Data Analysis on Insurance fraud claims detection

Submitted by

Dinesh Mutha

**INTRODUCTION TO PROJECT**

Insurance definition: “ An agreement by which a person pays premium to a company and the company promises to pay money if the person becomes injured or dies or to pay for the value of property lost or damaged

False insurance claims are insurance claims filed with the fraudulent intention towards an insurance provider.

Fraud is well identified problem that insurance companies face throughout the tenure of insurance. We will work on Automotive insurance fraud data for this blog.

Fraud is conducted by or against an insurer (insurance company) or an agent for his financial improvements.

Its not just the insurer or the agents, who commit frauds, there are policy holders or third claimants, company employees too who do frauds for financial gains.

Common frauds like inflating of claim values, misrepresentation of facts, submission made fraudulently for damages which never happened, staged accidental claims etc.

**Hardware and Software Used**

Processor Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz

Installed RAM 12.0 GB

System type 64-bit operating system, x64-based processor

HDD : 1 TB

Softwares used: Jupyter Notebook

Libraries Used:

Python

Numpy

Pandas

Matplotlib

Seaborn

Scikit Learn

xgboost

**Problem Definition**

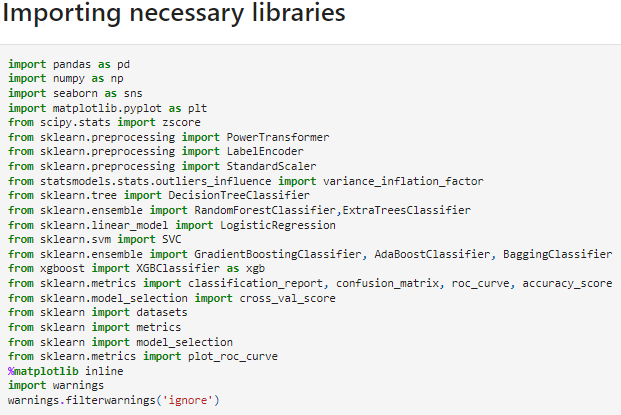
Insurance frauds are huge problem in insurance industry. It is very difficult to identify fraudulent claims unless we use Machine Learning. In this project we were provided with automotive insurance dataset having details of insurance policy, customer detail, accident type on which claims were presented.

Here we will learn how to create a predictive model to check if insurance claim is fraudulent or not.

**Step by Step process of Data Analysis.**

|  |
| --- |
| **PROBLEM DEFINITION** |
| **DATA SELECTION** |
| **EXPLORATORY DATA ANALYSIS** |
| **DATA PROCESSING** |
| **TRANSFORMATION** |
| **FEATURE SELECTION** |
| **MODEL SELECTION** |
| **MODEL TRAINING** |
| **MODEL EVALUATION** |
| **MODEL DEPLOYMENT** |

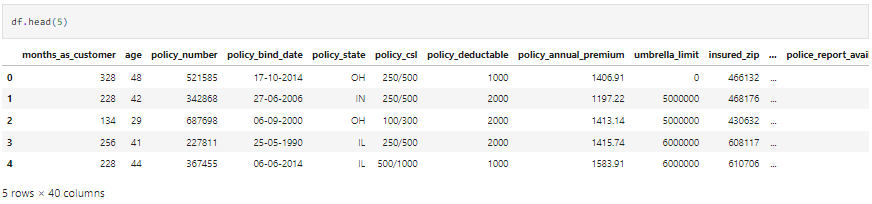
Now we know the life cycle of Machine Learning Model, we continue by importing the necessary libraries.



**Importing the dataset**

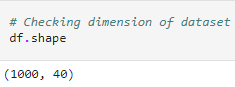


Imported dataset as ‘df’ which was in ‘.csv’ format as below



Dataset contained both numerical as well as categorical data columns. Here ‘fraud\_reported’ is our target column which has categories so this was a ‘Classification Problem’ in which we need to find whether claims are fraudulent or not.

We checked the no. of Rows (1000) and Columns (40) as below

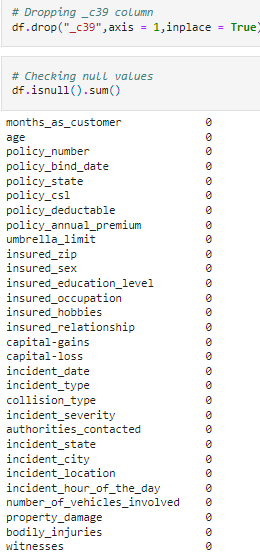


**Exploratory Data Analysis**

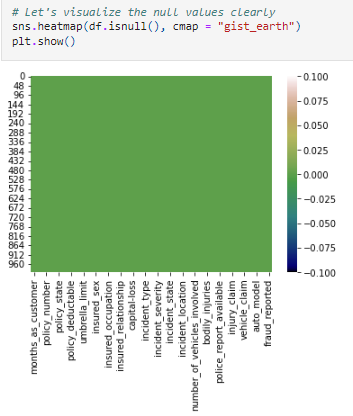
Herein we explored the data extense. Then we continued with our analysis part like feature extraction, imputing & encoding.



This gives the types of index, type of columns, no-null values and memory usage.We found \_c39 column had one unique value hence we dropped the column.

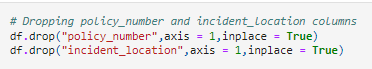


No null values were present further. We checked it through HEATMAP too



Here we found 3 data types namely integer, float and object.

We also found columns policy\_number & incident\_location have 1000 unique values, this means they contain one unique count. Hence we drop those columns.

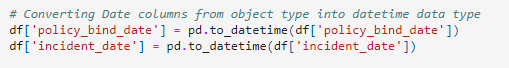


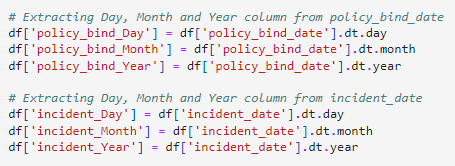
We could find >70% of the zero values in umbrella\_limit, we drop this column



Further we found insured\_zip column had 995 unique count whereas only 5 repeating values, hence we drop the insured\_zip column.

