## Insurance Claim Fraud Detection Project

**In association with Data Trained Academy Batch: 1842**

**PREPARED BY :- POONAM YADAV**

Example of an end-to-end machine learning project in Data Science for beginners.

I am going to write about a complete end-to-end project for Insurance Claim Fraud Detection which should serve as a guiding path for many Data Science aspirants.

I have written down all the techniques in the form of sub-topics that I will be explaining one by one. And those sub-topics are as follows:  
  
1.      Problem Definition.  
2.      Data Analysis.  
3.      EDA Concluding Remark.  
4.      Pre-Processing Pipeline.  
5.      Building Machine Learning Models.  
6.      Concluding Remarks.

Let’s start with the problem definition or a short introduction on the Insurance Claim Fraud Detection that I have chosen to elaborate.

**Introduction:**

Insurance fraud is a deliberate deception perpetrated against or by an insurance company or agent for the purpose of financial gain. Fraud may be committed at different points by applicants, policyholders, third-party claimants, or professionals who provide services to claimants. Insurance agents and company employees may also commit insurance fraud. Common frauds include “padding,” or inflating claims; misrepresenting facts on an insurance application; submitting claims for injuries or damage that never occurred; and staging accidents.

Thanks to Data Science and Machine Learning, which has been very useful in many industries that have managed to bring accuracy or detect negative incidents. Here in this blog, I have created a Machine Learning model to detect if the claim is fraudulent or not. Here various features have been used like insured information, insured persons, personal details and the incident information. In total the dataset has 40 features and 1000 entries rows of data. Using all these previously acquired information and analysis done with the data I have achieved a good model that has 95% accuracy. Let’s see what are the steps involved to attain this accuracy.

Various visualization techniques have also been used to understand the Correlation, Multicollinearity and importance of the features with respect to the algorithms.

Note: Various special words are used in the article assuming the fact that the reader is aware of the language used in data science.

**Hardware & Software Requirements & Tools Used:**

### Hardware used:

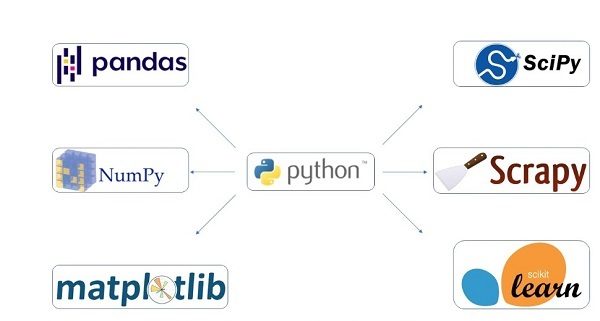
* Processor: Core i5 -10300H CPU @ 2.50GHz
* RAM: 8 GB
* Operating System: 64-bit
* ROM/SSD: 1 TB SSD
* Graphics: NVIDIA GeForce GTX 1650 Ti

**Software requirement**:

* Anaconda Navigator - Jupyter Notebook

**Libraries Used**:

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scipy
* Date Time
* Scikit Learn



Heading forward we will try to understand the problem statement and the dataset.

1. **Problem Definition**

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.

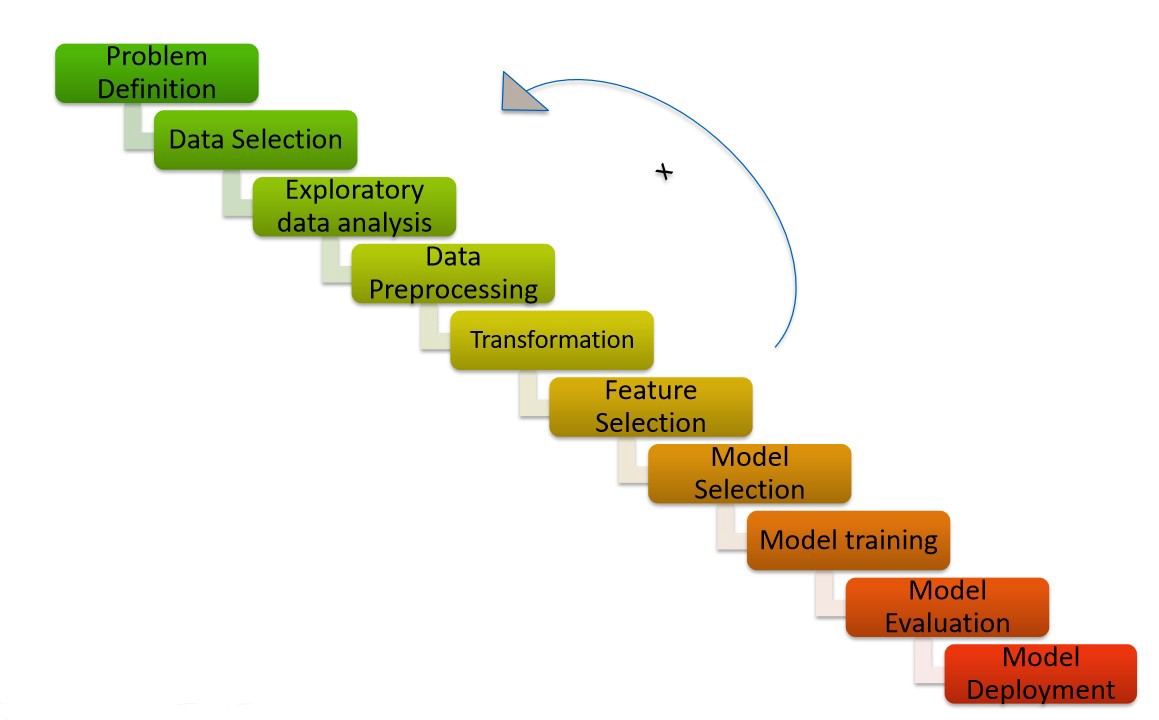
In this project, we have a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.

In this we will be working with some auto insurance data to demonstrate how we can create a predictive model that predicts if an insurance claim is fraudulent or not.

Let’s proceed step by step in the data analysis process.

1. **Data Analysis**

In order to build a Machine Learning Model, we have a Machine Learning Life Cycle that every Machine Learning Project has to touch upon in the life of the model. Let’s take a look at the model life cycle and then we will look into the actual machine learning model and understand it better along with the lifecycle as we move forward.

****

Now that we understand the lifecycle of a Machine Learning Model, lets import the necessary libraries and proceed further.

**Importing the necessary Libraries:**

To analyze the dataset or even to import the dataset, we have imported all the necessary libraries as shows below.

Pandas has been used to import the dataset and also in creating data frames.

Numpy has been used for numerical tasks.

Seaborn and Matplotlib have been used for Data Visualization.

Date Time has been used to extract day/month/date separately.

Scipy has been used in the Zscore method for removing outliers.

Sklearn has been used in the model building.

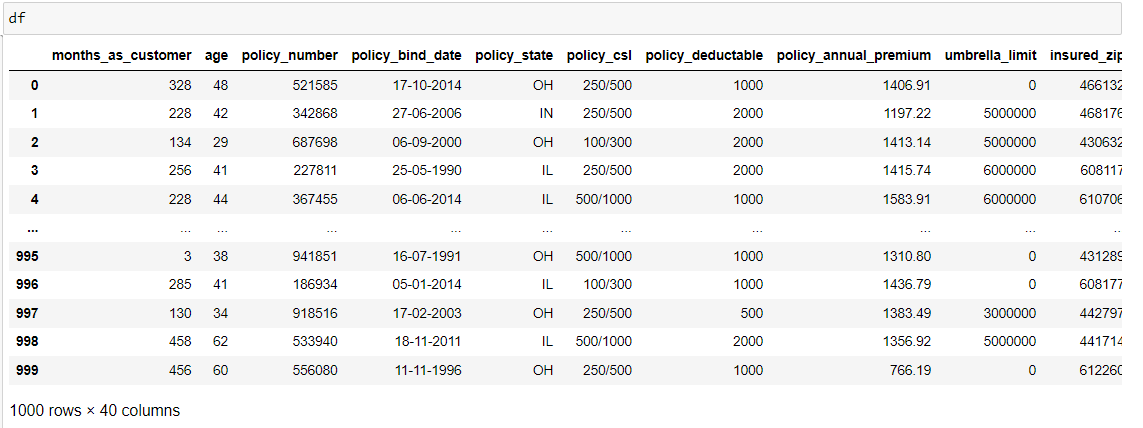
**Importing the Dataset**

Let’s import the dataset first.



Copied the raw data and saved it as a csv file on my local computer after which I imported the entire dataset on this Jupyter Notebook with the help of pandas.

I have imported the dataset which was in “csv” format as “df”. Below is how the dataset looks.



By observing the dataset, we could make out that the dataset contains both categorical and numerical columns. Here "fraud\_reported" is our target column, since it has two categories so it is termed to be a "Classification Problem" where we need predict if an insurance claim is fraudulent or not. As it is a classification problem hence, we will be using all the classification algorithms while building the model that we will see as the data analysis proceeds.

1. **Exploratory Data Analysis (EDA)**

As per the lifecycle of the machine learning model we have already completed points 1 and 2. Now let’s move on to the point 3, 4, 5 and 6 which is the most crucial part of any machine learning model. If we prepare the dataset, analyze it and clean it in the best way possible the better model accuracy we will get, or the model can get over fitted or under fitted. We will discuss further all the steps that are used.

**Data Preparation:**

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

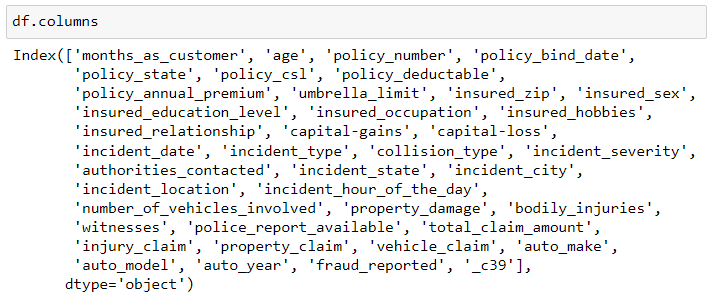
Let’s start with checking shape, unique values, value counts, info etc.

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

By using ‘df.shape’ we also figured out how many rows and columns we have. We have got the result that we have 1000 rows and 40 columns. PCA can be done, however I decided not to lose any data at this time as the dataset is comparatively small and the first lesson of a data scientist is to preserve as much data as possible hence proceeded will all the data.



Out of 40 columns 39 are independent columns and remaining one is our target variable “fraud\_reported".



By using ‘df.columns’ all the columns in the dataset are listed as shown above.

The columns are under different categories as follows:

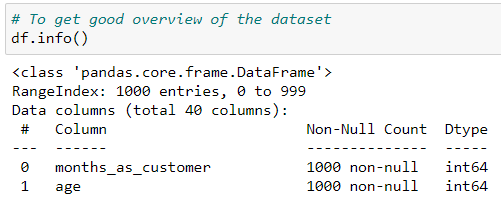
Company's data for insurance claim policy: months\_as\_customer, age, policy\_number, policy\_bind\_date, policy\_csl, policy\_deductable, policy\_annual\_premium, umbrella\_limit

Personal details of the customers: insured\_zip, insured\_sex, insured\_education\_level, insured\_occupation, insured\_hobbies, insured\_relationship, capital- gains, capital-loss.

Details of the incident: incident\_date, incident\_type, collision\_type, incident\_severity, authorities\_contacted, incident\_state, incident\_city, incident\_location, incident\_hour\_of\_the\_day, number\_of\_vehicles\_involved, property\_damage, bodily\_injuries, witnesses, police\_report\_available, total\_amount\_claimed, injury\_claim, property\_claim, vehicle\_claim, auto\_make, auto\_model, auto\_year

Target label:

fraud\_reported: Y-YES / N-NO

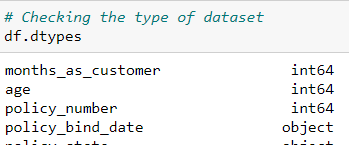


This gives the information about the dataset which includes indexing type, column Name, non-null values, dtypes and memory usage of the dataset.

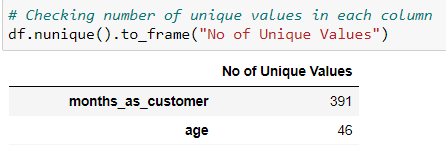
Here the column \_c39 has 0 non null values which means it has NaN throughout the data so we can drop this column.



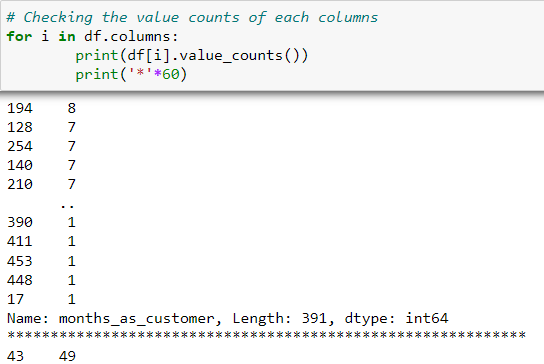
Checking the data types using the “df.dtypes” below.



