Insurance Fraud Detection with Machine Learning Models

**Problem Definition:**

In this project, you are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made. In this example, you will be working with some auto insurance data to demonstrate how you can create a predictive model that predicts if an insurance claim is fraudulent or not.

# Introduction

Insurance fraud is defined as a deception which is deliberately committed against/by a claimant/ an insurance company or agent with the motive of financial gain. At any step of a transaction or application the fraud may be committed. Inflating claims, distorting and/or misrepresenting facts on an insurance application, submitting claims for damages that may have never occurred as well as staging accidents are some of the most commonly occurring frauds.

Fraudulent Insurance claims have been a huge problem in the industry for well over a century. Over the due course of time it has been getting progressively difficult to identify fraud claims. According to the Federal Bureau of Investigation, over $40 Billion every year is the cost of insurance fraud transactions.

Machine Learning is a boon in helping the Auto Insurance industry with this problem.

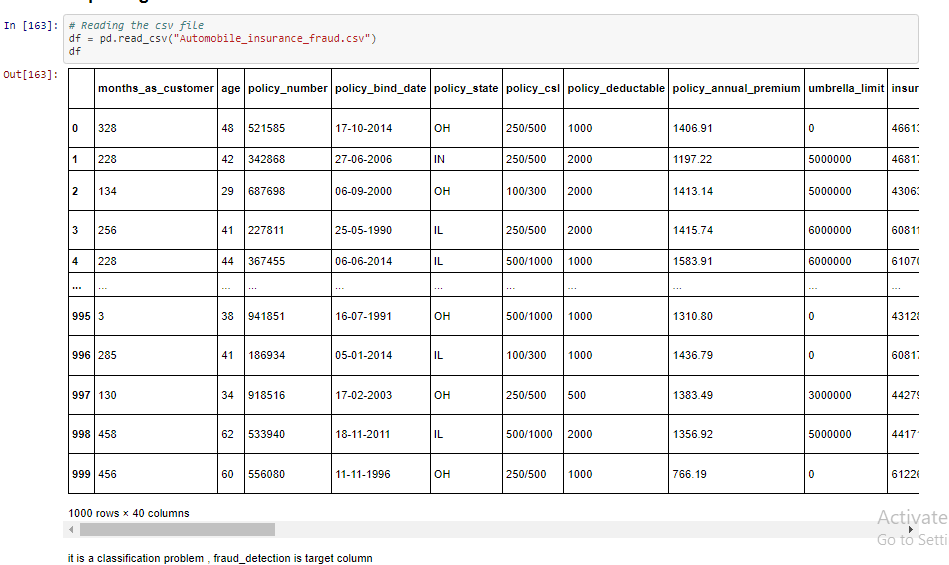
**Executive Summary:**

In this project, a dataset was provided with the details of the insurance policy along with the customer details, as well as details of the accident on the basis of which the claims have been made.

The Dataset was first cleaned, the various feature columns were analysed, then with feature engineering and based on strength of correlation and ANOVA f-score values, the feature columns were selected that would best predict the Target variable, to train and test machine learning models.

The auto insurance dataset was worked with to build a predictive model that best predicts if an insurance claim is fraudulent or not. Several models were trained and fitted with a part of the dataset and then tested with a different part of the dataset. The model that performed the best with the best confusion matrix performance, classifiers models, ROC-AUC score and cross validation performance was then selected and tuned further with hyper parameter tuning techniques.

**About the Dataset:**

****

The given dataset consists of 40 columns and 1000 rows.