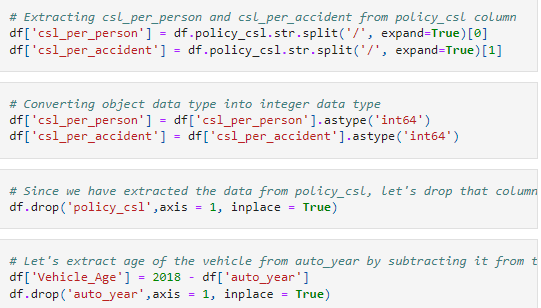


Further continuing with Feature Extraction. policy\_bind\_date and incident\_date have object datatype needs to be converted to in date-time format.Then we extract Day, Month, & Year from both columns.

Then we drop policy\_blind\_date, incident\_date columns.



From the features we find 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.

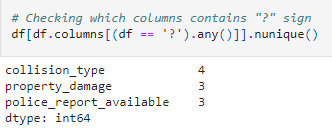


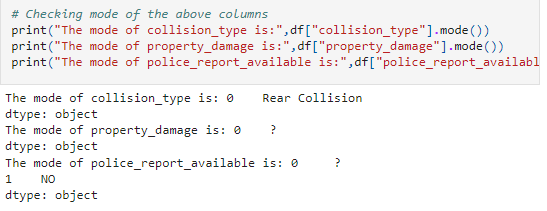
After dropping policy\_csl column, incident\_year column had one value throughout hence we drop this column too.



**Imputing of NULL values**

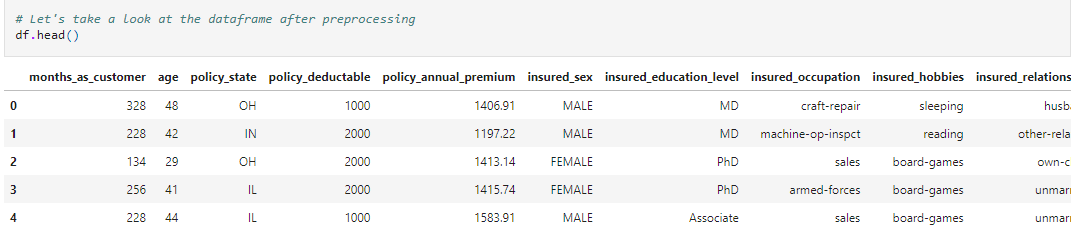
Imputation is process to fill null values in dataset using mean, median or mode, we had ‘?’ values in some columns which need to be filled.



As they have categorical data in it, we replaced the values with most frequently occurring values i.e mode values.

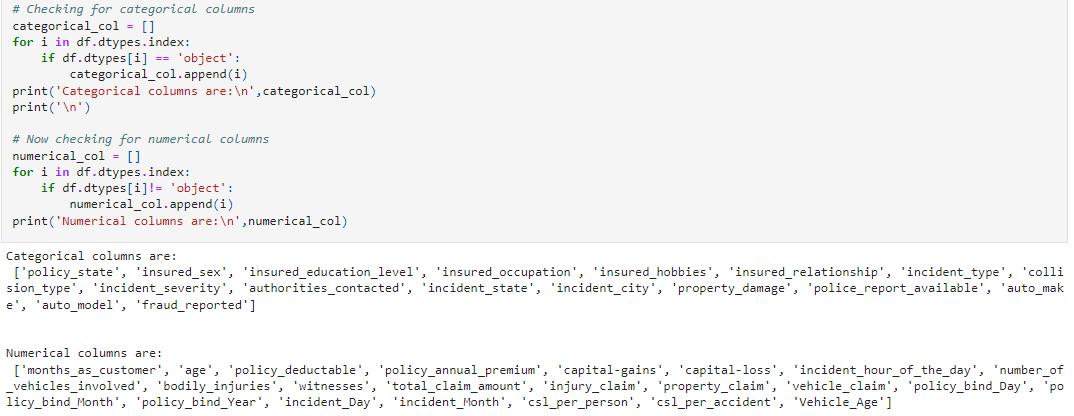
Mode of property\_damage & police\_report\_available is “?”, i.e most of the data is ?, we will fill these with 2nd highest count of respective columns.

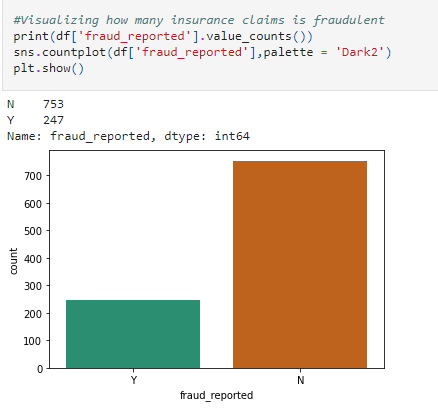
After data cleaning the dataset looks like this:



