\*\*Blog\*\*

**Insurance Claims**

**Fraud Detection :-**

****

Submitted BY:-

**Diptiranjan Pradhan**

*Batch no:- 1840*

******

***Introduction:-***

Insurance fraud is an illegal act by either the buyer or seller of an insurance contract. It occurs when a claimant attempts to obtain some benefit or advantage they are not entitled to, or when an insurer knowingly denies some benefit that is due.

As we know there are many types of fraud obtain now a days. Among all here we are going to discuss about Automobile insurance fraud .Coz it’s a major fraud among all.

🡪 But now the question is how to predict whether the insurance climbed is fraudulent or not. So here I am build a Machine learning model which can predict the claim is fraudulent or not.

* Here we have a dataset which contains many information.We have to predict the claim is fraudulent or not from this dataset.

**Problem statement :-**

**Business case:**  
 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.

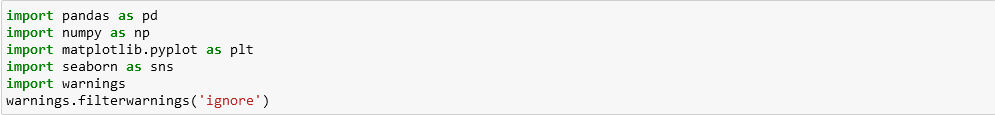
In this project, we are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.

In this example, we 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.

In this perticular problem we have to look into the insured person details and incident details and analyse the samples to know wheather the claim is fraudulent or not.

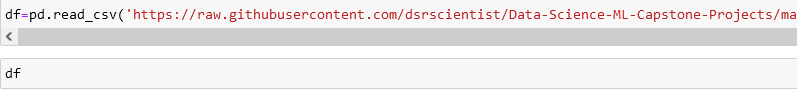
* Import required libraries:-

1st I have to import all the required libraries which are needed.

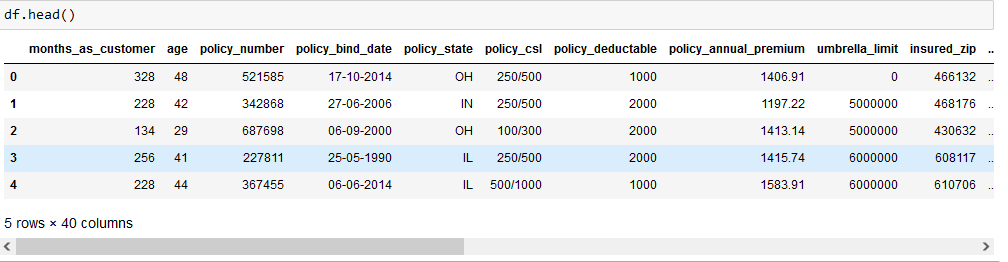


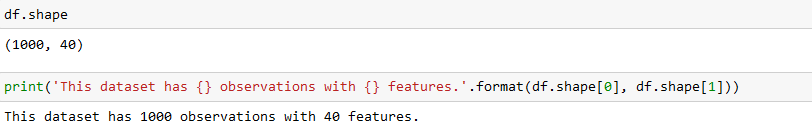
* **Importing the Dataset:-**

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

****

Here we import the dataset.



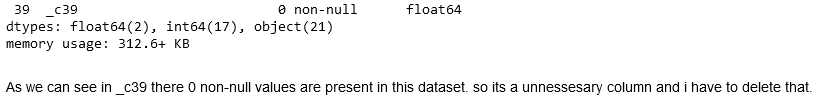
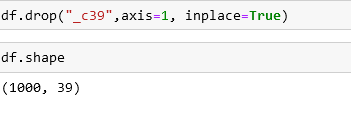


**Data Analysis:-**

. As we can see I have about 40 features in the dataset.

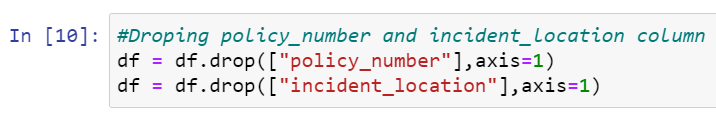
Now looking into the target ‘fraud\_reported’ and I have to make sure the data type of target column to decide the type of problem.

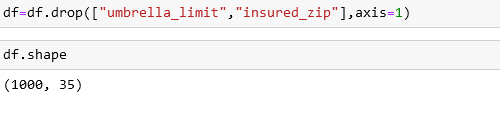
- By looking into the features I can say that I have both numerical and categorical columns with some unnecessary entries. So now we have to clean this data.