**The Independent Feature columns are:**

**months\_as\_customer: Number of months for which the person has been a customer**

**age: Age of Customer**

**policy\_number: Identification number of policy**

**policy\_bind\_date: Time period between effective date of coverage and policy issuance.**

**policy\_state: State where policy is active**

**policy\_csl: Policy Combined single limit**

**policy\_deductable: Amount paid before the insurance company starts paying up.**

**Policy\_annual\_premium: The total amount of premium paid annually**

**Umbrella\_limit: Provides excess limits and gives additional excess coverage**

**Insured\_zip: Zip Code of the Insured address**

**insured\_sex : Gender**

**Insured\_education\_level: Education Background of Insured**

**Insured\_occupation: Occupation of Insured**

**Insured\_hobbies: Hobbies of the Insured**

**Insured\_relationship: Relationship of the Insured**

**Capital-gains: Capital Gains made from insurance**

**Capital-loss: Capital Loss incurred**

**Incident\_date: Date on which Incident Occured**

**incident\_type: Type of Incident**

**Collision\_type: Type of collision**

**incident\_severity: Severity of Incident**

**Authorities\_contacted: Whether authorities were contacted**

**Incident\_state: State where incident occurred**

**incident\_city: City where incident occurred**

**incident\_location: Location of incident**

**Incident\_hour\_of\_the\_day: Time of the day when incident occurred**

**number\_of\_vehicles\_involved: Number of vehicles involved in incident.**

**property\_damage: Whether there was property damage or not**

**Bodily\_injuries: Severity of bodily injuries**

**witnesses: Number of Witnesses**

**Police\_report\_available: Whether police reports are available**

**Total\_claim\_amount: Total amount of claim**

**Injury\_claim: Injury Claim amount**

**Property\_claim: Property Claim amount**

**vehicle\_claim: Vehicle Claim amount**

**Auto\_make: Make of Vehicle**

**Auto\_model: Model of Vehicle**

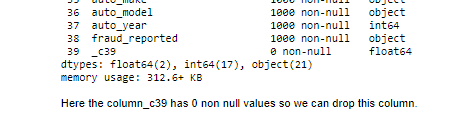
**Auto\_year: Year of Vehicle Manufacture**

**The Target Variable to predict is given in the column:**

**Fraud\_reported: Whether fraud was reported as Yes or No**

**Data Cleaning:**

Upon inspecting all the columns in the dataframe, it is observed that column \_c39 has 0 non Null values present.

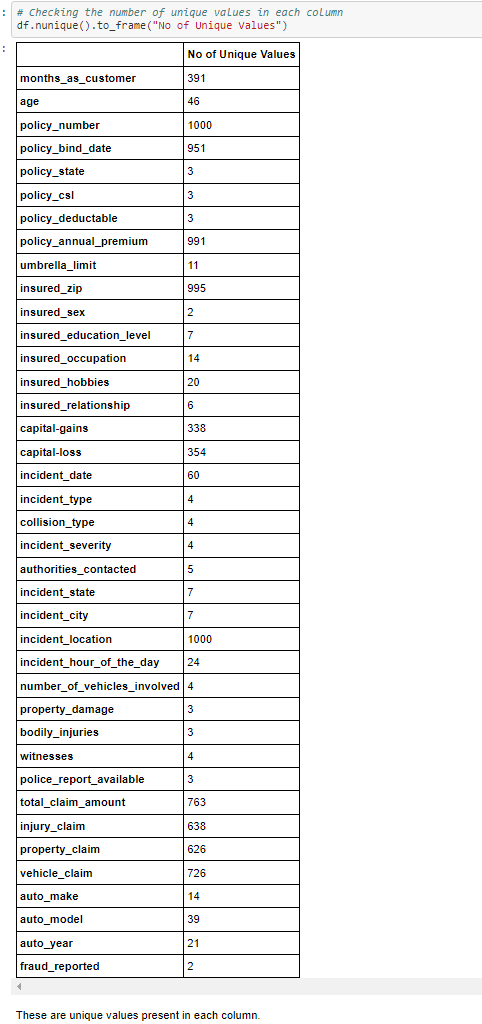


\_c39 has no usable data present. Other columns appear to have no null values. Therefore it will be dropped.

****

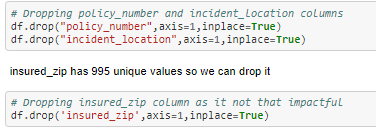
**Check for unique values**

policy\_number and incident\_location have 1000 unique values.

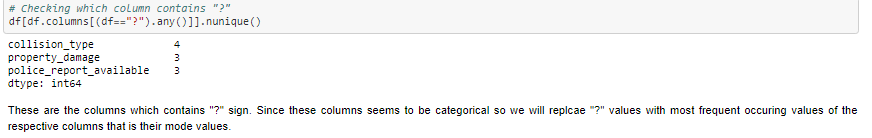
****

## Data Preprocessing

policy\_number and incident\_location have 1000 unique it is not required for prediction and we can drop it.And insured\_zip has 995 values .So, we can drop that too.