**Data Visualization**

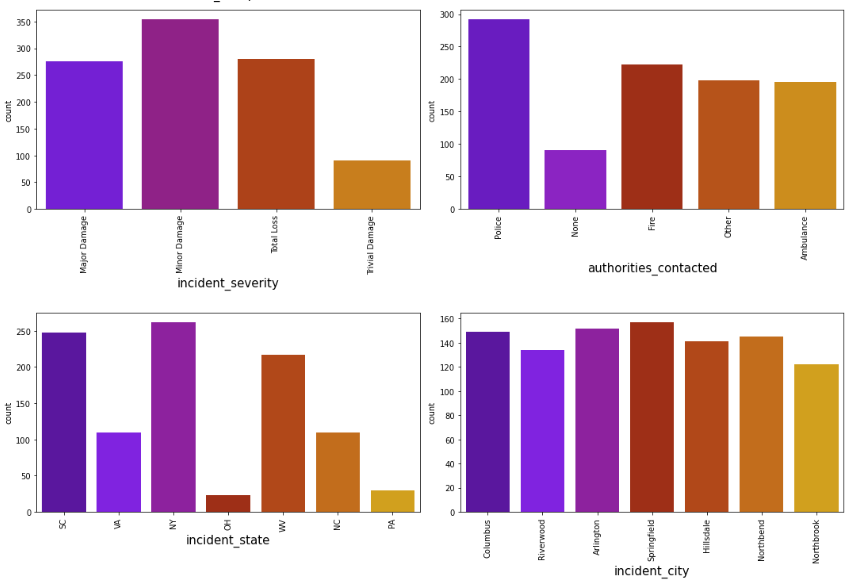
Checked the numerical and categorical columns to visualize the features vs target.

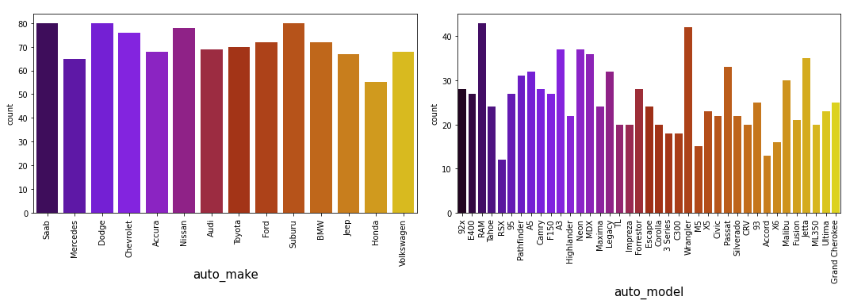




From above plot we get to know that maximum cases reported are not fraud. Since this is our target column we see that data is not balanced properly.. We will balance the data by oversampling method.

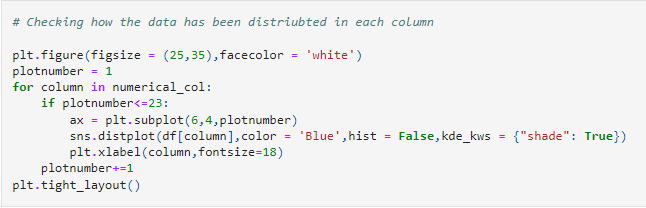


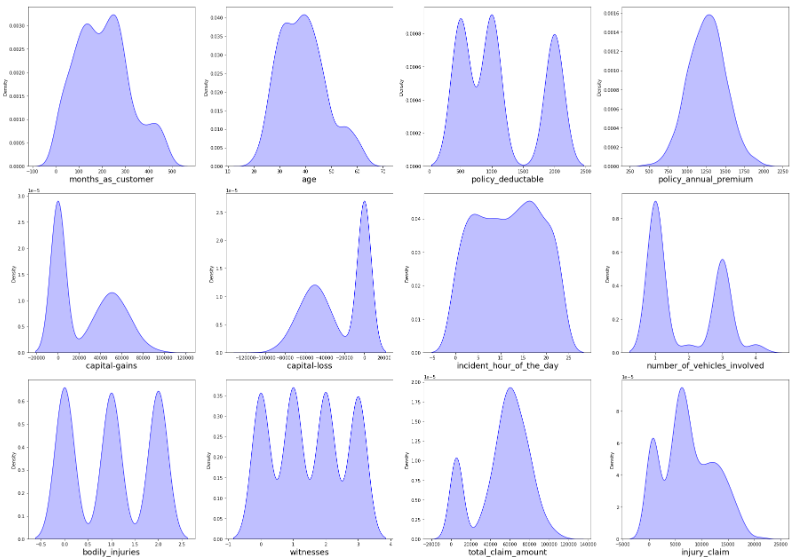


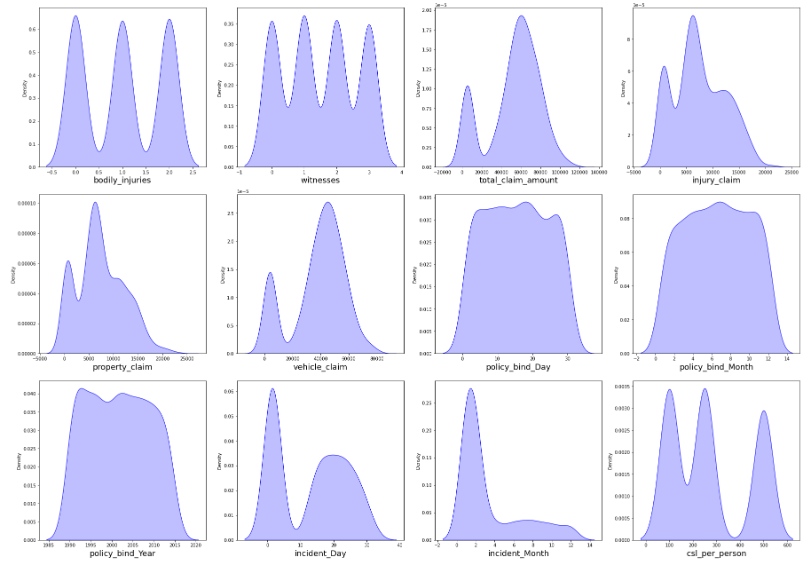


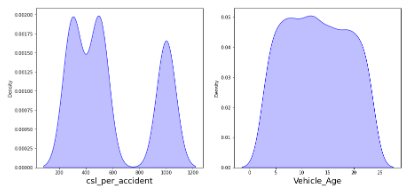
From count plots above we can observe :-

1. Machine operation inspector cover most of the data followed by professional speciality in the Insured occupation.
2. Insured hobbies are covered by reading the most, followed by exercise while others have average data count.
3. Minor damages count highest in incident severity whereas trivial damage is lowest.
4. Authorities contacted for accident occurrence are Police followed by Fire. Authorities in ambulance and Others are contacted the same count and none are contacted very less compared to all.
5. In incident city all columns have almost equal count.
6. Companies like Saab, Subaru, Dodge, Nissan & Volkswagen have highest count in the vehicle manufacturing companies.
7. Vehicle models like RAM, Wrangler have highest counts whereas RSX & Accord have lowest count.