* **Data Preparation and cleaning :-**
* **1st** we have to do some statistical analysis like checking shape, nunique, value\_counts, info etc.
* After reading the value counts if there is any unnecessary columns in the dataset ,we have to drop those columns.
* ****
* After reading this I found \_c39 column whose entries are all NaN.So we have to drop this column. 

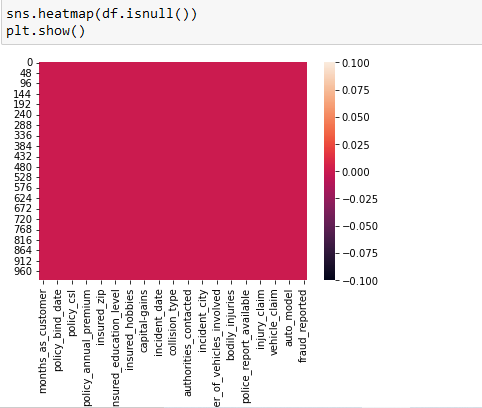
As we can see we have 39 columns after doping that column.

* After seeing the value counts of each column policy\_number and incident\_location has 1 element with 1000 value counts which means all the values are unique. These features will not help us in model building so I have dropped them.
* And I also noticed that insured zip is the zip-ID given to insurance person and this also will not help us in model building so I’m going to drop this column.

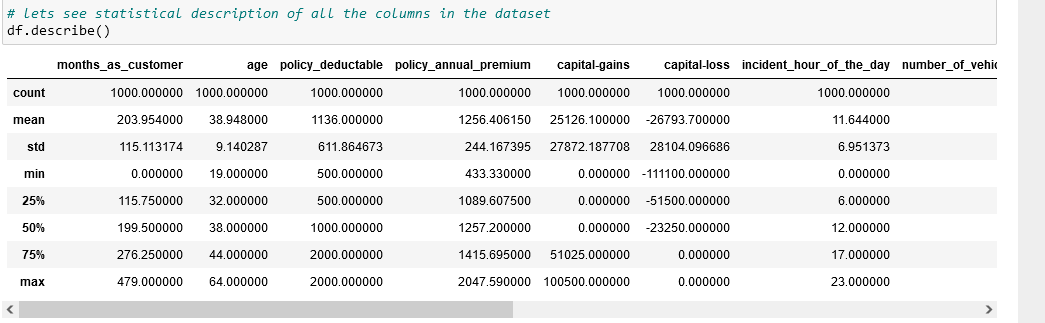


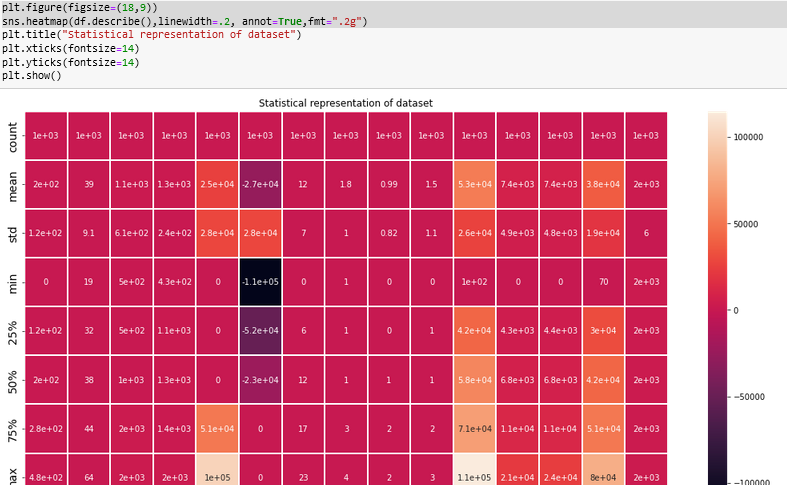


After droping those unnecessary columns we have 35 features left in our dataset.

**Checking for Null values :-** 

* As we can observe there is no null values are present in our dataset
* Describe the Dataset:-



* -Here we can see difference in mean and median(50% percentile) in columns, months\_as\_customer, policy\_deductable, capital-gains, capital-loss, number\_of\_vehicles\_involved, witnesses, total\_claim\_amount, injury\_claim, property\_claim, vehicle\_claim which denotes presence of skewness.
* -Here we can observe presence of outliers as well as most of the column have big gap in 75%percentile and max.
* 

This is the statistical representation of of our description of the dataset.