****

Now, cheking for the value counts of the columns we can also observe that some columns have "?" values

****

Checking the mode of the columns: collision\_type, property\_damage, police\_report\_available

Checking value\_count of property\_damage column and police\_report\_available

Replacing "?" by their mode values

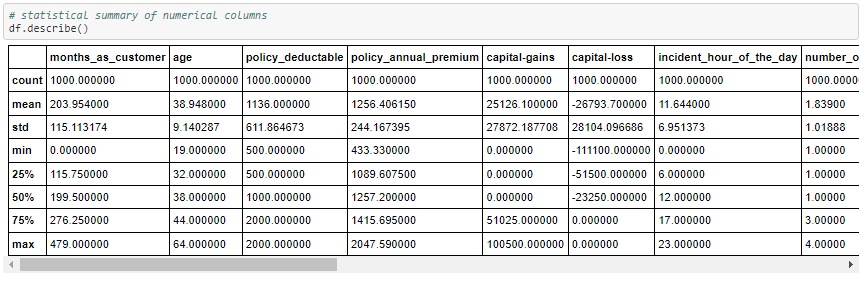
The policy\_csl column showing object data type but it contains numerical data, may be it is because of the presence of "/" in that column. So first we will extract two columns csl\_per\_person and csl\_per\_accident from policy\_csl colums and then will convert their object data type into integer data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

Extracting csl\_per\_person and csl\_per\_accident from policy\_csl column

Converting object datatype into integer data type

extract age of the vehicle from auto\_year by subtracting it from the year 2018

**Exploratory Data Analysis**

****

There is a huge difference in 75% and max it shows that huge outliers present in the columns.

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.

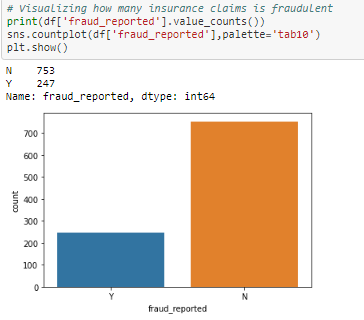
And in some of the columns like total\_claim\_amount, vehicle\_claim...etc we can observe the median is greater than the mean which means the data in the columns are skewed to left.

### This is a Classification Problem since the Target variable / Label column ("fraud\_reported") has Categorical type of Data.

**Univariate Analysis**

**Analyzing the Target Class**

#### There are 2 unique categorical values in the Label column / target variable, viz. ‘Y’ and ‘N’.



**Class**

**'N' : Has 75.30% of total values**

**'Y' : Has 24.70% of total values**

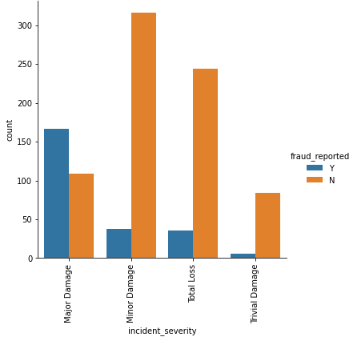
#### **Therefore, the Classes are imbalanced.**

#### 

**Feature Engineering**

**Upon analyzing the rest of the Feature Columns, following observations are made:**

* Majority of the cases are Multi-vehicle Collision and Single Vehicle Collision and are Rear Collisions.
* Most incidents take place between January and February.
* Minor Damage is most common followed by Major Damage and Total loss.
* Most common authorities contacted were the Police followed by Fire force.
* Majority reported no property damage.
* There are no police reports available for most cases.