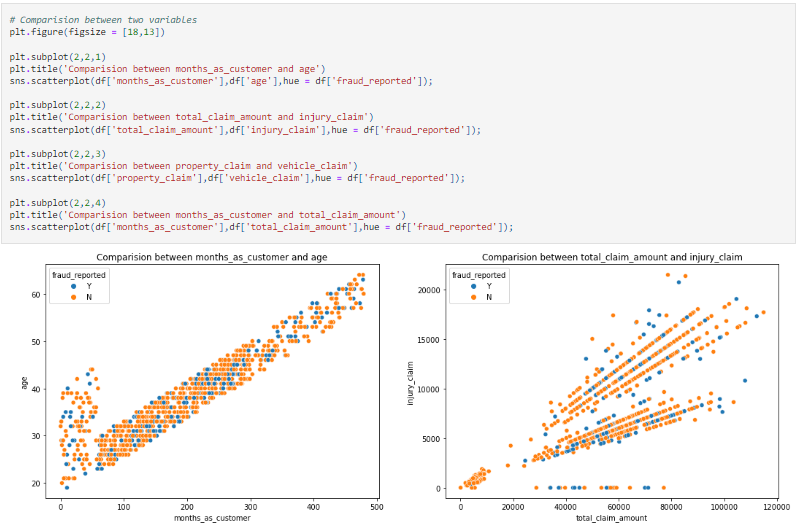


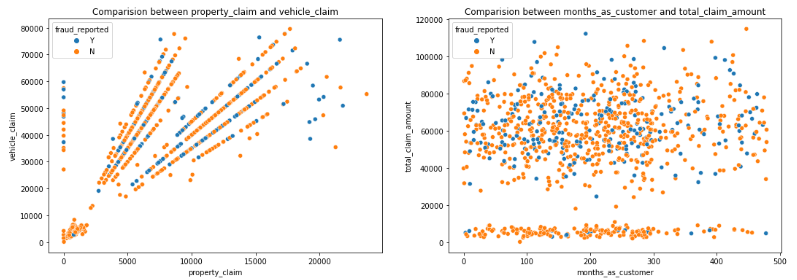


We can see that data is normally distributed in most columns. Some columns like capital gains and incident months have mean > median, hence are skewed to right.

The data in capital loss is skewed to left since the median > mean. We will remove the skewness further.

**Bi-variate Analysis**



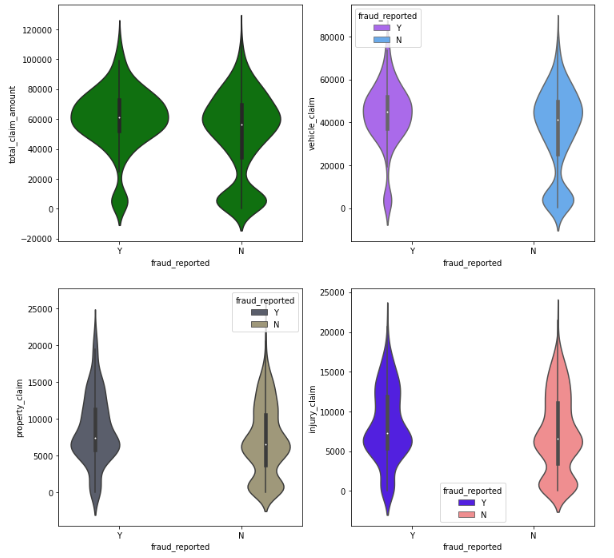
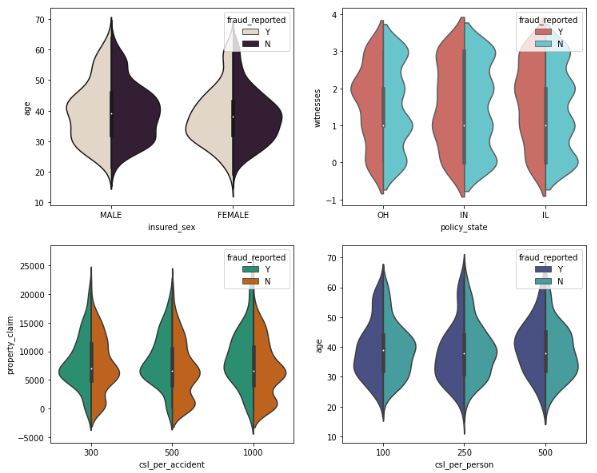
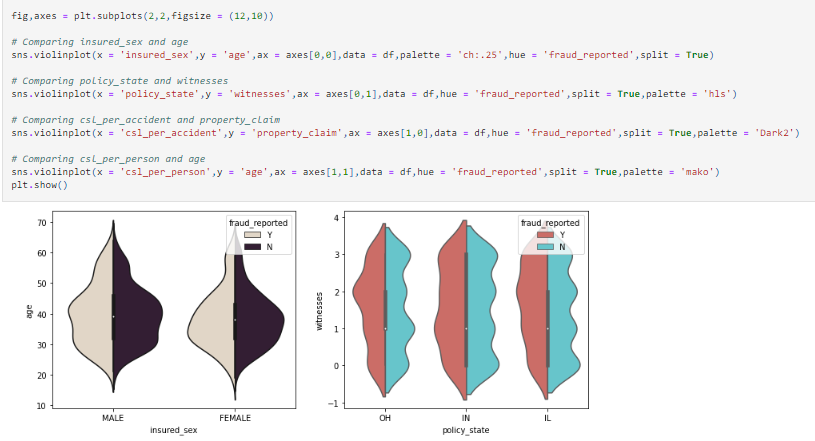


We see positive linear relation in age and month\_as\_customer. Increases in the month\_as\_customers also increases the age, also the fraud reported is very low in this.