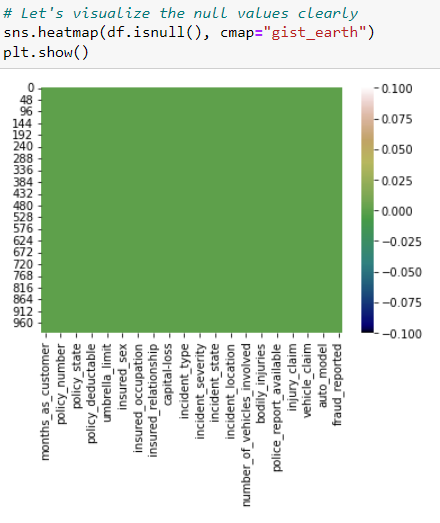
Checking the number of unique values of each column in the data set using the code below



Checking the value counts of each columns using the code below.



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

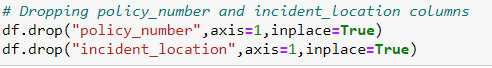


**Observations:**

* We can see that there are null values in the dataset.
* The dataset contains 3 different types of data namely integer data type, float data type and object data type.
* 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.



* We can observe the columns “policy\_number” and “incident\_location” have 1000 unique values. So, it not required for the prediction so we can drop it.



* 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 column has more than 70% of zero values, let's drop that column.

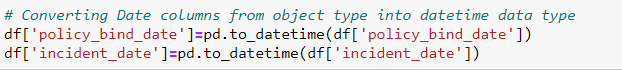


* The column insured\_zip is the zip code given to each person. If we take a look at the value count and unique values of the column insured\_zip, it contains 995 unique values that mean the 5 entries are repeating. Hence, we will drop it.

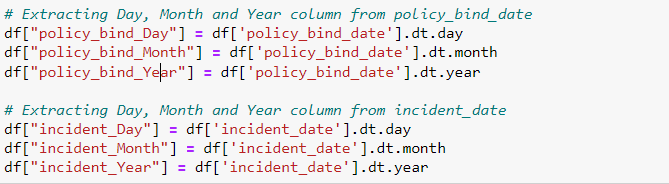


**Feature Extraction:**

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.



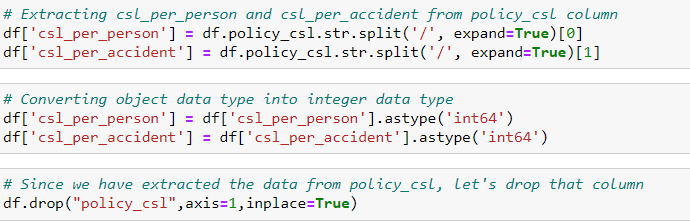
Now that we have converted object data type into datetime data type. Now let's extract Day, Month and Year from both the columns.



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.

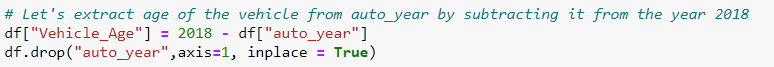


After extracting we have dropped the policy\_csl feature.

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.

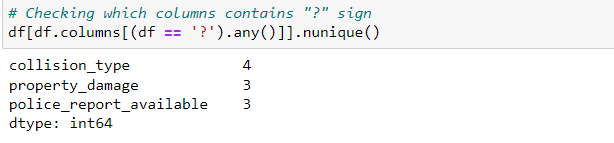


Here we have extracted age of the vehicle on the basis of auto year by assuming the data is collected in the year 2018 as below and dropped “auto year” column after extraction.

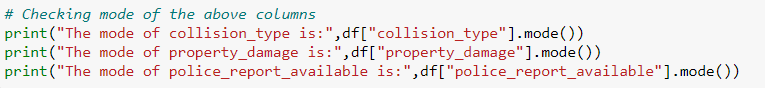


**Imputation:**

Imputation is a technique to fill null values in the dataset using mean, median or mode. 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.

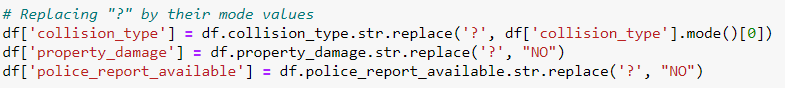


These are the columns which contains "?" sign. Since these columns seems to be categorical so we will replace "?" values with most frequently occurring values of the respective columns that is their mode values.

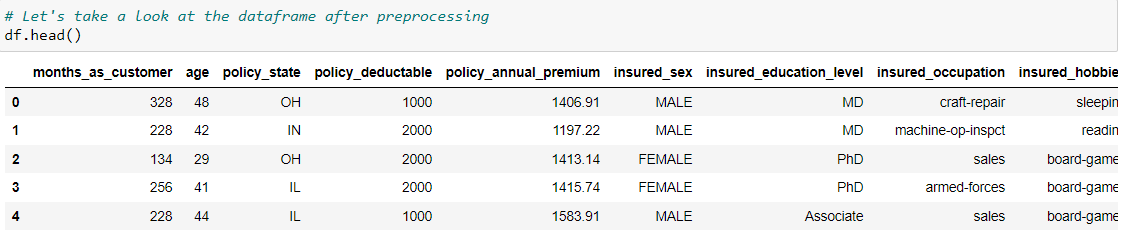


The mode of property damage and police\_report\_available is "?", which means the data is almost covered by "?" sign. So, we will fill them by the second highest count of the respective column.





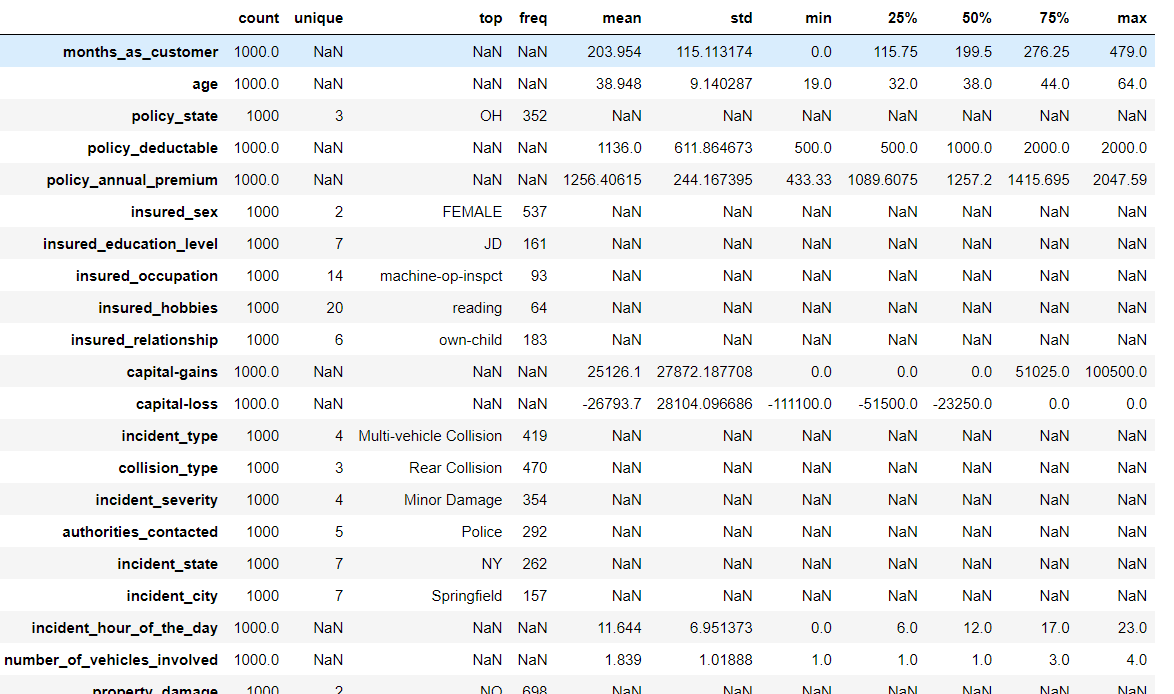
Now after all the data cleaning until now, the dataset looks like below



**Statistical Summary of Dataset**



Here I am using the "describe" method along with its parameter "all" to include each and every column present in our dataset irrespective of them being numeric or text data. I have also used the transpose option to make sure that we are able to see the column information properly without having to scroll through multiple times.



The summary of this dataset looks perfect since there is no negative/ invalid values present.

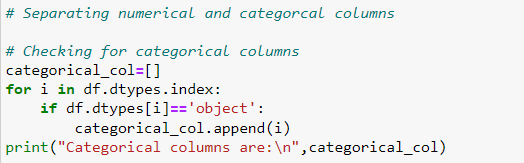
From the above description we can observe the following things:

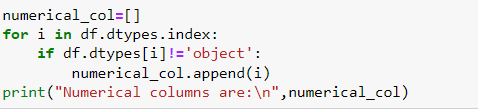
Here the counts of all the columns are equal which means there are no missing values in the dataset. In some of the columns like policy\_deductable, capital-gains, injury\_claim etc we can observe the mean value is greater than the median (50%) which means the data in those columns are skewed to right. And in some of the columns like total\_claim\_amount, vehicle\_claim etc we can observe the median is greater than the mean which means the data in the columns are skewed to left. And some of the columns have equal mean and median that means the data is symmetric and is normally distributed and no skewness present. There is a huge difference in 75% and max it shows that huge outliers present in the columns.

**Data Visualization:**

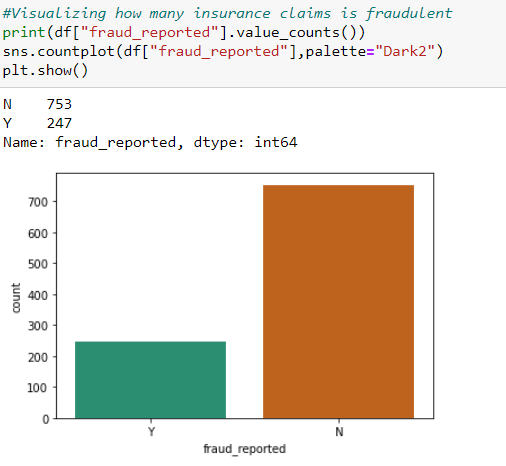
**Preparing for Visualization**

We will look into the categorical and numerical columns so that we can create two different lists and visualize the features accordingly.