**Bivariate Analysis**

### 

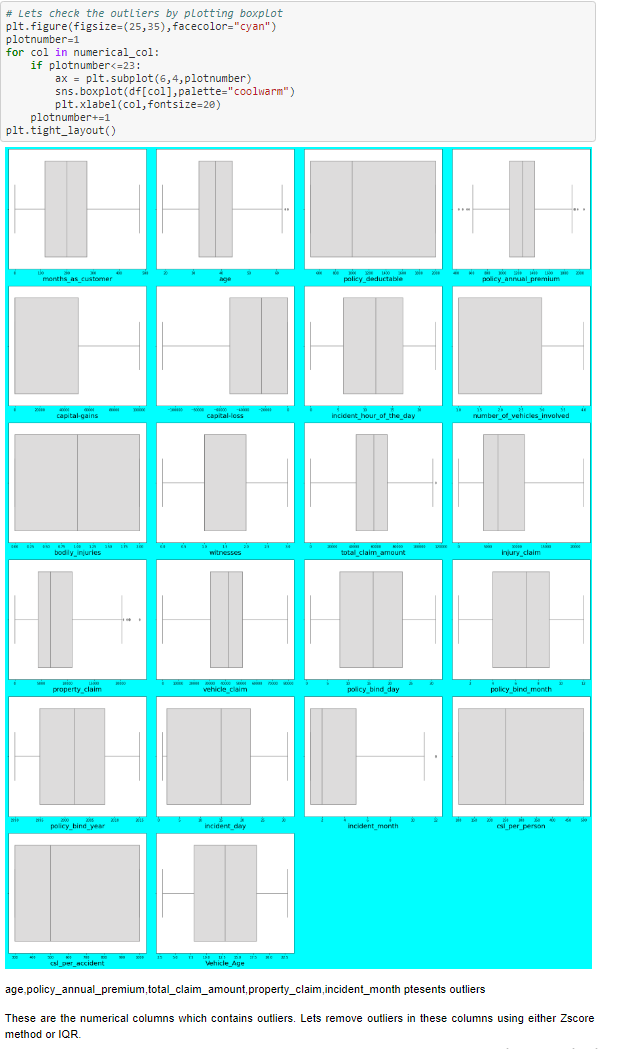
**Following observations can be made from above graphs:**

* Fraud report is high in "OH" policy\_state.
* The fraud reported very less for the people who have high school education and in "JD" education have high fruad report.
* The people who have high insured education are facing insurance fruadulent compared to the people with less insured education level.
* The people who are in the position exec-managerials 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.
* Higher the total claim amount, more the fraud claims are filed.
* Higher the injury claim amount, more the fraud claims are filed.
* Higher the property claim amount, more the fraud claims are filed.
* Higher the vehicle claim amount, more the fraud claims are filed.
* policy state,policy csl,insured sex,authorities contacted,bodily injuries,incident city, witnesses don't seem to contribute to fraud probability.
* Education levels of JD and High school and MD contribute most to the fraud claims filed.
* The fraud report is high for the customers who have other relative and it is less for unmarried people.
* Single vehicle collision and multi vehicle collision contribute most to the fraud claims filed.
* Incidents in states SC and NY contribute most to the fraud claims filed.
* fraud claims are more for 1 and 3 vehicles involved in accident
* fraud claims are more for rear collision in accident
* fraud claims are most for Major damage reported
* fraud claims are most for hours 10,14,16,18(office rush hours) and 23 of the day
* fraud claims are more when no property damage is reported
* fraud claims are more when no police report is available

# Identifying the Outliers

age,policy\_annual\_premium,total\_claim\_amount,property\_claim,incident\_month presents outliers which identifies and needs to be removed

These are the numerical columns which contains outliers. Lets remove outliers in these columns using either Zscore method or IQR.

****

**Z-score —** Call scipy.stats.zscore() with the given data-frame as its argument to get a numpy array containing the z-score of each value in a dataframe. Call numpy.abs() with the previous result to convert each element in the dataframe to its absolute value. Use the syntax (array < 3).all(axis=1) with array as the previous result to create a boolean array.

**Interquartile range —** The interquartile range can be used to detect the outliers present in the dataframe.

Calculate the interquartile range for the data by using scipy.stats.iqr module.

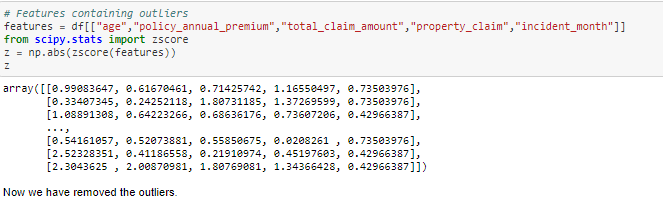
Multiply the interquartile range by 1.5.

Add 1.5 x interquartile range to the third quartile. Any number greater than this is a suspected outlier.