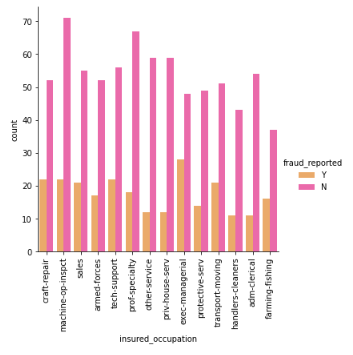
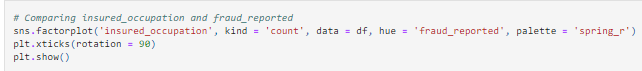
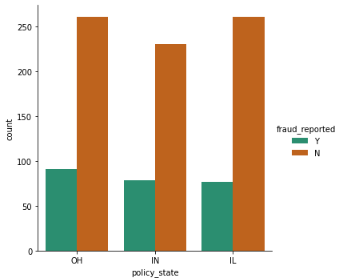
In next graph we observe a positive linear relation in total claim amount & injury claim.

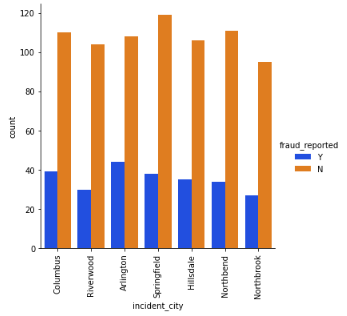
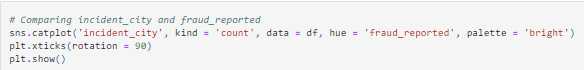
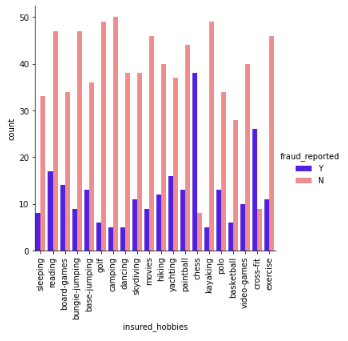
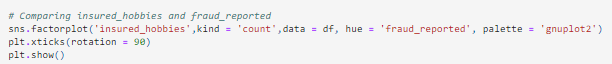
Third plot resembles second plot, the property claim increases as the vehicle claim increases.

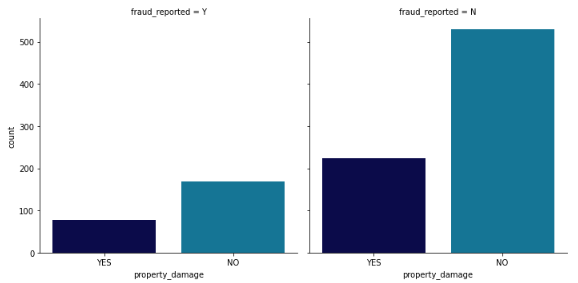
Fourth plot does not have much relation between the features.



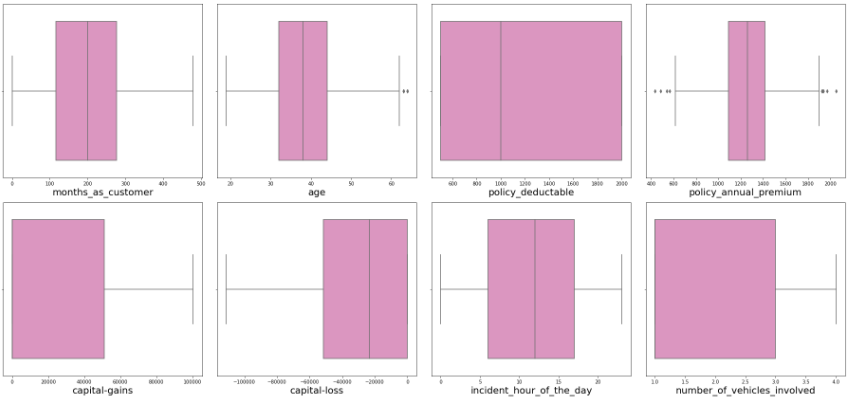
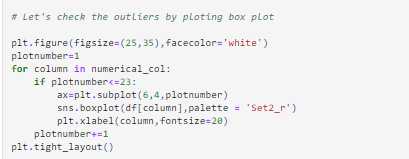
Data Visualization is a technique where comparison and plotting the data becomes self explanatory, as seen above. Lets move with some more visualization plots.