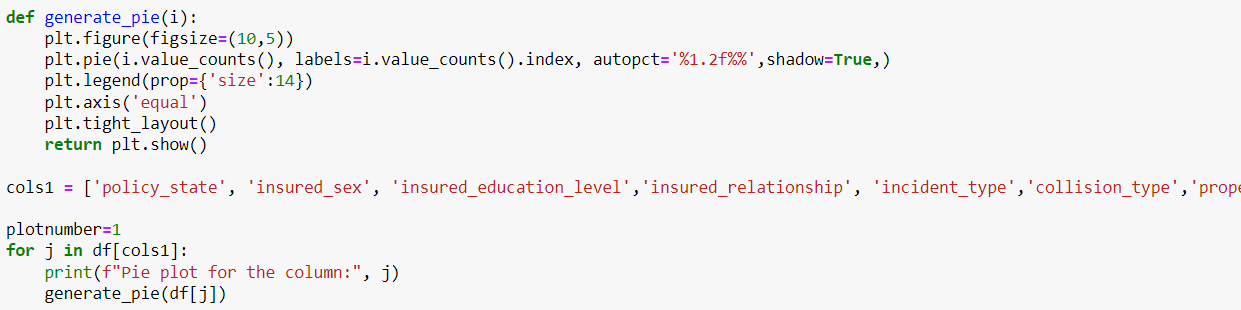


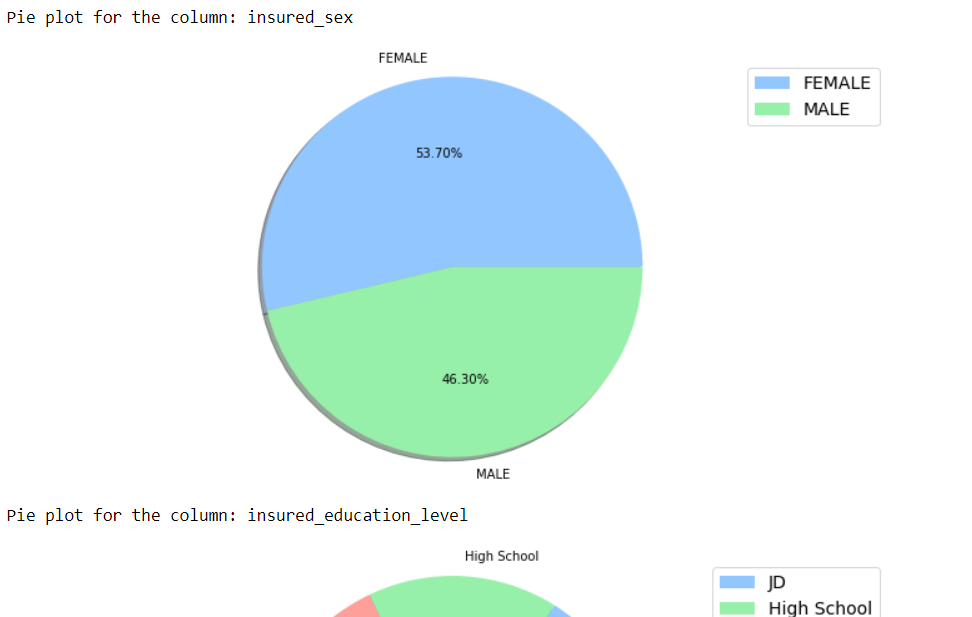


**Univariate Analysis (Categorical Columns)**

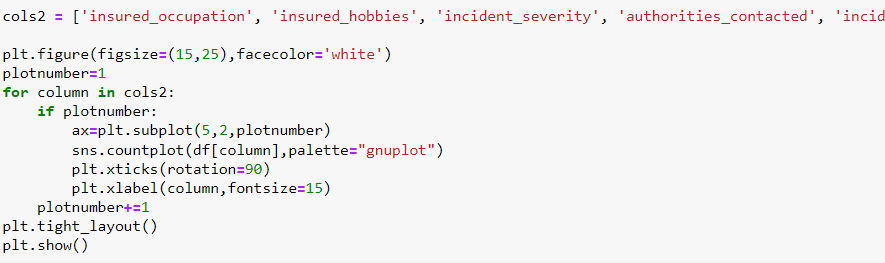


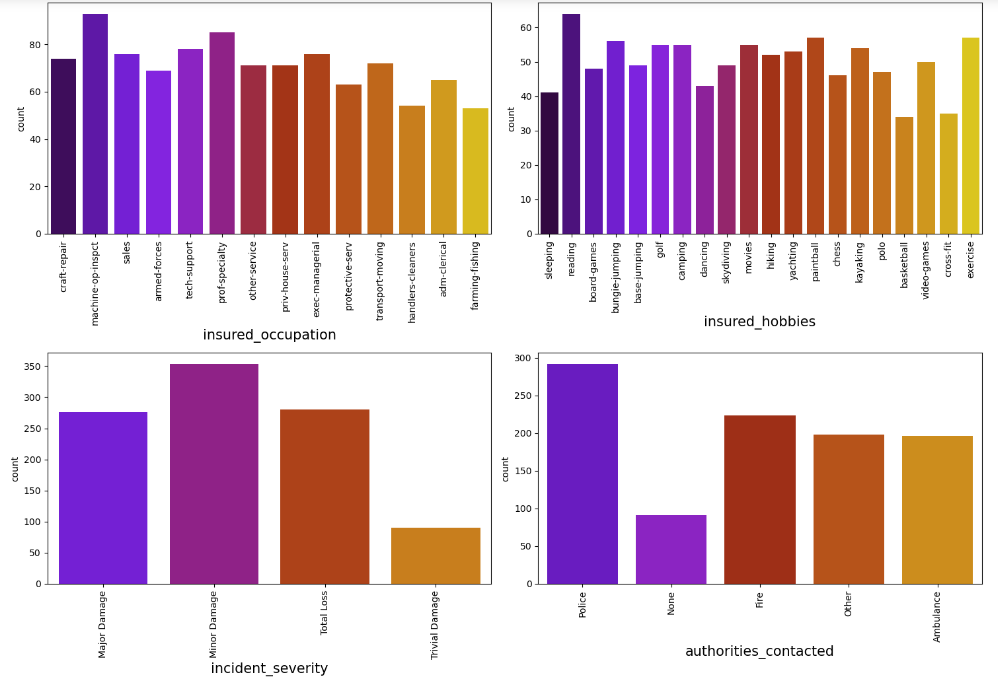
From the plot we can observe that the count of "N" is high compared to "Y". We can assume that "Y" stands for "Yes" that is the insurance is fraudulent and "N" stands for "No" means the insurance claim is not fraudulent. Here most of the insurance claims have not reported as fraudulent. Since it is our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

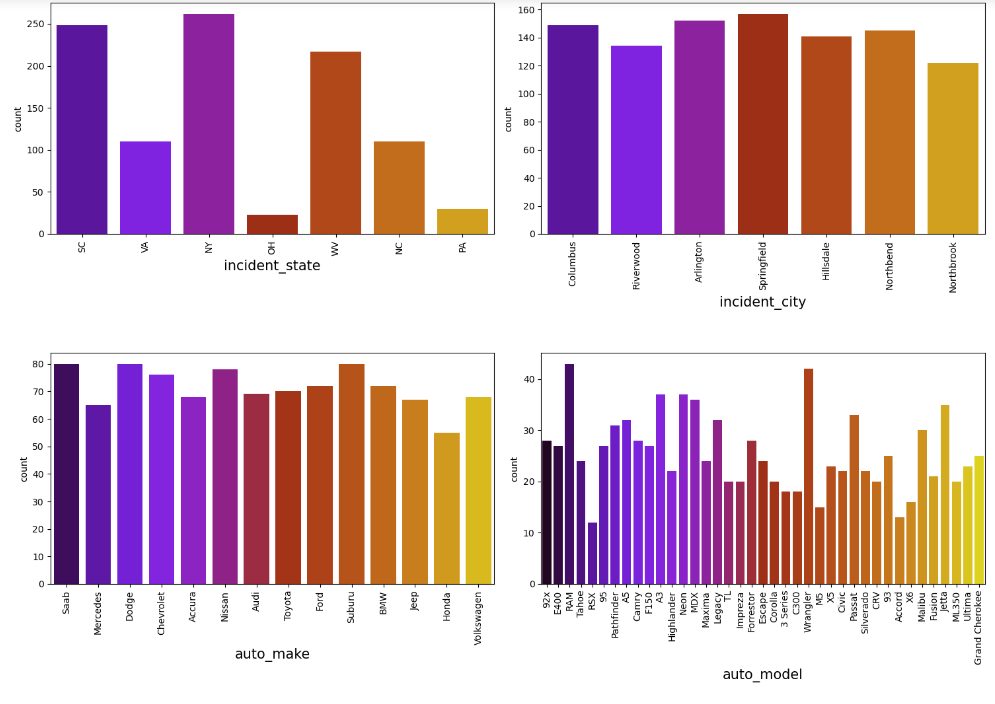




Above are the pie plots for some of the categorical columns



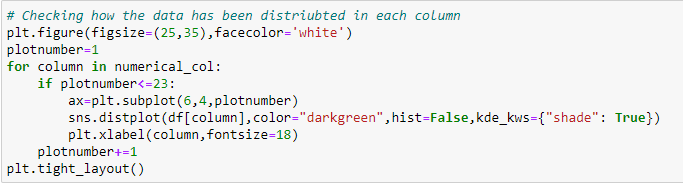




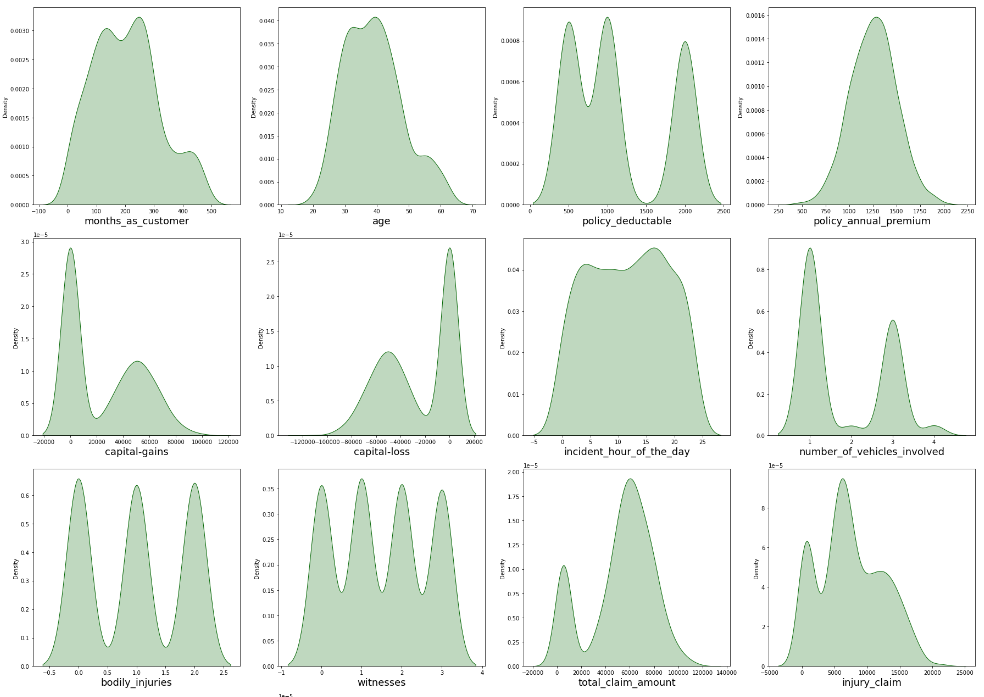
By looking into the count plots, we can observe the following things:

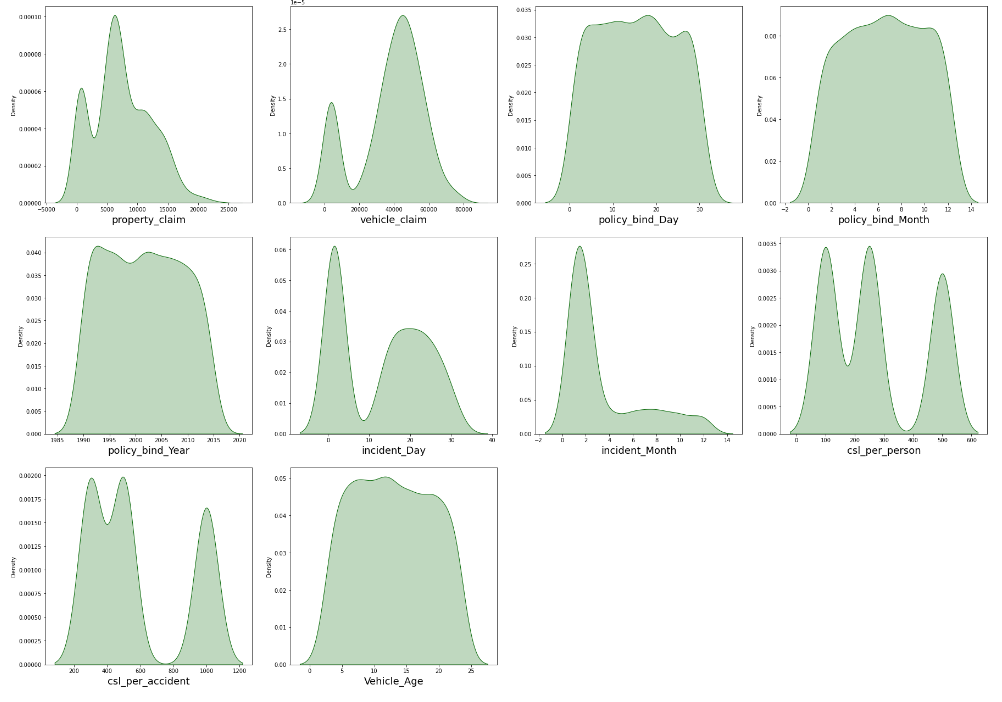
* In the insured occupation we can observe most of the data is covered by machine operation inspector followed by professional speciality.
* With respect to insured hobbies, we can notice reading covered the highest data followed by exercise. And other categories have the average counts.
* The incident 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 cover 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 others.
* With respect to the incident state, New York, South Carolina and West Virginia states have highest counts.
* In incident city all have almost 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.

**Checking the Distribution of the dataset (numerical columns)**



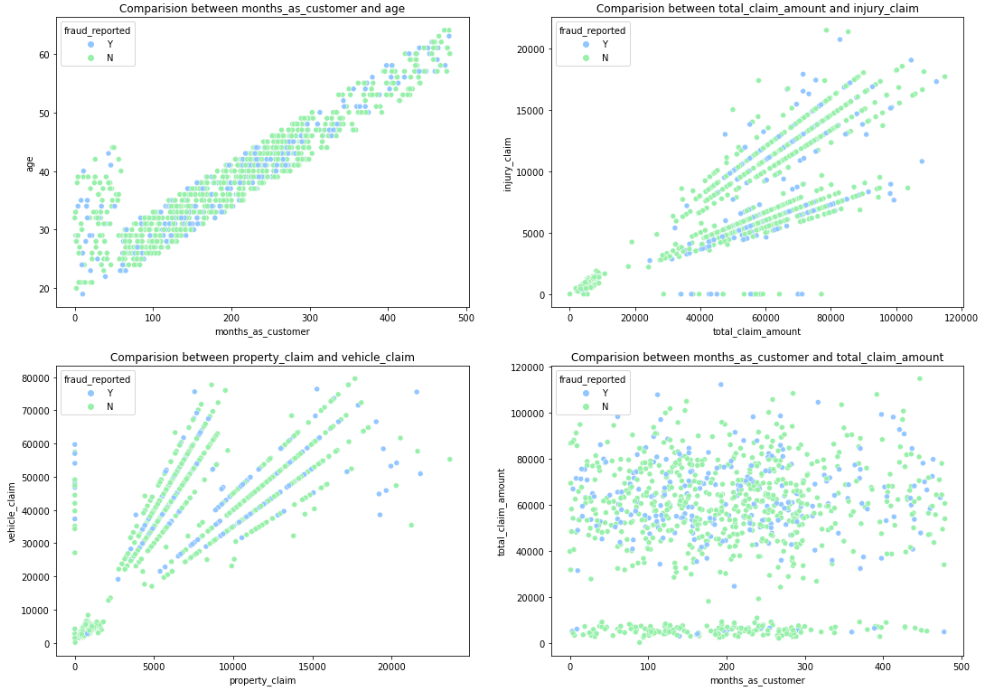
The data is normally distributed in most of the columns. Some of the columns like capital gains and incident months have mean value greater than the median, hence they are skewed to right. The data in the column capital loss is skewed to left since the median is greater than the mean. We will remove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