# Extracting Data:-[¶](http://localhost:8888/notebooks/Insurance%20claim%20fraud%20detection%20project.ipynb#Extracting-Data:-)

# There is many data which needs extraction.

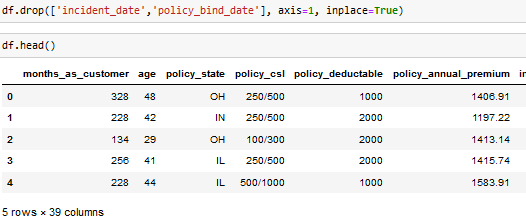
# Here 1st we extract this datatype object to datetime datatype.

# 

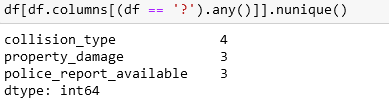
# 

Here we use lambda function to extract dateime type to date,month and year.

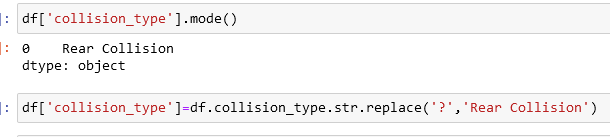
As we have changed those column to datetime type ,we don’t need those columns and we have to drop those columns..



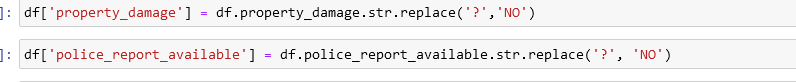
* I just had a look into value count of extracted columns I found single unique entries in incident\_year column which means all the entrices in this column are same so I don’t want to keep this unnecessary column in the dataset, so I have droped it.
* After extracting all the necessary columns from the old columns we have to drop old columns. If we don’t drop those columns they will behave as duplicate columns and create multicolinearity issue.
* I have noticed some unnecessary entries in the dataset like ‘ ? ’. It may be because of some tying errors or some techinal error we got some entries as ‘?’. So now it’s time to replace those unwanted entries.

****

* Checking for ‘ ? ’ entries in all the columns. I found these entries in 3 columns.



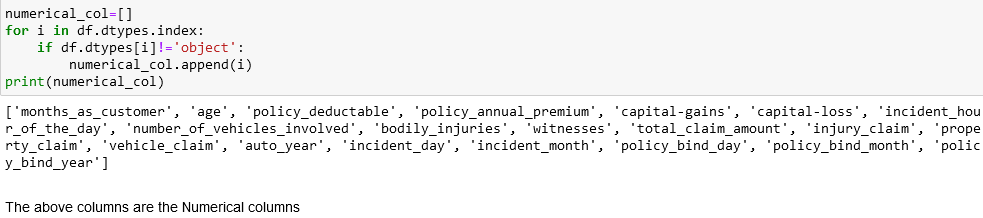
As we know collision\_type is a categorical type column so I have replaced the ‘ ? ’ values with it’s mode.



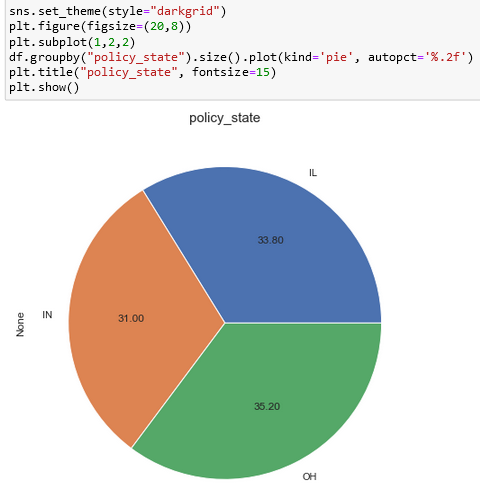
* And in property\_damage column NO has maximum count so I have replaced ‘?’ with NO.
* In police\_report\_available column NO has maximum count so I have to replace ‘?’ with NO.

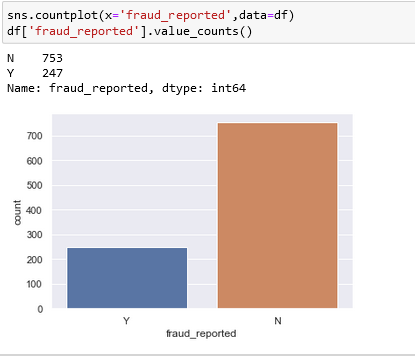
Now all the feature extraction is complete and the data is set for analysis.