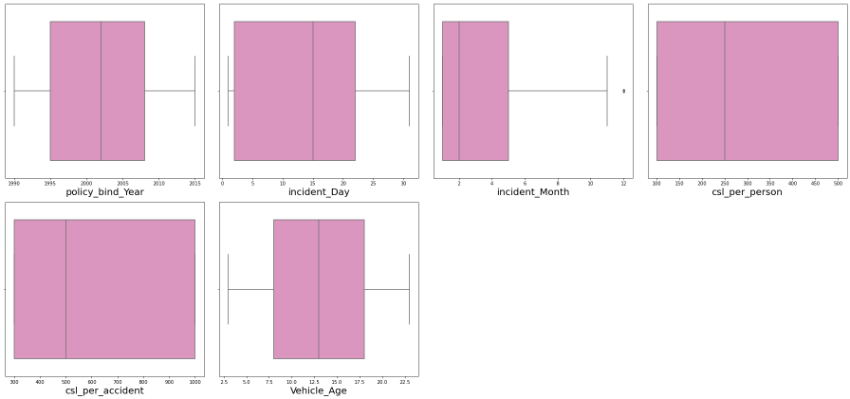
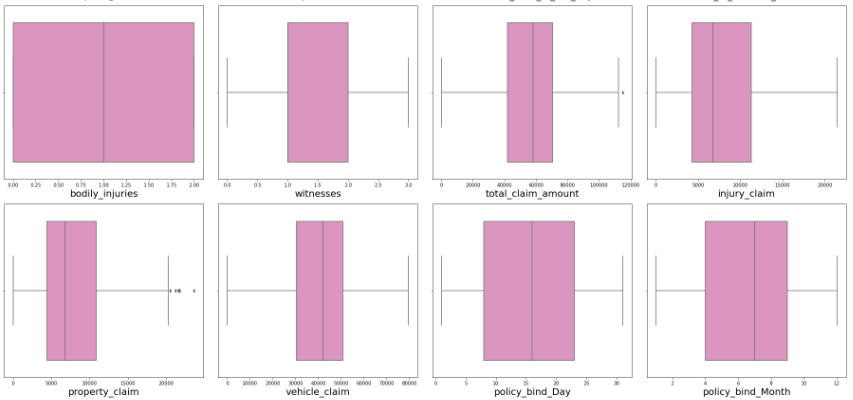




After data visualization EDA part we have looked into various aspect of the dataset, like checking of null values and imputing it, extracting date time, the value counts and doing the feature extraction. Now we will perform another analysis where we check outliers and remove them. We will also look for the skewness of the dataset and remove those..

Identifying Outliers



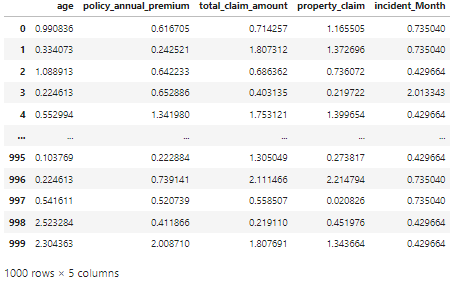


From above box plot we identify the outliers and we find the outliers in the columns below:

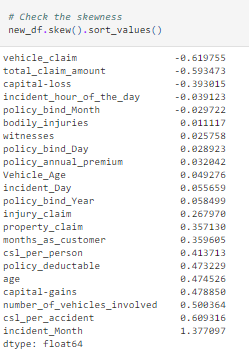
Age, policy\_annual\_premium, total\_claim\_amount, property\_claim & incident\_month.

These numerical columns contains outliers we will remove the outliers in these columns using Z score method.

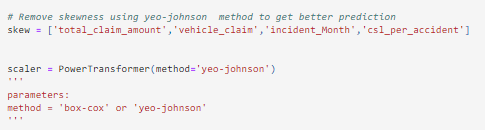


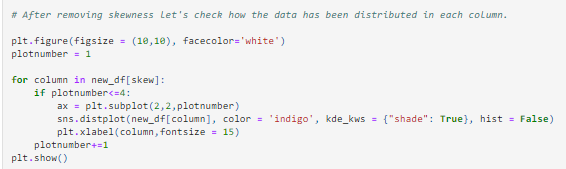


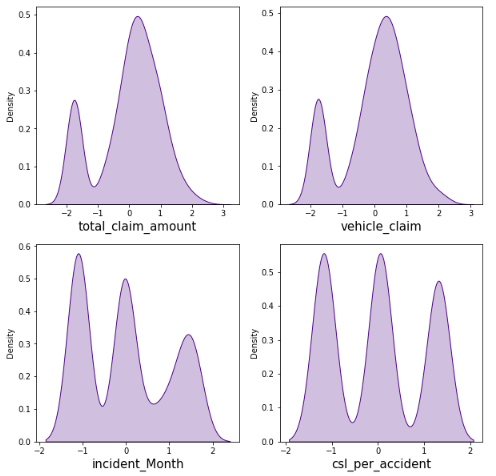
 Here we have removed the outliers whose Z score is less than 3.Now we work on Skewness of dataset.



We observe some skewness in dataset, hence we use YEO-JOHNSON method for skewness removal.



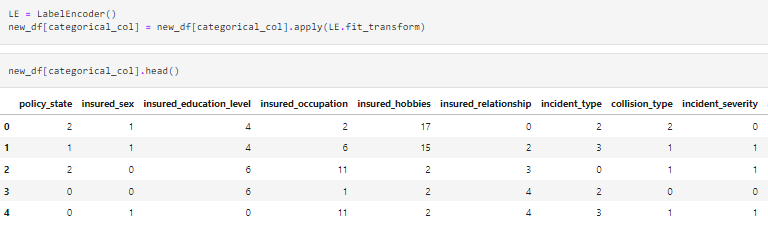
After removing skewness in dataset we check the data distribution.