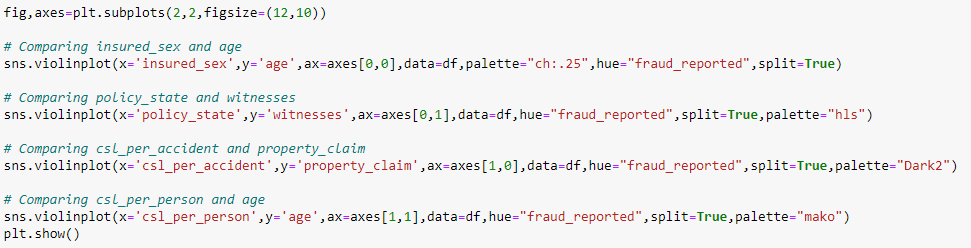


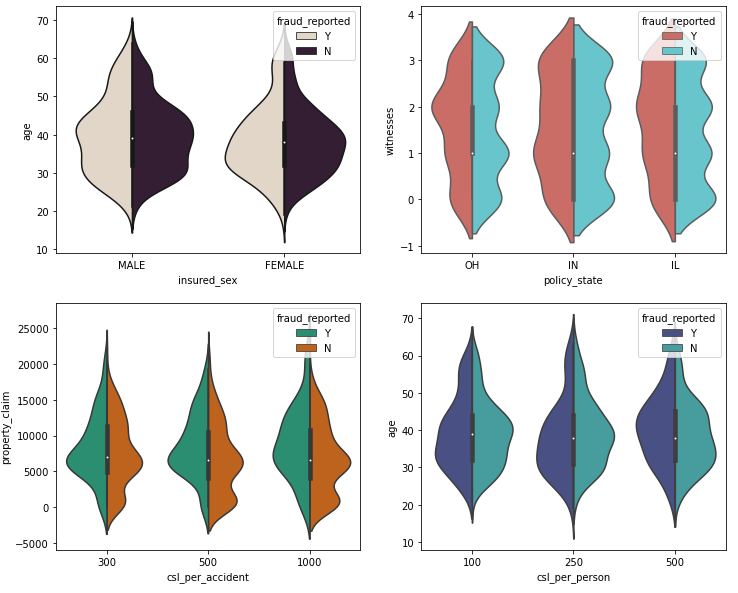


Output:

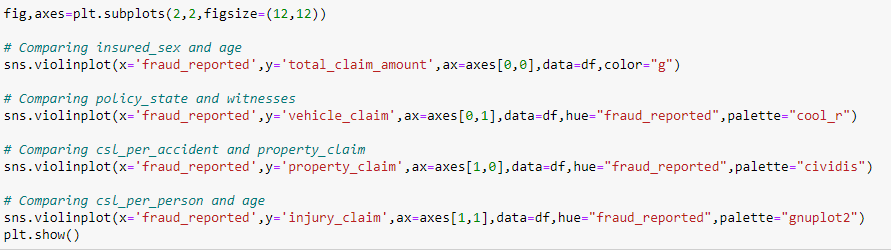


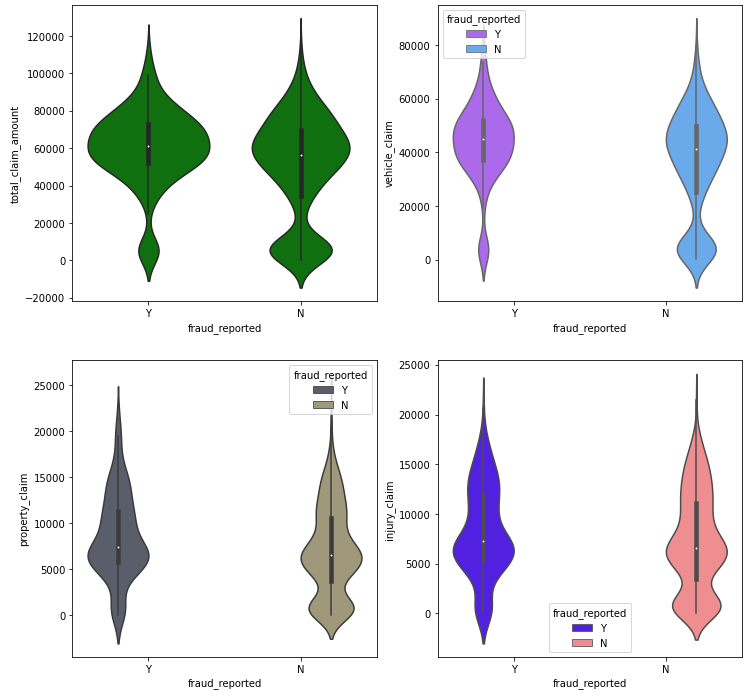
* There is a positive linear relation between age and month\_as\_customer column. As age increases the month\_as customers also increases, also the fraud reported is very less in this case.
* In the second graph we can observe the positive linear relation, as total claim amount increases, injury claim also increases.
* Third plot is also same as second one that is as the property claim increases, vehicle claim also increases.
* In the fourth plot we can observe the data is scattered and there is not much relation between the features.





The fraud report is high for both the males females having age between 30-45. The people who own the policy state "IN" have high fraud report. The person who has csl per accident insurance by claiming property in the range 5000-15000 have the fraud report. The csl\_per\_person with age 30-45 are facing the fraudulent reports.

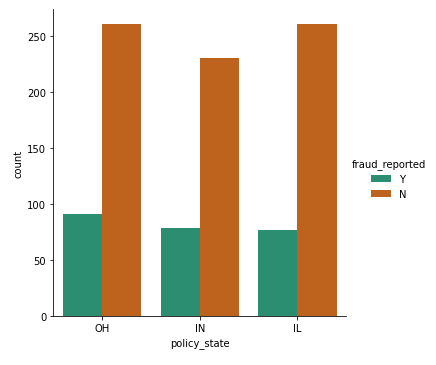


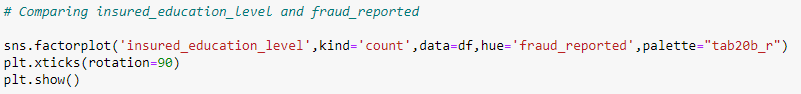


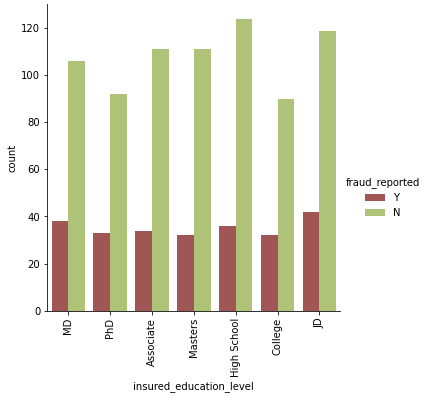
Most of the fraud reports found when the total claimed amount is 50000-70000. The fraud report is high when the claimed vehicle is between 37000-57000. The fraud reports are high when the property claimed is between 5200-8500. Most fraud reported when injury claim are between 5000 to 8000.

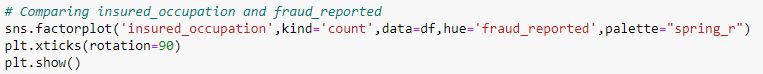
Visualization is a technique where by comparison and plotting the data becomes self-explanatory, which we have seen until now. Moving ahead with some more visualization plots before we can proceed to model building.

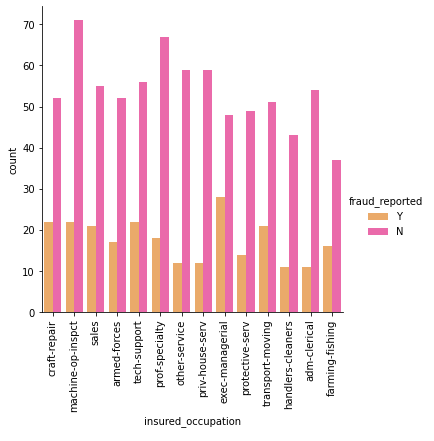


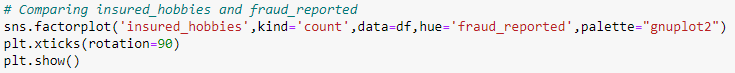


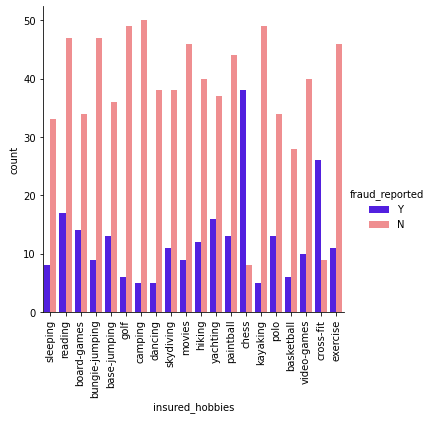


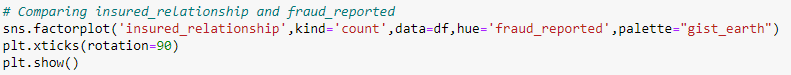


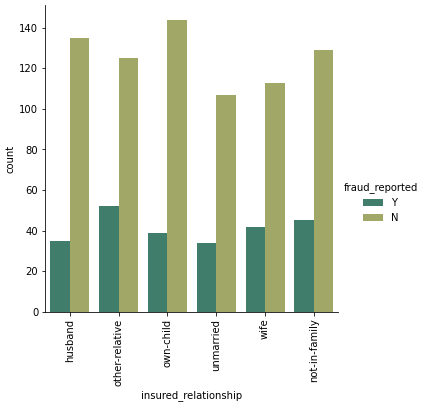


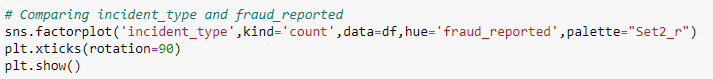


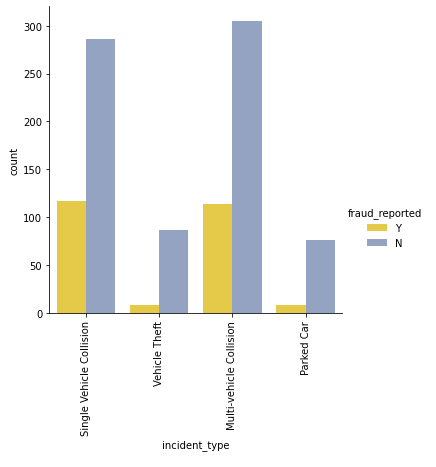


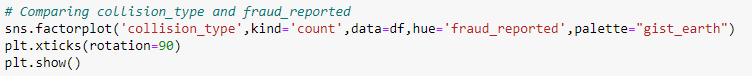


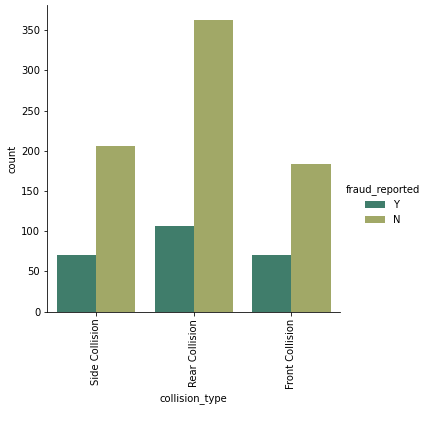


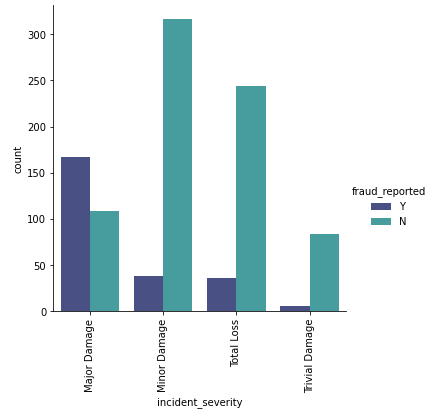
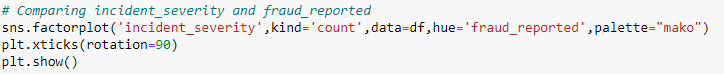