# Find catagorical & numerical columns:-

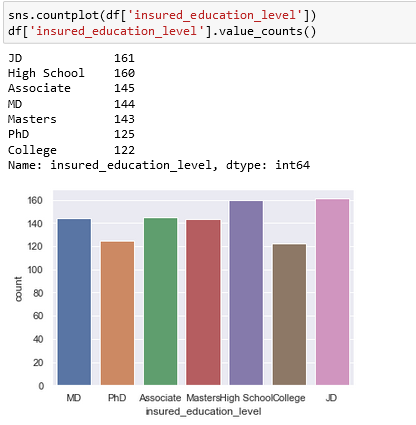


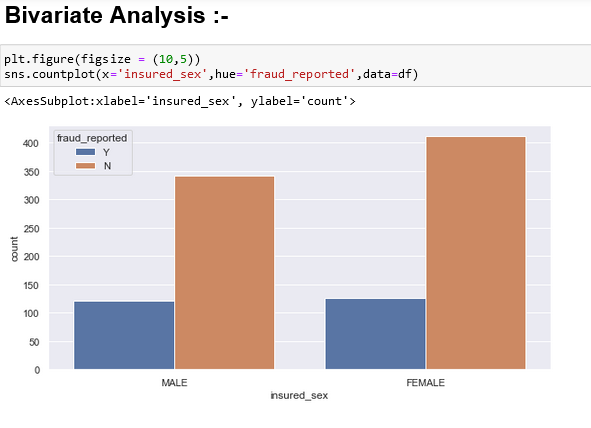
The above are the categorical and Numerical columns.

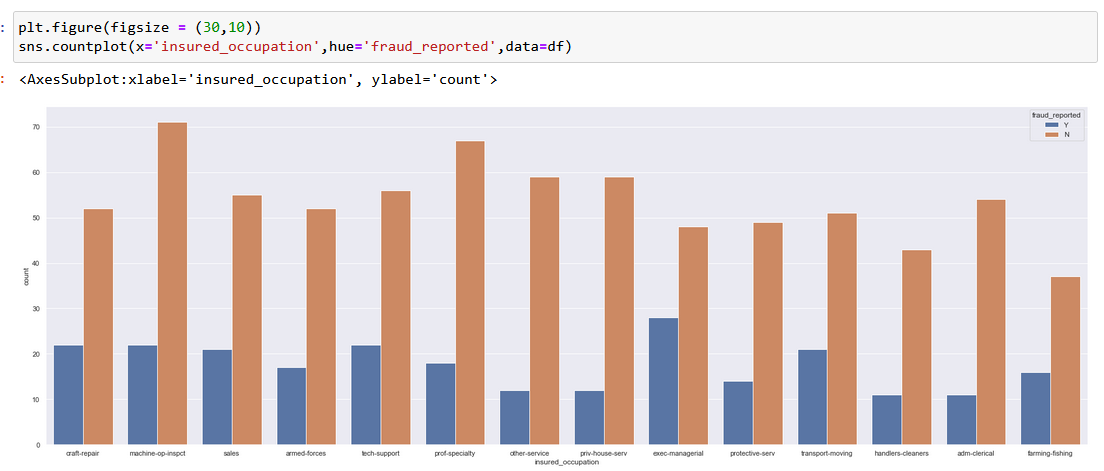
* Visualization of data:-  
   
* here we can see the Data is almost equally distributed within all the columns.
* the policy state is in IN is 31%,IL is 33.80% & Oh is 35.20 which is highest



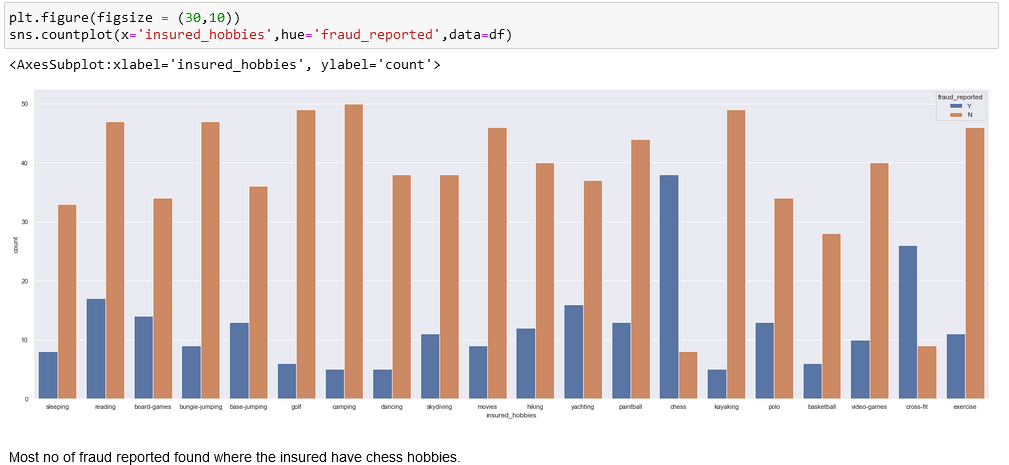
* from above countplot we can see 24.70% which is 247 in counts have reportedly fraud.