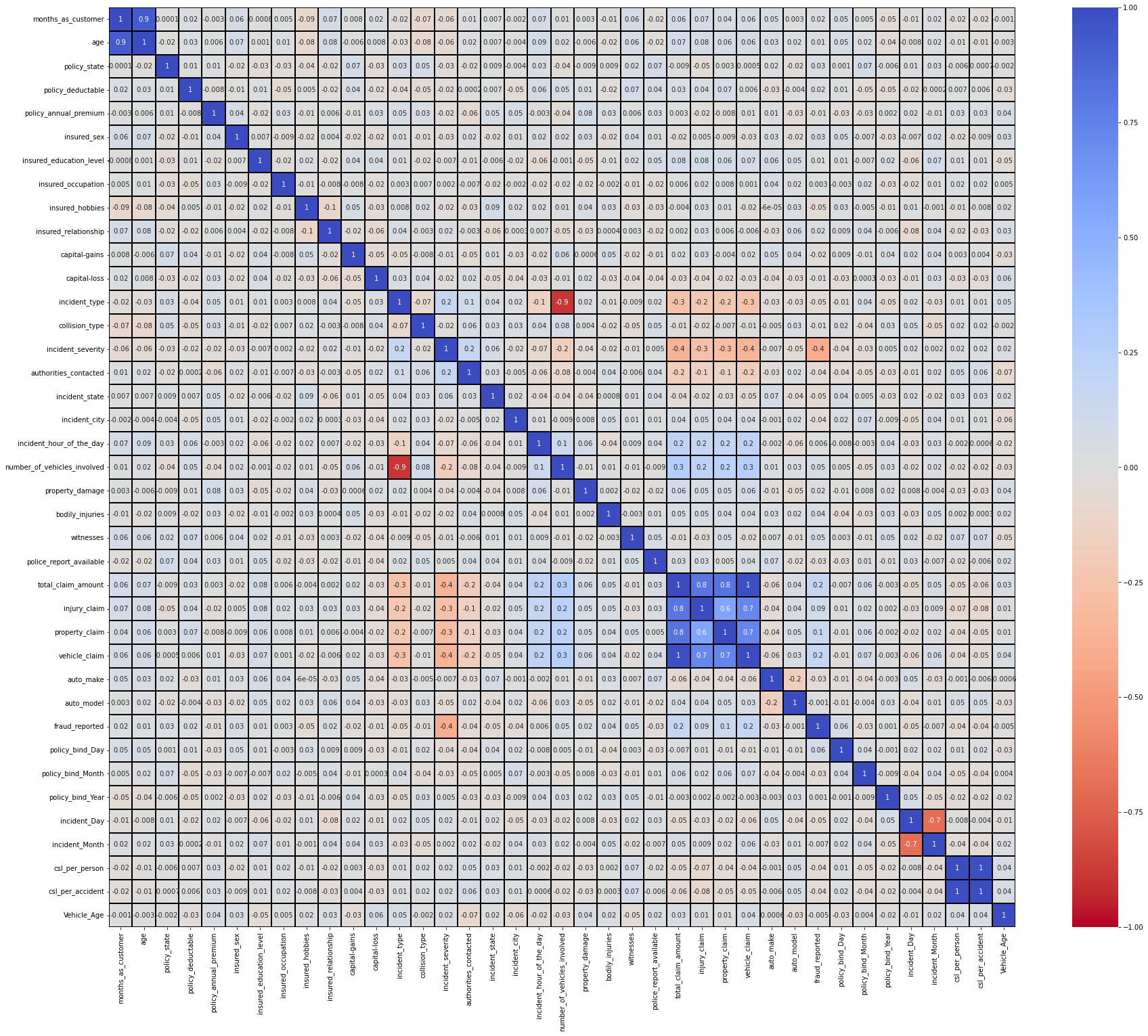


After removal of outliers and skewness we are ready with well distributed data available for preparing a model.

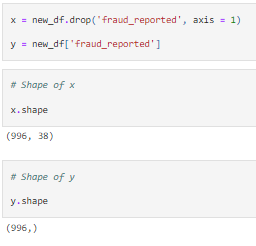
We can seen that dataset has both numerical and categorical data. Models understand numerical data only, hence we will encode the data. Also there can be some multi- colinearity which we will see through a heat-map and also remove it. We have also seen that target variable is imbalanced, which will be fixed by oversampling method. Finally we will scale the data to be ready to be trained and tested.

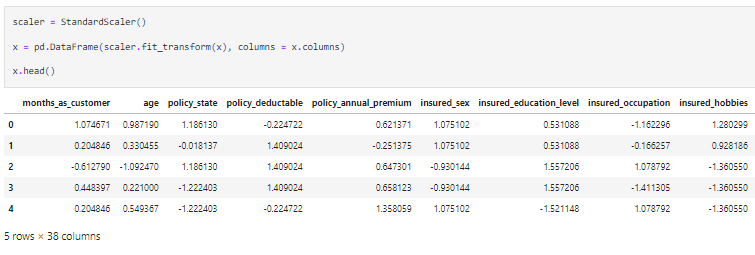
**Encode the categorical columns using Label Encoding**



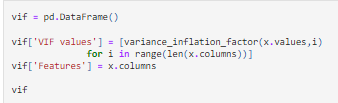
After Encoding, the dataset looks as above.Now we check the co-relation between target variable and other features using heat-map. This heat-map shows the correlation matrix. This heat map contains both positive and negative correlation.

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 leads to the multi colinearity problem. We check the VIF value to overcome this multi colinearity problem.

**Separating features and label variables into x and y** 

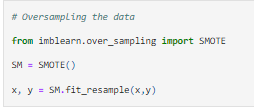
**Scaling the dataset using Standard Scalar** 

**Checking Multi Colinearity by VIF method.**

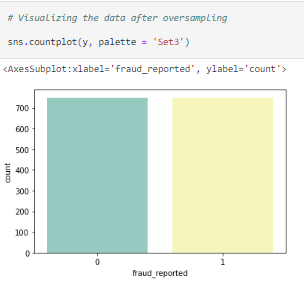


Some columns have VIF > 10 which mean they are causing multi colinearity problem. Let's drop the features having VIF > 10 among all the columns. These are total\_claim\_amount and csl\_per\_accident features, and we have dropped it.

Earlier we identified another problem of imbalance data in the target variable, we use oversampling method to treat it.

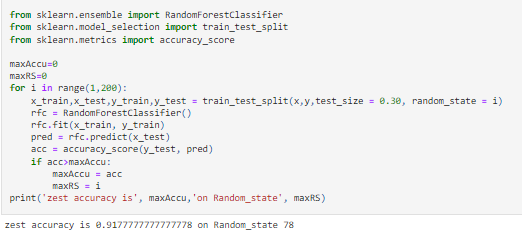


After treating the data we visualize the data



Now we see the data is balanced properly.