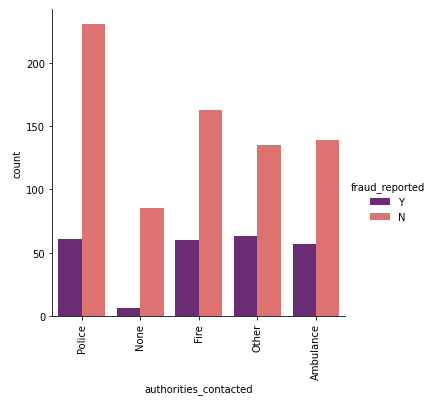


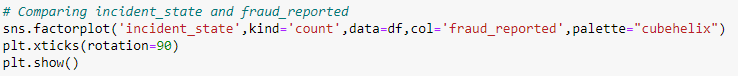


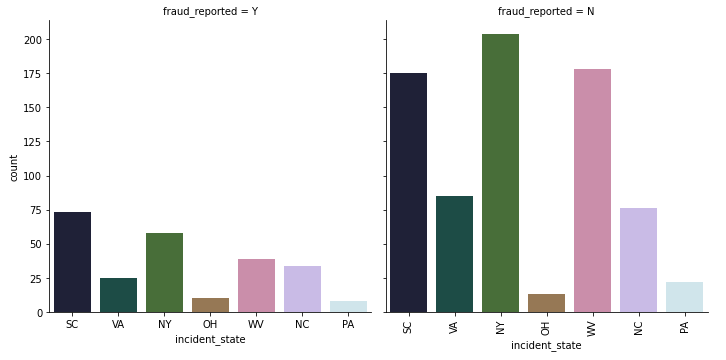


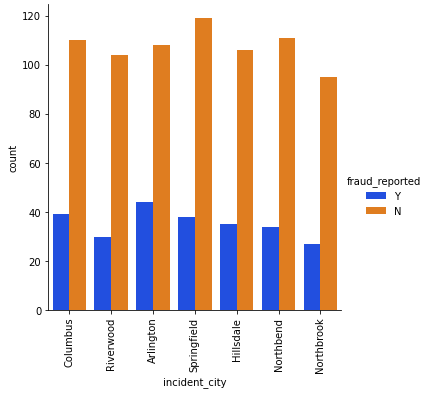
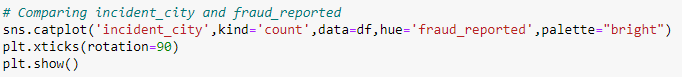




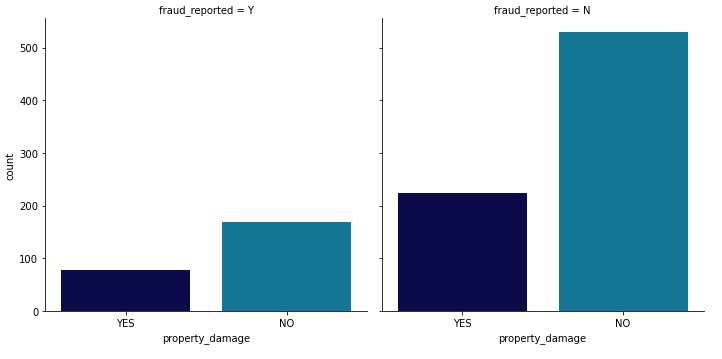


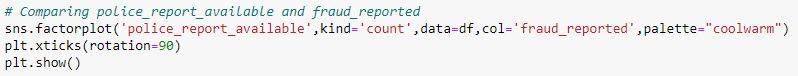


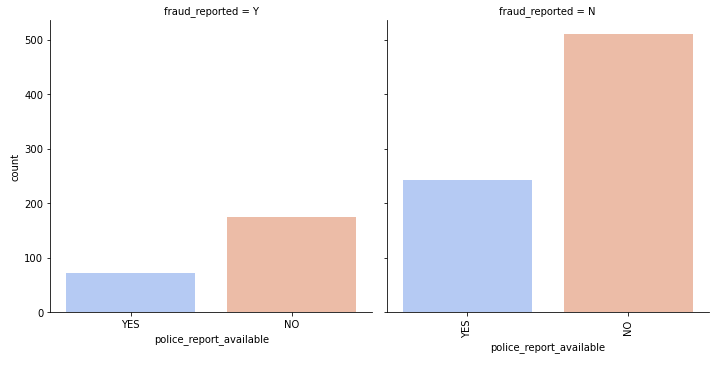


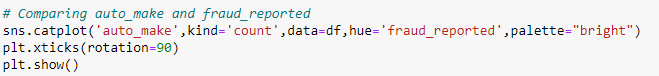


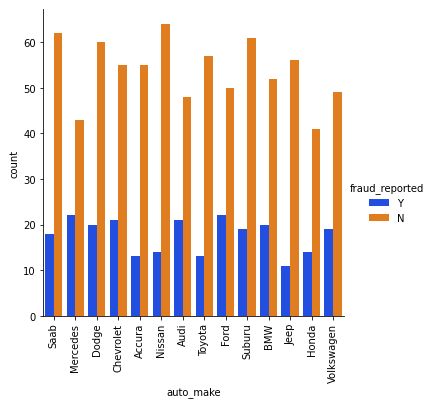


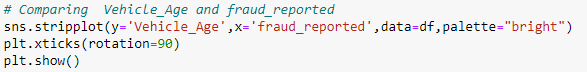








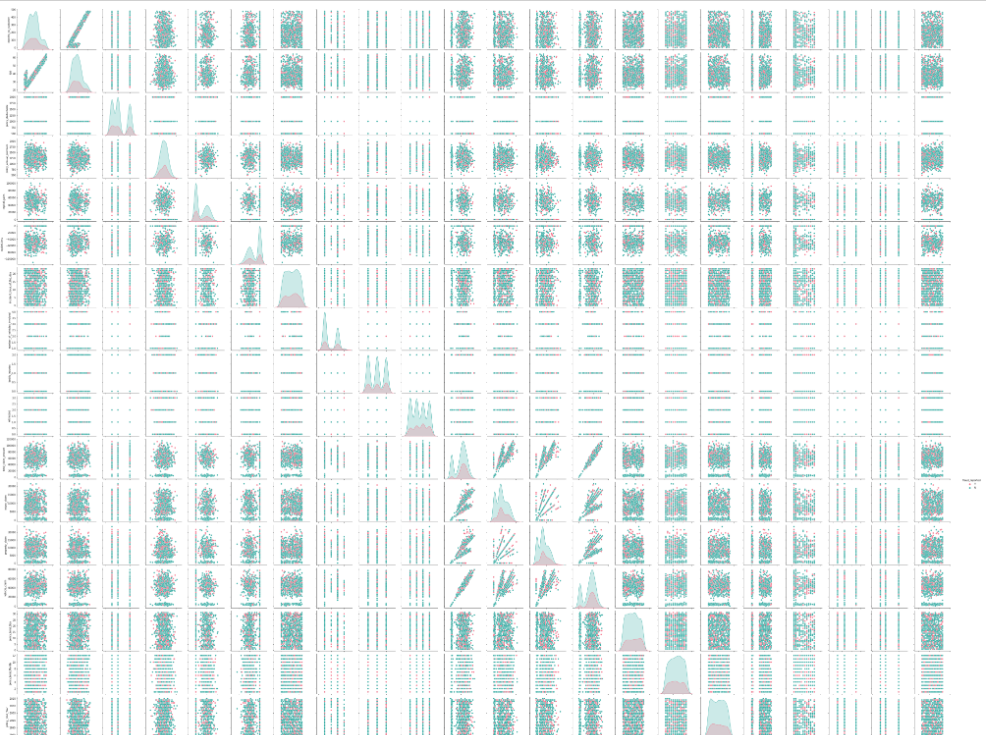






**Multivariate Analysis:**





In the pair plot we can see the relation between each variable with respect to other variables.

**Concluding Remark:**

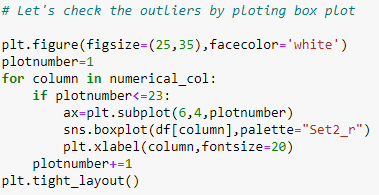
* Fraud report is bit high in the "OH" policy state.
* The fraudulent level is very less for the people who have high school education and the people who have completed their "JD" education have high fraud report. The people who have high insured education are facing insurance fraudulent compared to the people with less insured education level.
* The people who are in the position exec-managerial have high fraud reports compared to others.
* The fraud report is high for the people who have the hobby of playing chess and cross fit.
* The fraud report is high for the customers who have other relative and it is very less for unmarried people.
* In Multivehicle collision and single vehicle collision, the fraud report is very high compared to others.
* The fraud level is high in the collision type Rear Collision and other two collision type have average reports.
* The fraud report is high in Major damage incident severity and for Trivial Damage the report is less compared to others.
* The police contacted cases are very high and the fraud report is in equal for all the authorities except None which is the lowest.
* The state SC has high fraud reports compared to other states.
* The cities Riverwood and Northbrook have very less fraud reports compared to others.
* The customers who do not have any property damage case they have high fraud reports.
* If there is no police report available then the fraud report is very high.
* In all the auto make cases the fraud report is almost same.
* There is no significant difference between the features in Vehicle\_Age vs fraud\_reported.

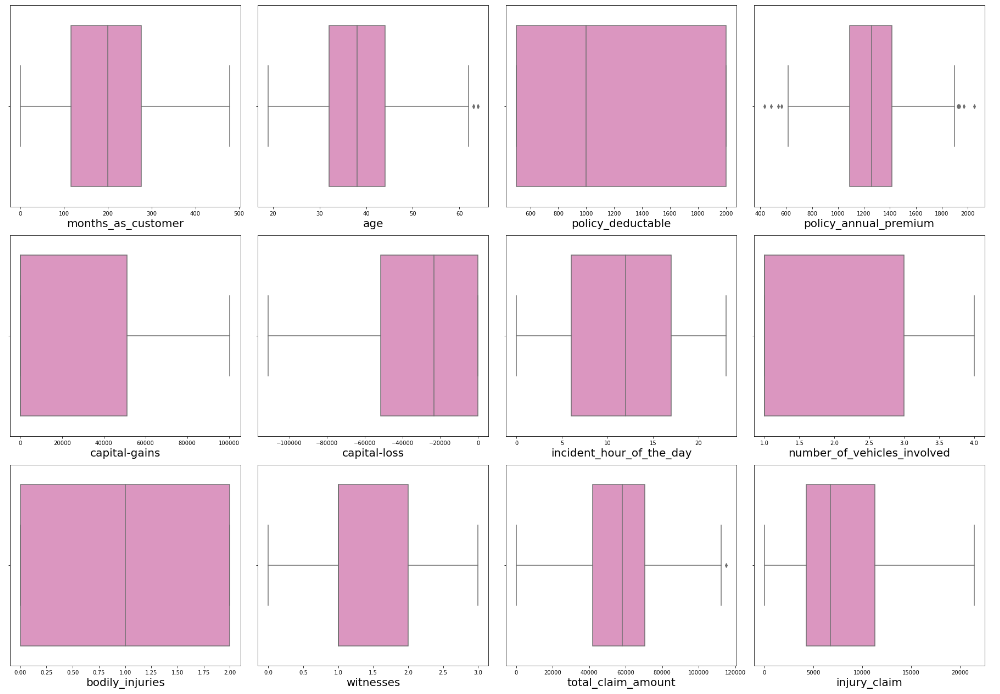
Now we have done with the visualization in order to analyze and understand the data. So, in this EDA part, we have looked into various aspect of the dataset, like looking for the null values and imputing, extracting date time, observing the value counts and doing the feature extraction etc.

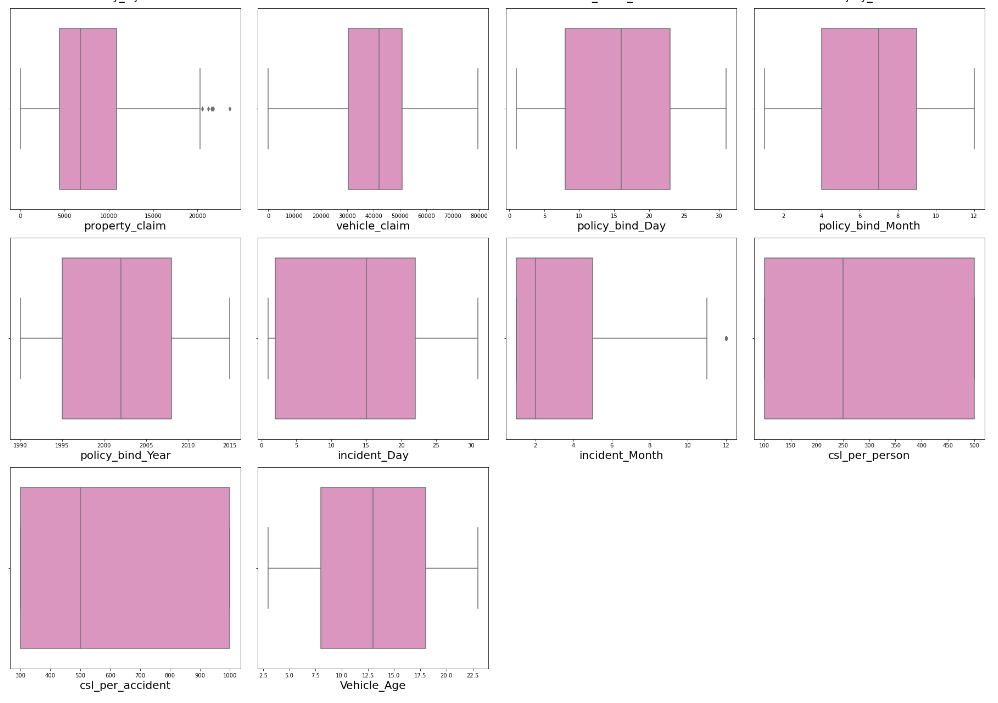
Now we will be performing Data Pre-processing by identifying the outliers and removing them. Along with it we will also look for the skewness of the dataset and remove the skewness.