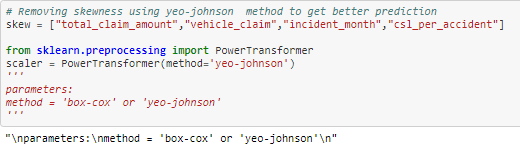
Subtract 1.5 x interquartile range from the first quartile. Any number lesser than this is a suspected outlier.

**Removing the outliers**

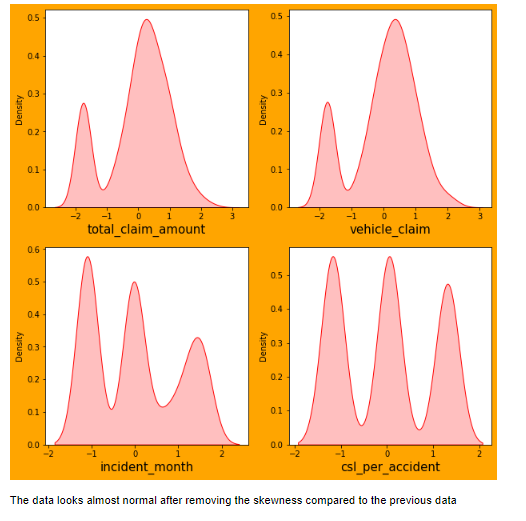
The outliers are removed by zscore method



## Checking and removing skewness using yeo-johnson method

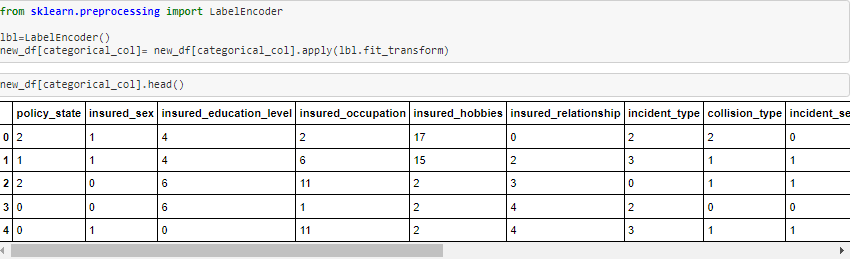
****

After removing skewness let's check how the data has been distributed in each column.

****

## Encoding the categorical data

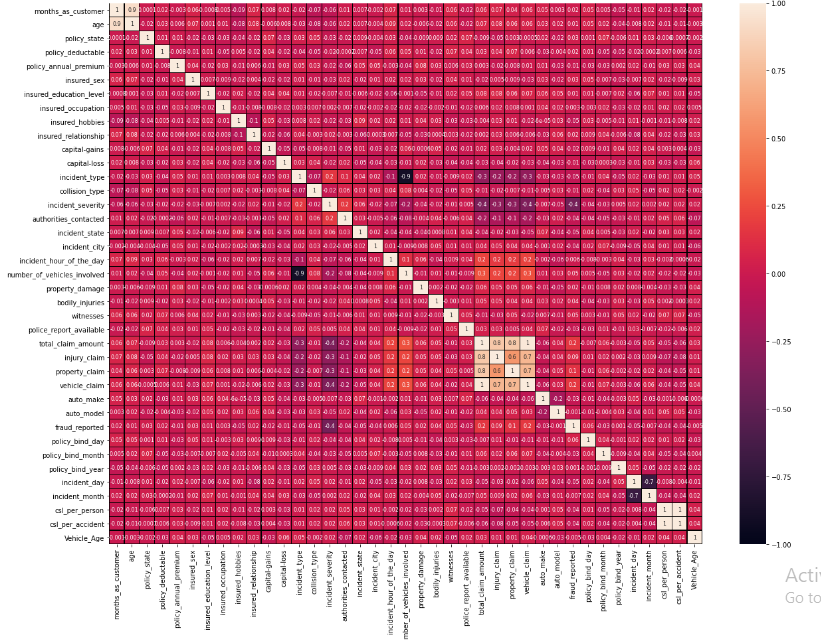
Encoding the categorical data by importing LabelEncoder from sklearn.preprocessing.