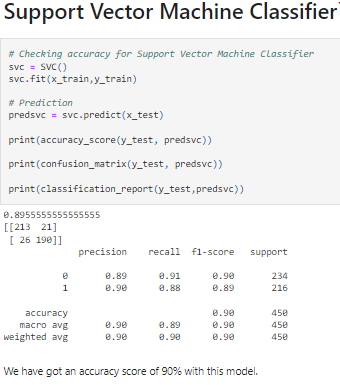
We will start modeling now by searching the Random state. Random state ensures the splits that you generate are reproducible. Scikit-learn use random permutations to generate the splits. The random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order.



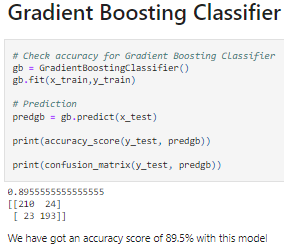
Here we have used RandomForestClassifier to find the best random state, as we have got an accuracy score of 92% at the random state of 78. Let’s build a model on above data.

Lets split the dataset into train and test using train\_test\_split. 

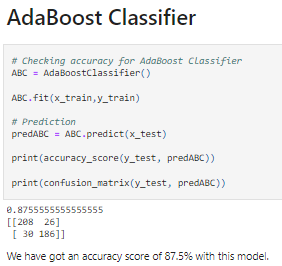
This model was created by using RandomForestClassifier , but just one model will not give us confidence on the data, hence we will build more models and check the accuracy.



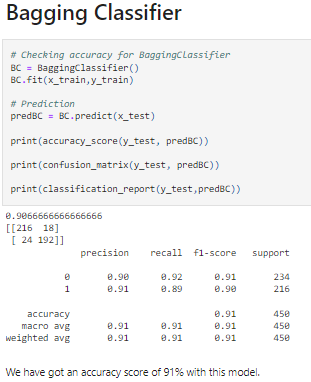
We have an accuracy score of 90% with SVC model.



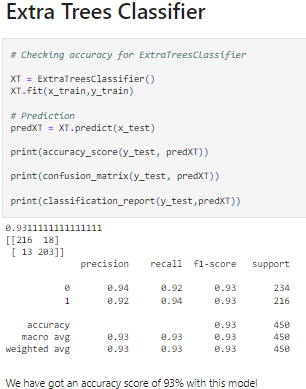
In Gradient Boosting Classifier we got accuracy score of 89.5%.



In ADA Booster Classifier we have accuracy score of 87.5% which is lower than Gradient Boosting Classifier.

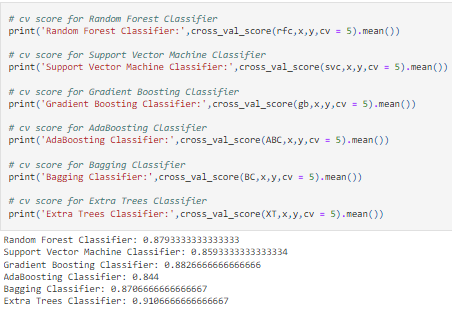


Bagging Classifier gave an accuracy score better than Gradient boosting classifier and ADA boosting classifier. Accuracy score is 91%.

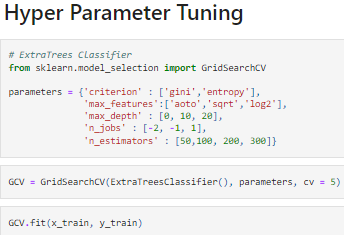


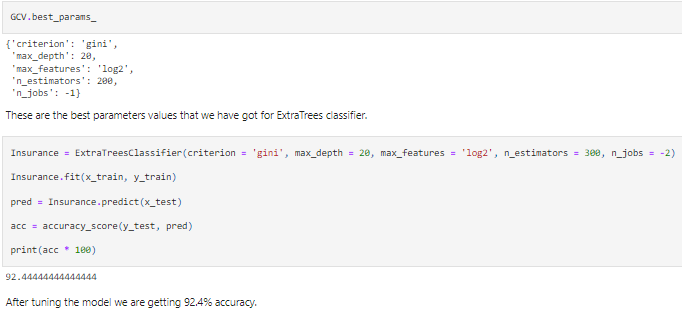
We have best accuracy score of 93.11% in Extra Trees Classifier.

Now lets check Cross\_Validation score of all the classifiers above.



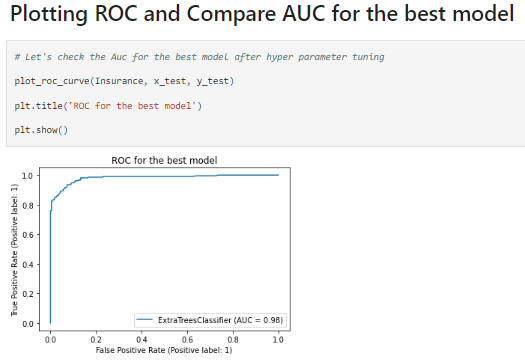
We get higher CV score of Extra Trees Classifier, hence we will use Extra Trees Classifier and Hyper Tune its parameters.





After getting best parameters We built a model and got model accuracy of 92.4% from Extra Trees Classifier.

**Plotting the AUCROC Curve**



**Saving the best model prepared.**



**Predicting the saved model.**



**AND WE CONCLUDE THE EXTENSIVE ANALYSIS**

In the beginning we discussed about the life cycle of Machine Learning Model, then you can see how we had touched each points and finally reached model building and prepared the model for deployment.

In every model building problem Data Analysis and Feature Engineering is the most crucial part.

We had handled numerical and categorical data and also built different machine learning models on the same dataset.

By hyper parameter tuning we can improve our model accuracy, for instance in this model the accuracy remained nearly same.

By using this Machine Learning Model we can easily predict if insurance claim is fraudulent or not and reject those applications which are considered as fraud claims.

**Thank you for your Patience**

**Submitted By**

**Dinesh Mutha**