1. **Pre-Processing Pipeline**

**Identifying the outliers**

****

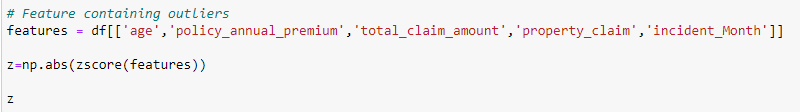


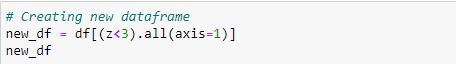


We have used box plot to identify the outliers and we can find the outliers in the following columns:

policy\_annual\_premium, total\_claim\_amount, property\_claim and incident\_month.

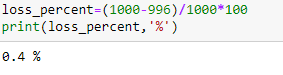
These are the numerical columns which contains outliers. Removing the outliers in these columns using Zscore method.





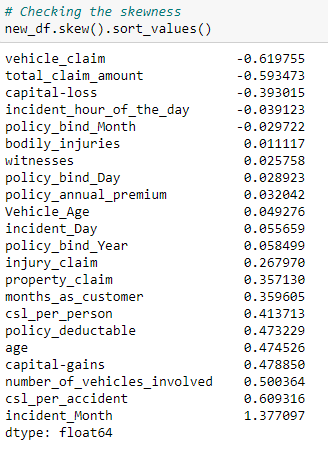
This is the new dataframe after removing the outliers. Here we have removed the outliers whose Zscore is less than 3.

**Percentage data loss:**

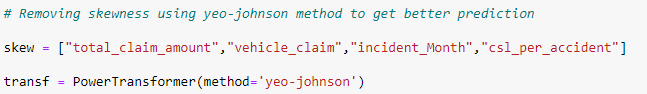


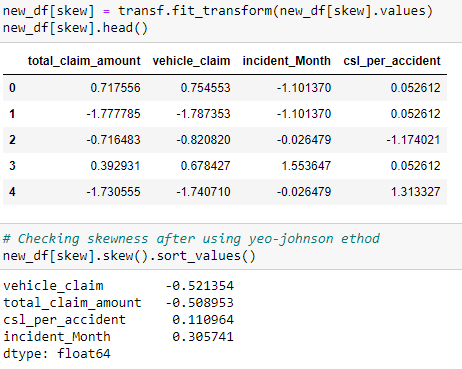
After removing the outliers, we are checking the data loss percentage by comparing the rows in our original data set and the new data set and 0.4% data loss is in the acceptable range.

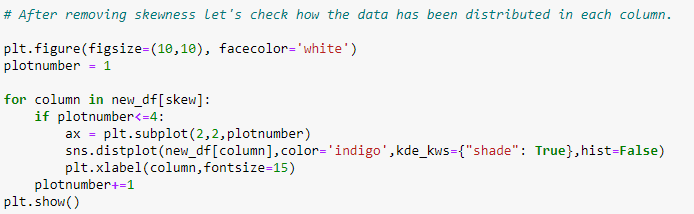
**Checking skewness in the data:**

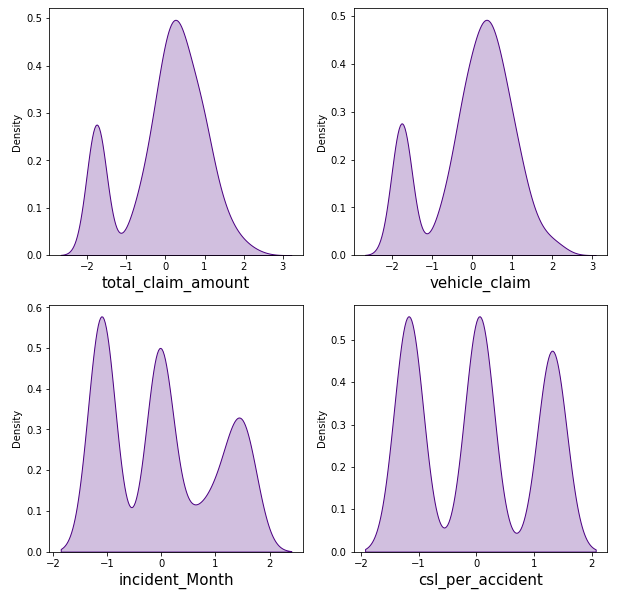


As we can see that skewness is present in the dataset, hence I am using the yeo-johnson method to remove the skewness.

****

****





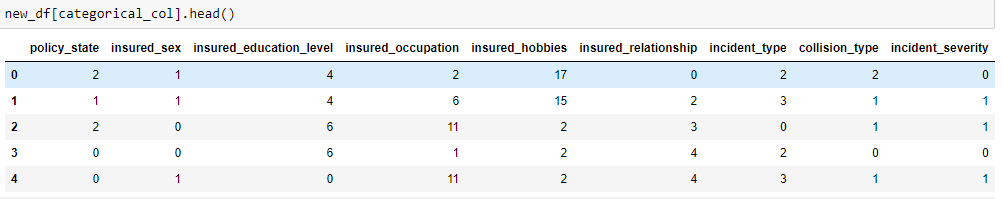
The data distribution looks better after removing the skewness compared to the previous data.

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

However, we have seen that the dataset has both numerical and categorical data. The model only understands numerical data; hence we will encode the categorical data. Also, we have seen that there can be some multicollinearity, that we will see through a heatmap and also further remove it. Again, we have also seen that the target variable is imbalanced, hence we will fix it by oversampling. And finally, we will scale the data so that it is ready to be trained and tested.

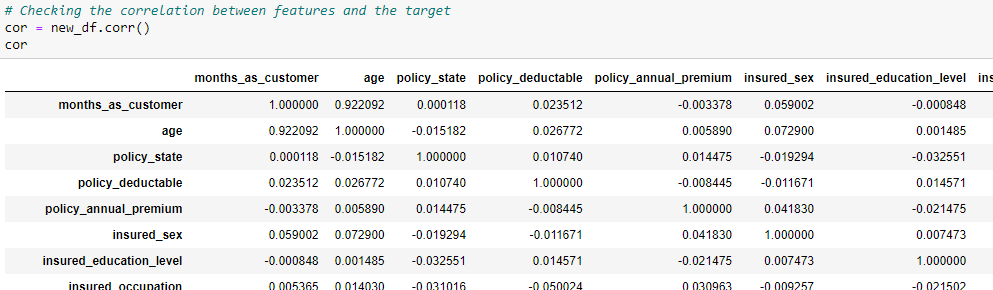
**Encoding the categorical columns using Label Encoding**

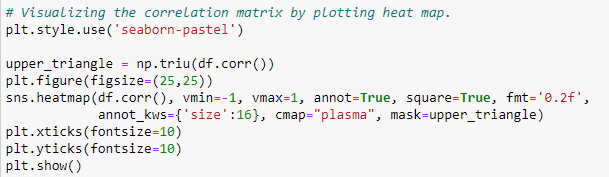
****

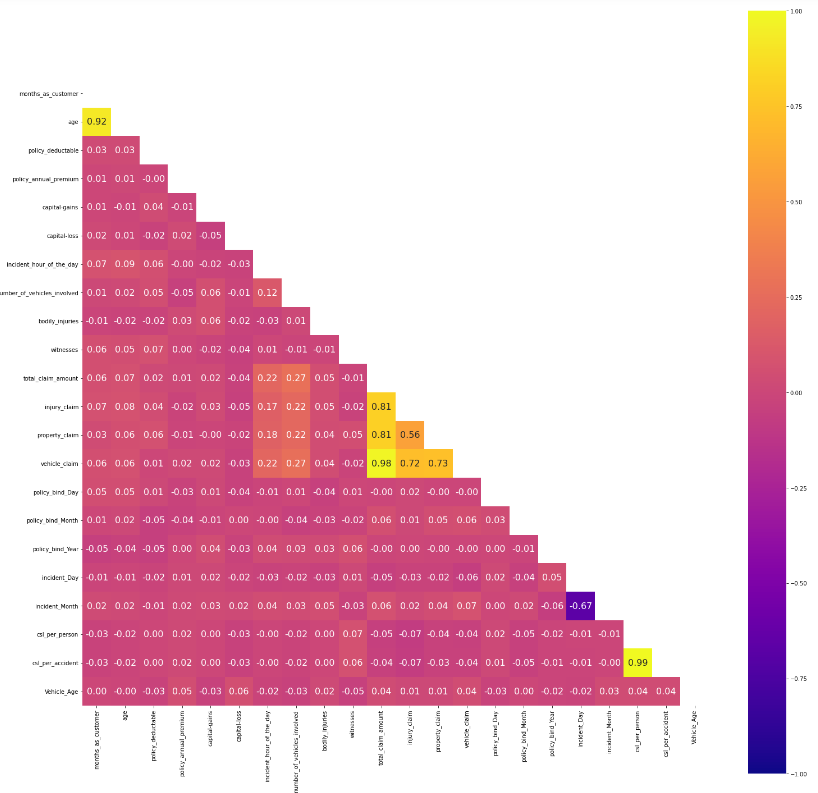


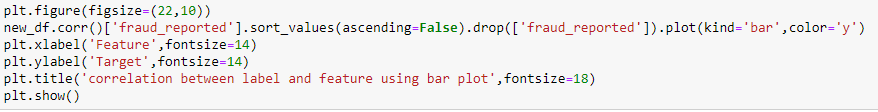
Now we have encoded the dataset using label encoder and the dataset looks like above.

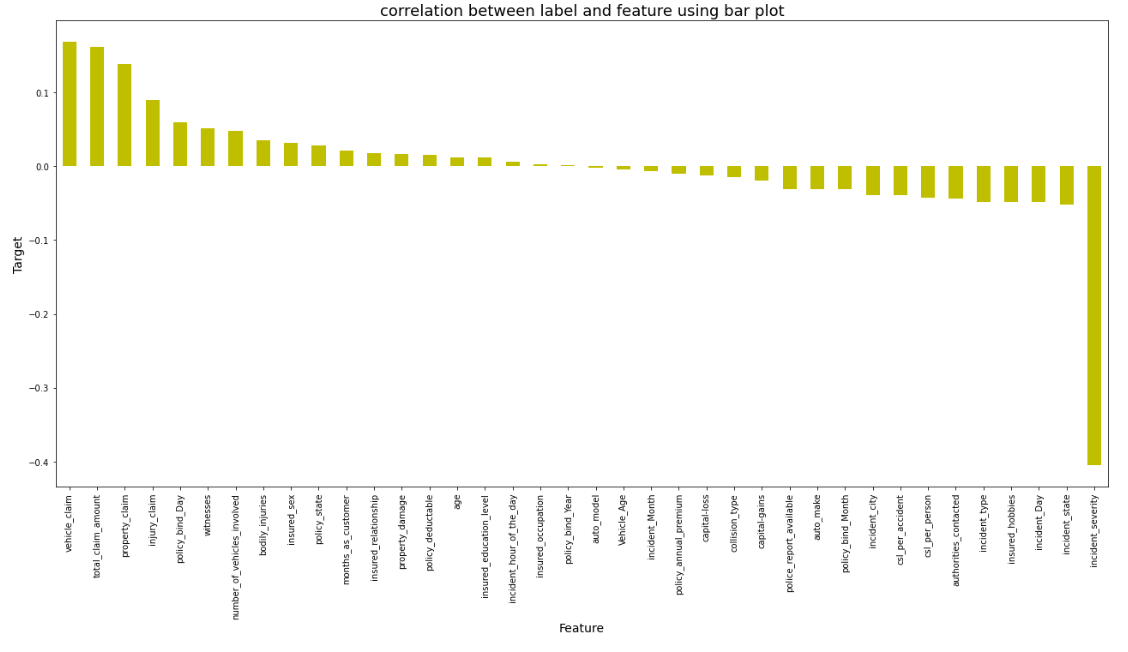
Moving forward, to check the correlation between the feature and target and also the relation between the features using the heatmap.









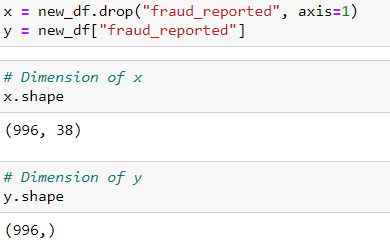


This heatmap shows the correlation matrix by visualizing the data. We can observe the relation between one feature to another.

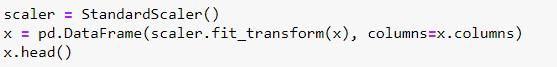
There is very less correlation between the target and the label. We can observe that most of the columns are highly correlated with each other which leads to the multicollinearity problem. We will check the VIF value to overcome with this multicollinearity problem.

Since the heatmap was not able to give us a clearer picture on positive and negative correlation columns we have generated the bar plot to get a better idea and we see that more than half the feature columns are positively correlated with our target label while all the remaining features are negatively correlated with our label column. This indicates that they are all required for the prediction of our classification label.

**Splitting the dataset into Features and Target**



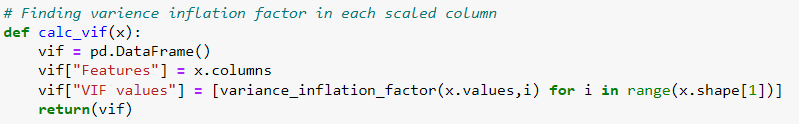
## Feature Scaling using Standard Scaler

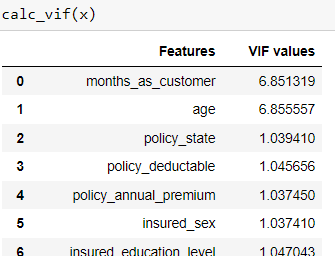


The data has now been scaled.