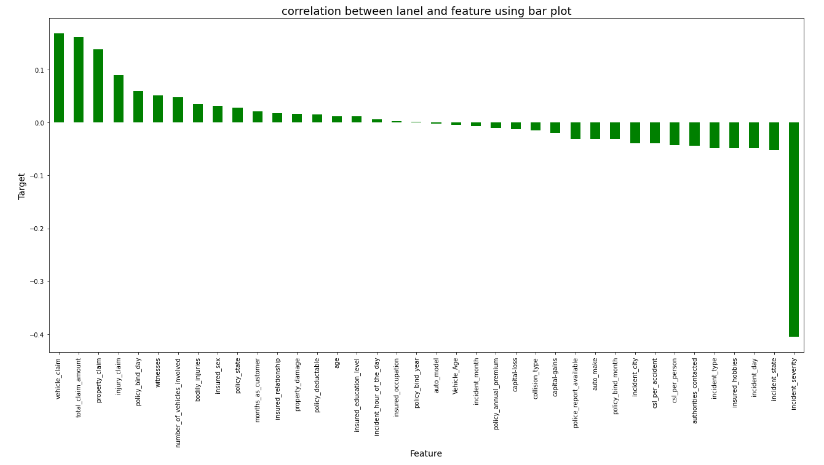
****

**Correlation**

We can see the correlation between dependent and independent variables after encoding the categorical data.

Visualizing the correlation matrix by plotting heat map.

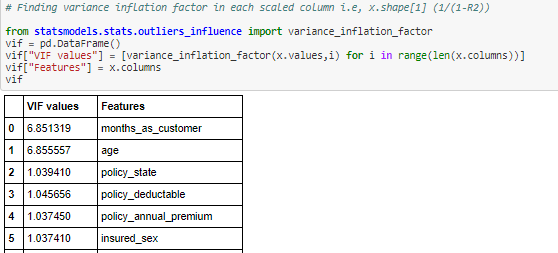
****

****

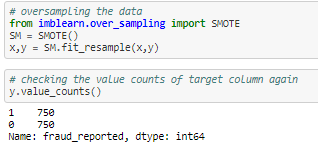
### From the bar plot we can observe that the columns policy\_bind\_year,insured\_occupation and auto\_model age are very less correlated with the target. We can drop these columns if necessary.

### Checking for Multicollinearity using Variance Inflation Factor

Variance inflation factor measures how much the variance of an independent variable is influenced / inflated, by its interaction/correlation with other independent variables.

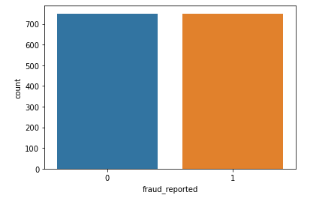
****

## Balancing of data and Oversampling:



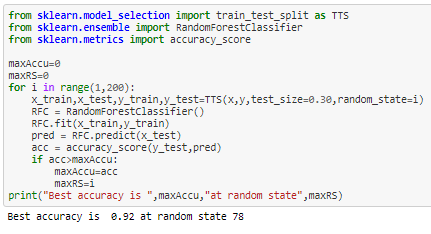
## We can see that the data is not balanced. Lets use oversampling method to balance the data.

Visualizing the data after oversampling



**Classification Model Building**

**Finding the Best Random State**



****

### Training the Models

RFC = RandomForestClassifier()

svc = support vector machine classifier SVC()

GB = GradientBoostingClassifier()

ABC = AdaBoostClassifier()

BC = BaggingClassifier()

ET = ExtraTreesClassifier()

XGB = xgb(verbosity=0)

**Analyzing Model Accuracies**

### Logistic Regression Model Accuracy

The trained Logistic Regression Model shows

F1 score of 0.83

Roc\_auc score of 0.8329

Cross validation score of 0.8320

Sensitivity(Recall for ‘ Fraud’ cases) is 0.86 and Specificity (recall of non-fraud cases) is 0.81

