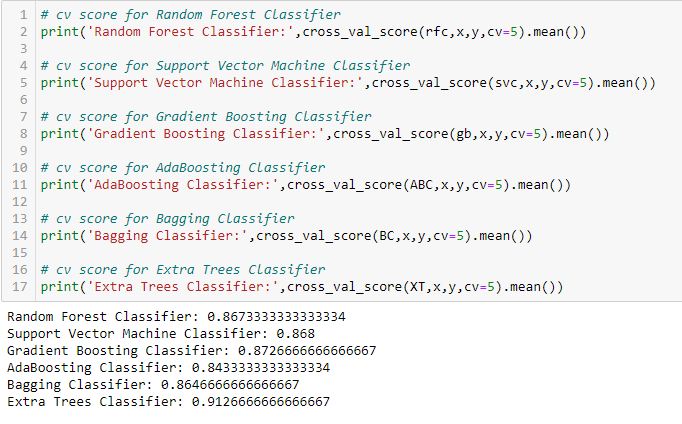


With BaggingClassifier we got an accuracy score of 88%.



With this model, ExtraTreesClassifier we have got accuracy score of 92%, which is better than RandomForestClassifier.

Before we can announce the best model, we always have to make sure that the model is not over fitting; hence we will perform cross validation of all the models built.



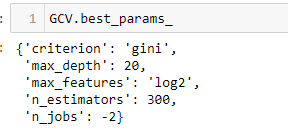
After the cross validation we can clearly see that ExtraTreesClassification is the best fit model.

Now, that we have found the best fit model, lets perform some HyperParameterTuning to improve the performance of the model.









Here we have got the best parameters, and we will build our final model using these parameters.

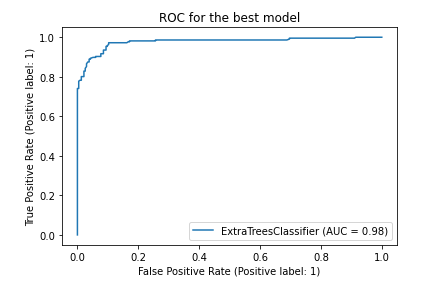


We have built our final model and we can see that the accuracy scores has increased by 1% from the cross validation score.



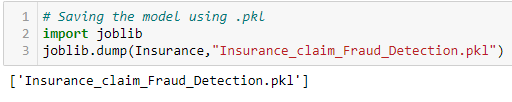
This is the confusion matrix for the model.

Plotting and AUCROC curve for the final model.

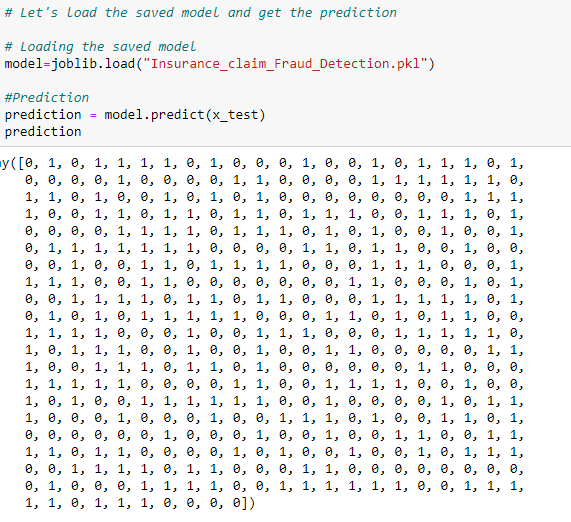


So here we can see that the area under curve is quite good for this model.

**Save the model**



**Predicting the model**



**Concluding Remarks**



In the beginning of this blog, we described the lifespan of a Machine Learning Model; you can see how we touched on each aspect before moving on to model building and deployment.  
This sector requires a strong understanding of data, and data analysis and feature engineering are critical components of any model building problem.  
  
You can see how we handled numerical and categorical data, as well as how we built various machine learning models using the same dataset.  
  
Using hyper parameter adjustment, we can increase our model accuracy; for example, in this model, the accuracy stayed constant.

Using our machine learning model, we can quickly predict if an insurance claim is fake or not, and we can reject applications that are considered fraudulent claims.

**THANK YOU!**