In the heat map we have found some features having high correlation between each other which means multicollinearity exists. So, let's check the VIF value to solve multicollinearity problem.

**Checking Multicollinearity**

****

****

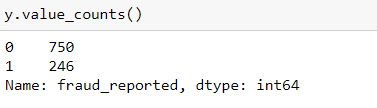
We can observe that some columns have VIF above 10 that mean they are causing multicollinearity problem. Let's drop the feature having high VIF value amongst all the columns.

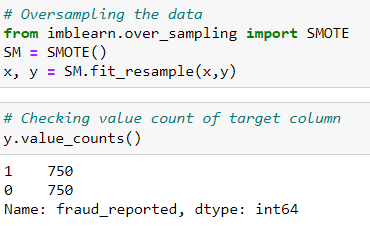




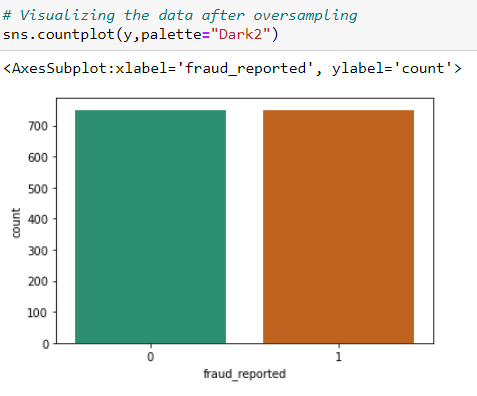
I have dropped the total\_claim\_amount and csl\_per\_accident features with VIF more than 10. Now we have overcome the multicollinearity issue as all the VIF values are less than 10 in all the columns.

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 we can proceed with modelling.

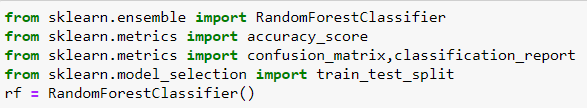


1. **Machine Learning**

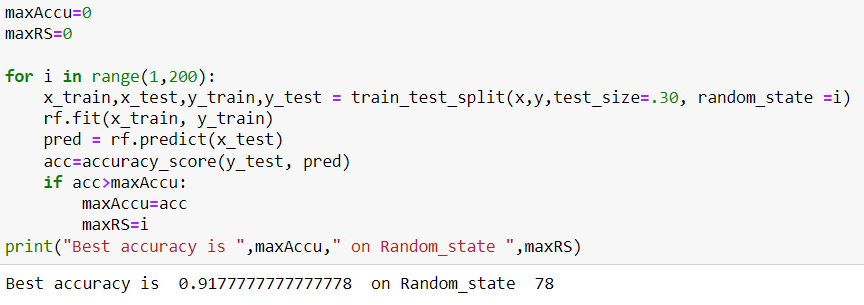
**Finding best random state**

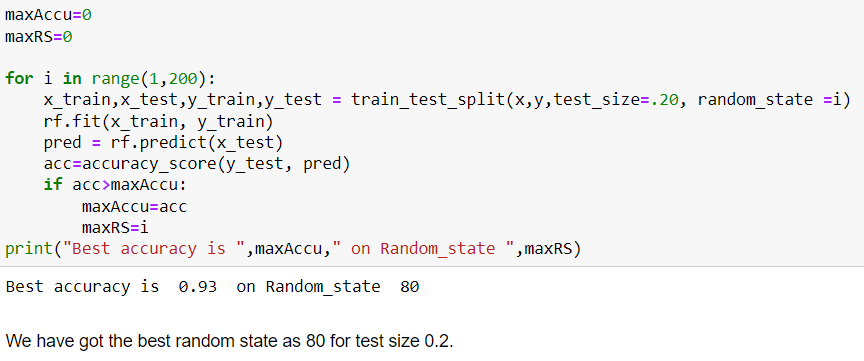
let’s find the best random state in which we can build the model*.*

(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.)



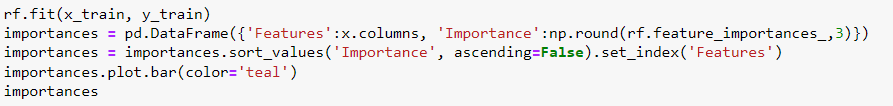
For Test size of 0.30

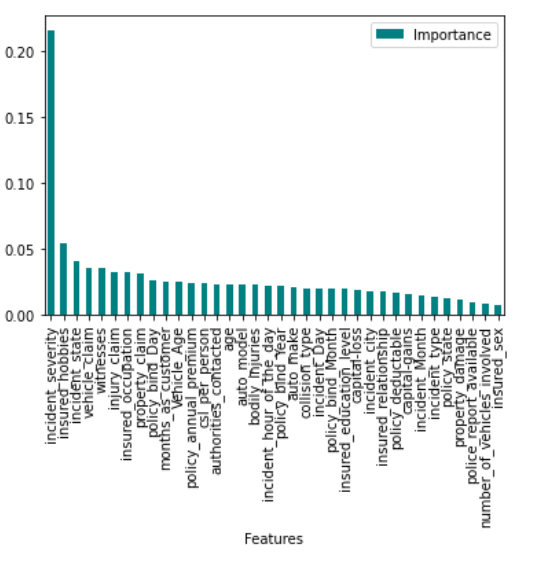


For Test size of 0.20

Here we have used the RandomForestClassifier to find the best random state and got an accuracy score of 93% at the random state of 80. Let’s use this random state to build our models.

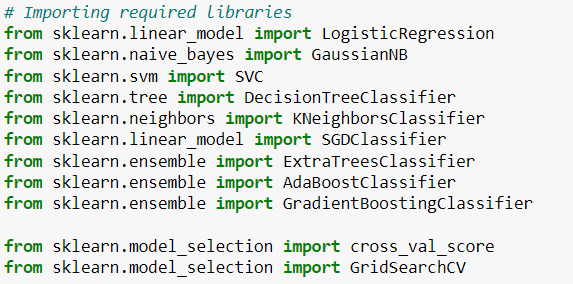
**Feature importance bar graph**

****

****

This bar plot shows us the importance of the features using random forest algorithm on predicting our Target variable.

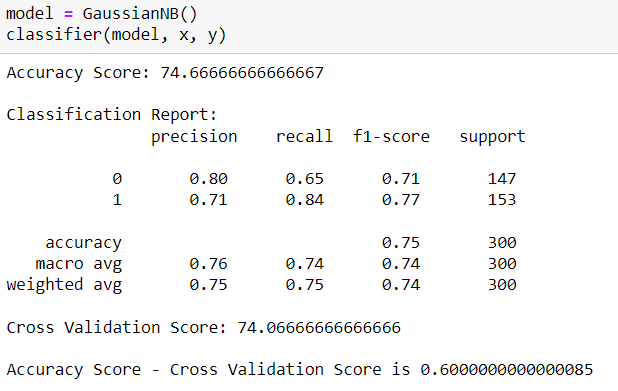
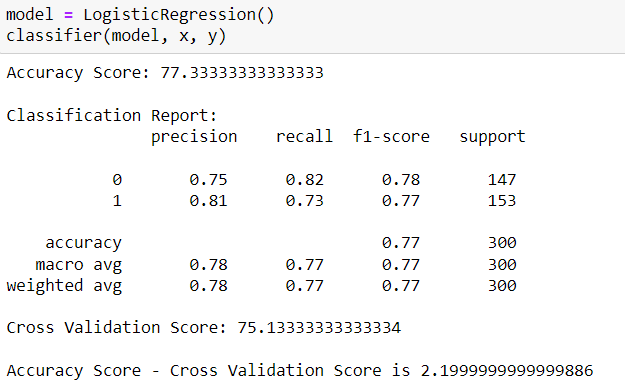
We will import all the required libraries as shown below.

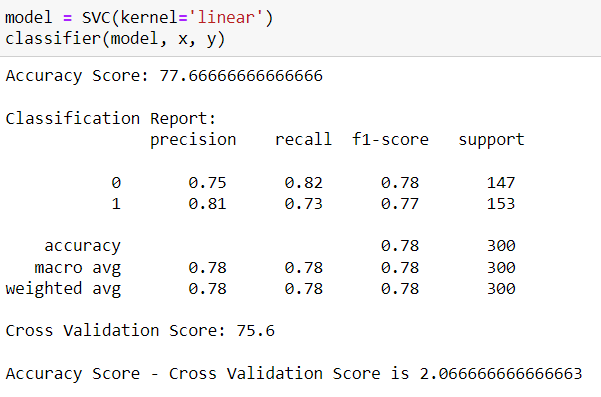
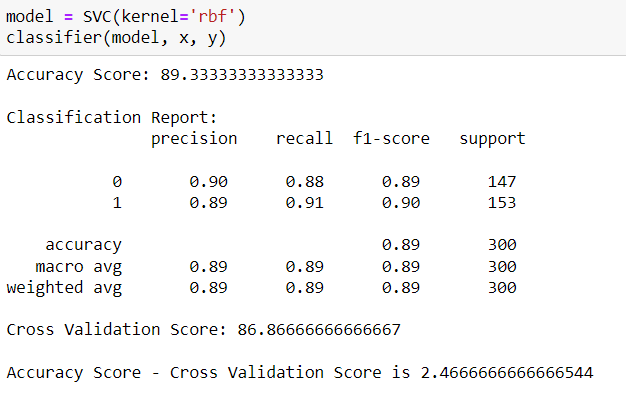


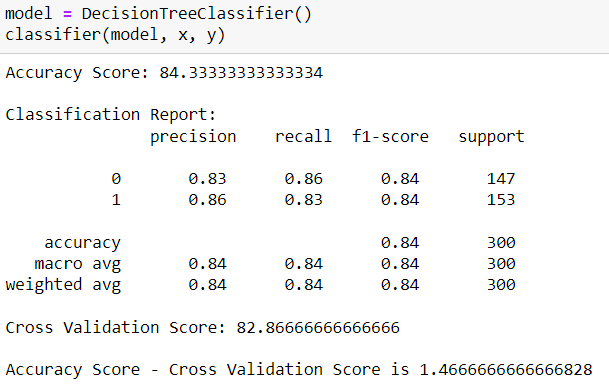
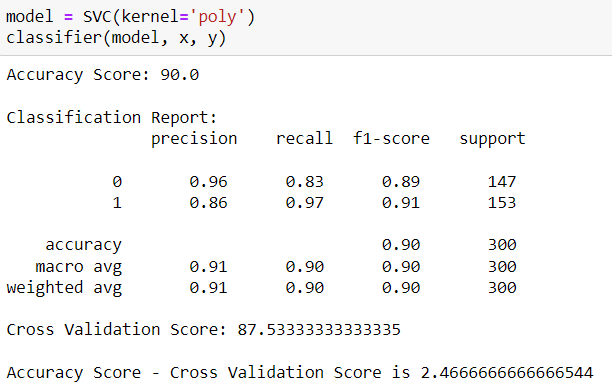
I have created a function which can be used for all the classifier machine learning algorithms and get the Accuracy score, Classification Report, Cross Validation Score and Accuracy Score - Cross Validation Score for comparison between all the algorithms for finding out the best suited algorithm for Hyper Parameter Tuning.

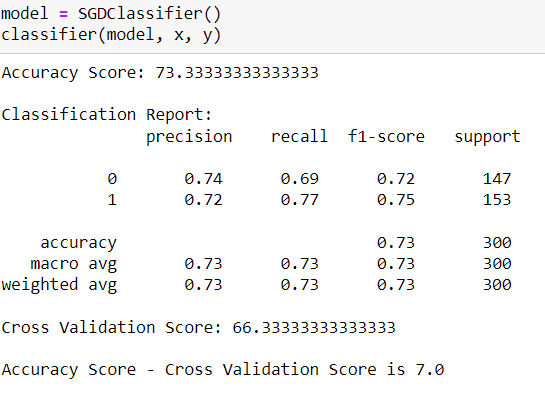
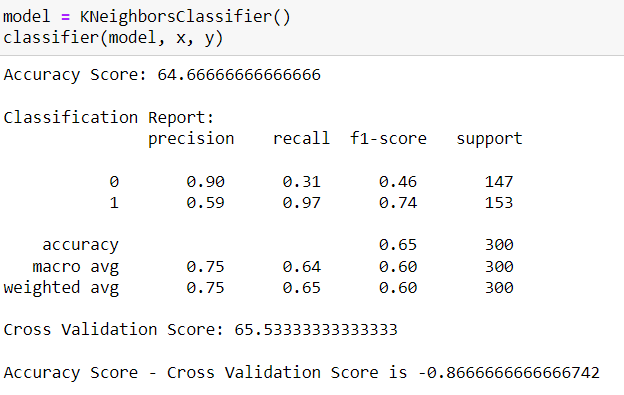
**Machine Learning Algorithms**

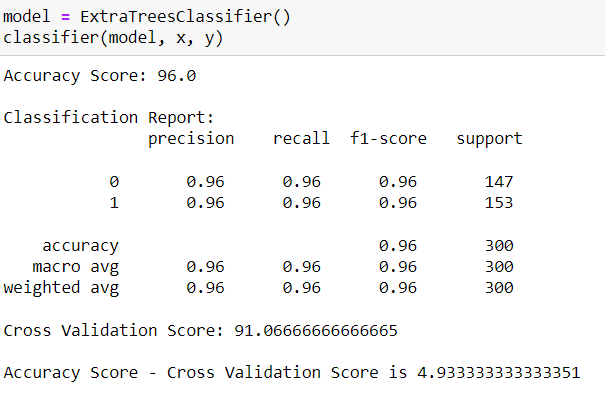
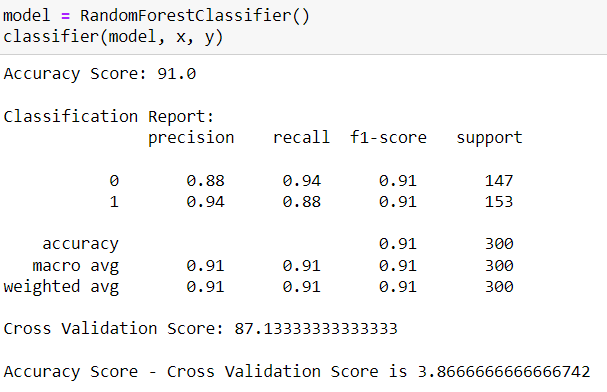
We will run various algorithms and check which has the highest Accuracy score, Cross Validation Score and (Accuracy Score - Cross Validation Score) for comparison between all the algorithms.