Precision for ‘ Fraud’ cases is 0.82 and Precision for non-fraud cases is 0.85

### Random Forest Classifier Model Accuracy

The trained Random Forest Classifier Model shows

F1 score of 0.91

Roc\_auc score of 0.9057

Cross validation score of 0.86

Sensitivity(Recall for ‘ Fraud’ cases) is 0.92

Precision for ‘ Fraud’ cases is 0.89 and Precision for non-fraud cases is 0.93

## Support Vector Machine Classifier

F1 score of 0.89

Roc\_auc score of 0.89

Cross validation score of 0.84

Sensitivity(Recall for ‘ Fraud’ cases) is 0.89

Precision for ‘ Fraud’ cases is 0.89 and Precision for non-fraud cases is 0.90

## GradientBoostingClassifier

F1 score of 0.89

Roc\_auc score of 0.89

Cross validation score of 0.8933

Sensitivity(Recall for ‘ Fraud’ cases) is 0.89

Precision for ‘ Fraud’ cases is 0.89 and Precision for non-fraud cases is 0.90

### AdaBoost Classifier Model Accuracy

The trained AdaBoost Classifier Model shows

F1 score of 0.85

Roc\_auc score of 0.8511

Cross validation score of 0.8500

Sensitivity(Recall for ‘ Fraud’ cases) is 0.84 and Specificity (Recall of non-fraud cases) is 0.85

Precision for ‘ Fraud’ cases is 0.85 and Precision for non-fraud cases is 0.87

## BaggingClassifier

F1 score of 0.87

Roc\_auc score of 0.8688

Cross validation score of 0.87

Sensitivity(Recall for ‘ Fraud’ cases) is 0.87 and Specificity (Recall of non-fraud cases) is 0.85

Precision for ‘ Fraud’ cases is 0.85 and Precision for non-fraud cases is 0.87

## ExtraTreesClassifier

F1 score of 0.92

Roc\_auc score of 0.89

Cross validation score of 0.9177

Sensitivity(Recall for ‘ Fraud’ cases) is 0.94 and Specificity (Recall of non-fraud cases) is 0.85

Precision for ‘ Fraud’ cases is 0.89 and Precision for non-fraud cases is 0.87

**XGB Classifier Model Accuracy**

The trained XGB Classifier Model shows

F1 score of 0.91

Roc\_auc score of 0.9022

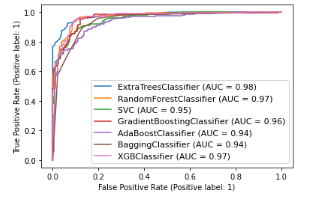
Cross validation score of 0.92

Sensitivity(Recall for ‘ Fraud’ cases) is 0.92 and Specificity (recall of non-fraud cases) is 0.88

Precision for ‘ Fraud’ cases is 0.91and Precision for non-fraud cases is 0.94

### 

### ROC AUC curves



Random Forest Classifier and XGB Classifier have the largest area under their respective curves both scored 0.97 on AUC

Since , Sensitivity summarizes the true positive rate, ie. how many we got correct out of all the positive cases and Specificity summarizes our true negative rate, which is how many we got correct out of all the negative cases. The model that performs the best in those criteria will be chosen.

Both Random Forest Classifier and XGB Classifier have performed the best on fraud and non fraud detections based on the fact that their Sensitivity, Specificity and Precision scores are the highest amongst all the model performances.

### Therefore, Based on the above graph and roc\_auc\_scores,XGB Classifier is the best model for the dataset, with AUC = 0.97 and roc\_auc\_score = 0.98[¶](http://localhost:8888/notebooks/InsuranceFraud_proj.ipynb#Based-on-the-above-graph-and-roc_auc_scores,XGB-Classifier-is-the-best-model-for-the-dataset,-with-AUC-=-0.97-and-roc_auc_score-=-0.9121)

### Hyper Parameter Tuning

GridSearchCV was used for Hyper Parameter Tuning of the ExtraTreeClassifier.

Based on the input parameter values and after fitting the train datasets,

The ExtraTreeClassifier was further tuned based on the parameter values yielded from GridsearchCV.

The Tuned ExtraTreeClassifier Model displayed accuracy of 92.43%

F1 score of 0.91

Sensitivity(Recall for ‘ Fraud’ cases) is 0.91 and Specificity (recall of non-fraud cases) is 0.92

Precision for ‘ Fraud’ cases is 0.91 and Precision for non-fraud cases is 0.92

### Concluding Remarks

In conclusion ExtraTreeClassifier Model is able to correctly distinguish between Fraud claims and legitimate claims with high accuracy.

The dataset had very limited data which is problematic as models show greater stability when the dataset is of a good size. However, a large set of feature columns enabled selecting a smaller feature size that provides the best accuracy for the model and obtaining results in optimal time.