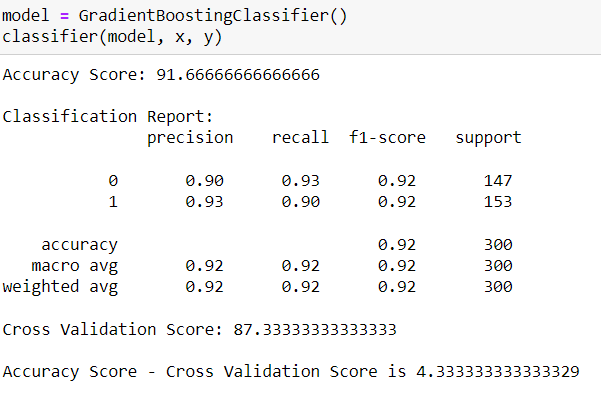
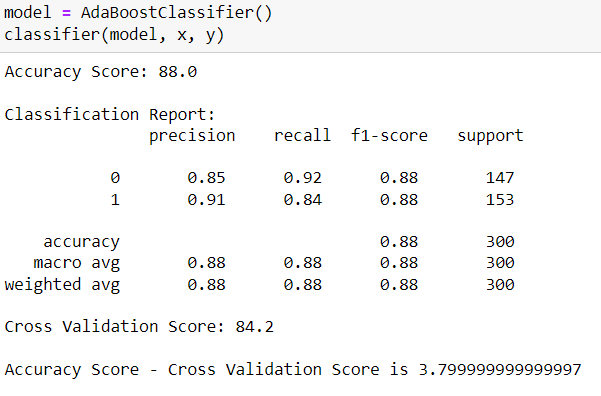












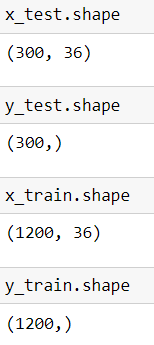
Referring all the above algorithms we can see that ExtraTreesClassifier gives the best results since the (Accuracy Score - Cross Validation Score) is the least comparing others while having higher Cross Validation Score and the highest Accuracy Score comparing all the other models.

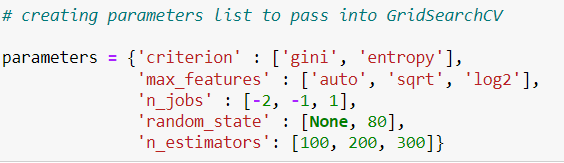
Now, that we have found the best fit model, lets perform Hyper Parameter Tuning to improve the performance of the model.

**Hyper parameter tuning**

Creating train\_test\_split and checking the shape of the subsets.

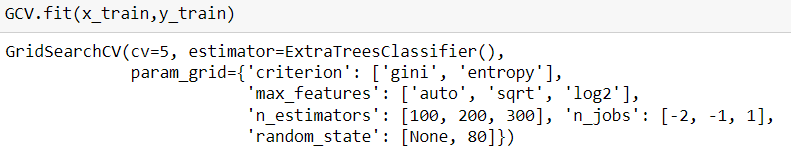




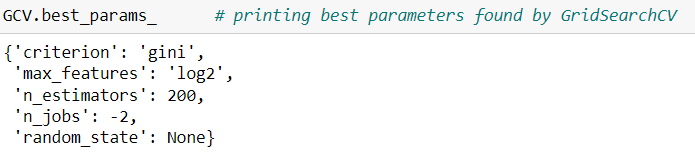
Creating a list of parameters to pass into the Grid Search CV.

Running Grid Search CV for ExtraTreesClassifier at cv = 5

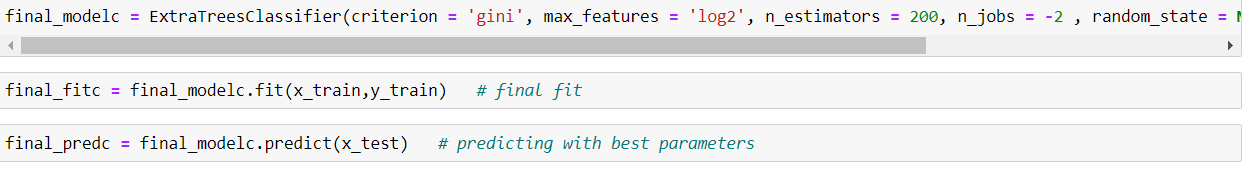




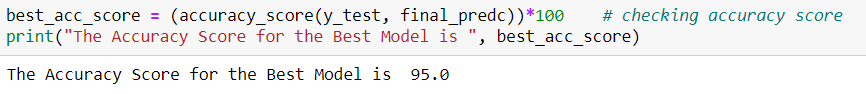
Getting the list of the best parameters from Grid Search CV.



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

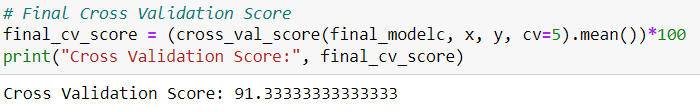


The Final Accuracy Score of the final Model.



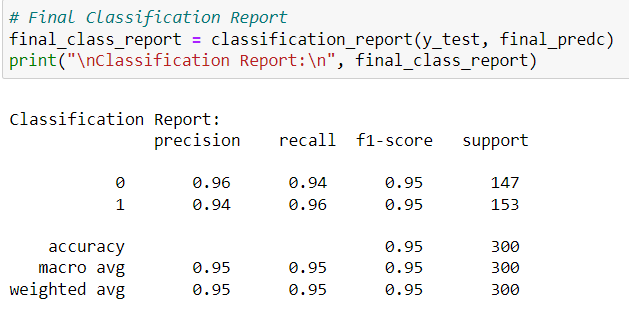
We successfully performed the Hyper Parameter Tuning on the Final Model.

The Final Cross Validation Score of the final Model.

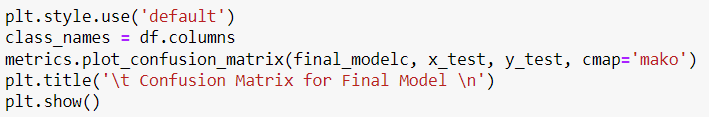


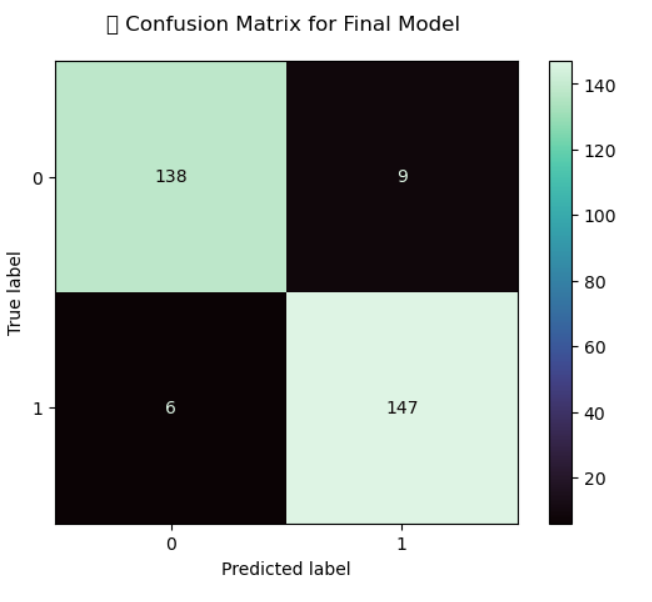
We got final accuracy score of 95% and Cross Validation Score of 91.3333% which is really good.

The Final Classification Report of the final Model.



The Final Confusion Matrix of the final Model.

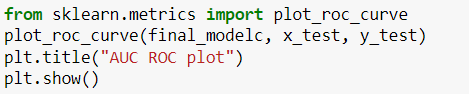


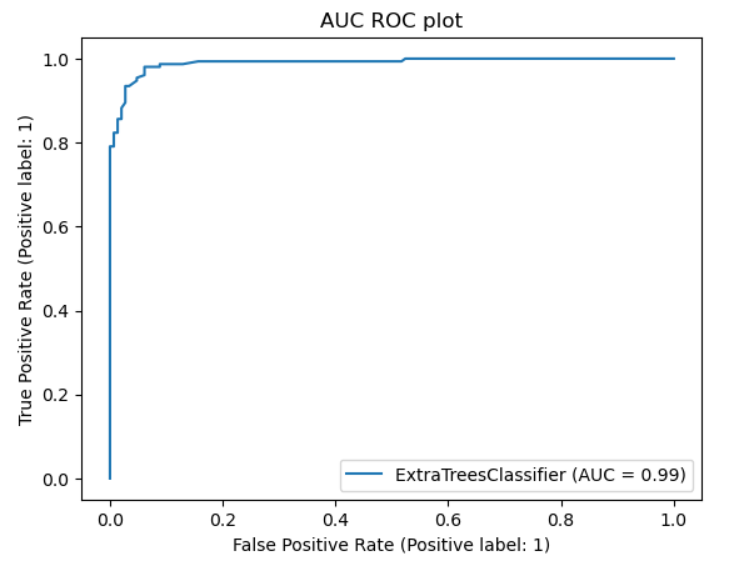


With the help of above confusion matrix, we are able to understand the number of times we got the correct outputs and the number of times my ML model missed to provide the correct prediction (depicted in the black boxes).

**AUC ROC curve**

Plotting and AUC ROC curve for the final model.

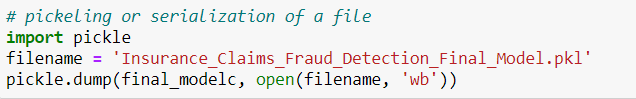




So here we can see that the area under curve is really good for this model.

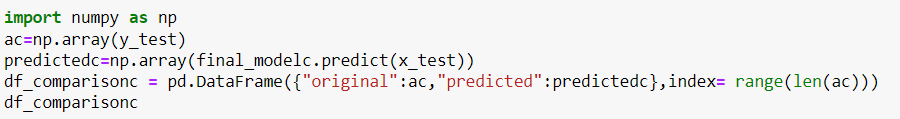
We got final accuracy score of 95% and Cross Validation Score of 91.3% and also AUC score is 0.99 which is really good.

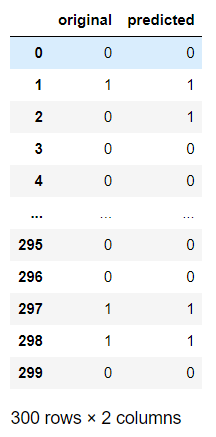
**Saving the model in pickle Format**

****

**Prediction Conclusion**

We will predict the "fraud\_reported" target column using the final model sending the “x\_test” set for predicting the “y\_test” and then compare the original “y\_test” and the predicted “y\_test” in a dataframe.



****

Hence predicted the "fraud\_reported" using the final Model and presented it as a data frame to compare the predictions.

Saving the comparison file as a csv file.



Saved the file as a csv for future reference purposes.

1. **Concluding Remarks**

In the beginning of the blog, we have discussed about the lifecycle of a Machine Learning Model, you can see how we have touched based on each point and finally reached up to the model building and made the model ready for deployment.

Let’s take a quick recap on all the steps that we went through starting from understanding the Problem Definition then going through the Data Analysis and EDA processes. We went through the necessary Pre-processing Data steps before the final Building Machine Learning Models step came into picture.

The Insurance industry area needs a good vision on data, and in every model building problem Data Analysis and Feature Engineering is the most crucial part.

You can see how we have handled numerical and categorical data and also how we build different machine learning models on the same dataset. And using hyper parameter tuning we can improve our model accuracy. Hence, we ended up with a prediction accuracy of 95% after all the above steps are completed.

Using this machine Learning Model, we can predict whether the insurance claim is fraudulent or not and we could reject those application which will be considered as fraud claims. Saving the Insurance industry from possible fraudsters.

**Thank You**