.

* Look at the factor plot for insured\_education\_level. The fraud count is less with college level educated and fraud not reported count is high for high school level educated which means less educated people are more trust worthy comparatively.
* This count plots show the qualification of the insured persons.As we can see most of the insured persons have qualification of JD and Highschool and all are equali distributed
* 
* Above is the factor plot for Insured sex I noticed that in both the genders the count of fraud reported is same. But the fraud not reported is high with females. Which means females are more trust worthy than men.

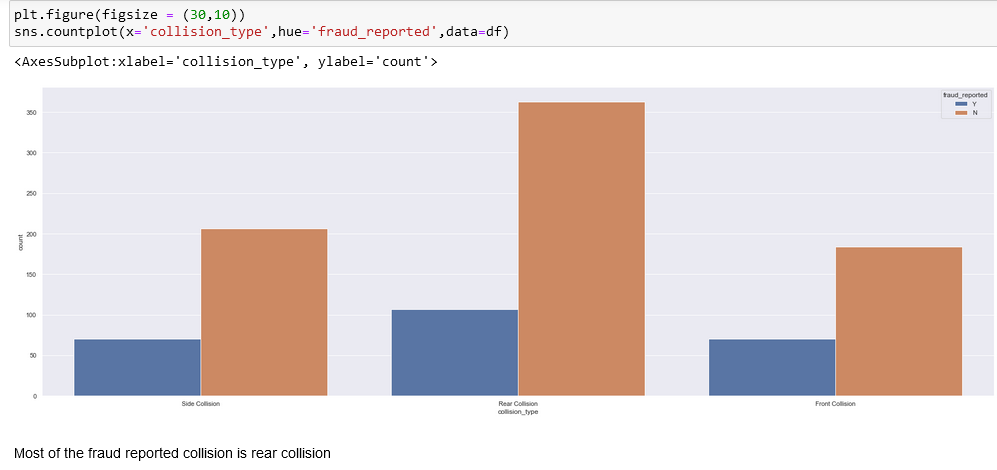


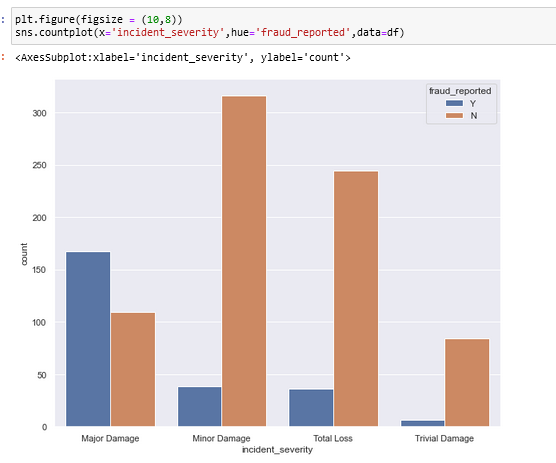
* Next plot is for insured occupation column. And the fraud count is more for exec-manegerial persons and fraud not reported count is high for machine-op-inspct. Which means good occupated persons looks most fraudulent.
* We can see most no of fraud\_reported occupation is Exec-managerial.

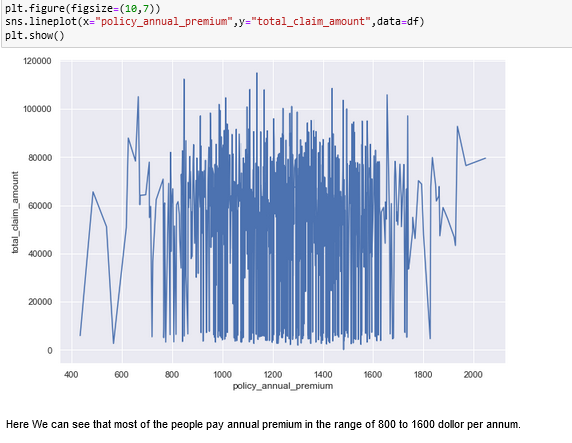


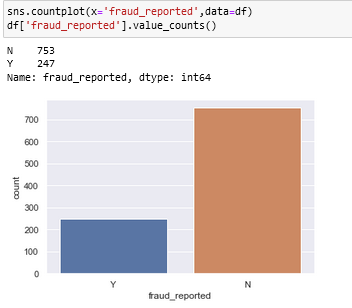
* If we look into the plot, most of the insured people having chess and cross-fit as a habit are more fraudulent. And fraud not reported count is high for insured people having camping, golf and kayaking as there habit.



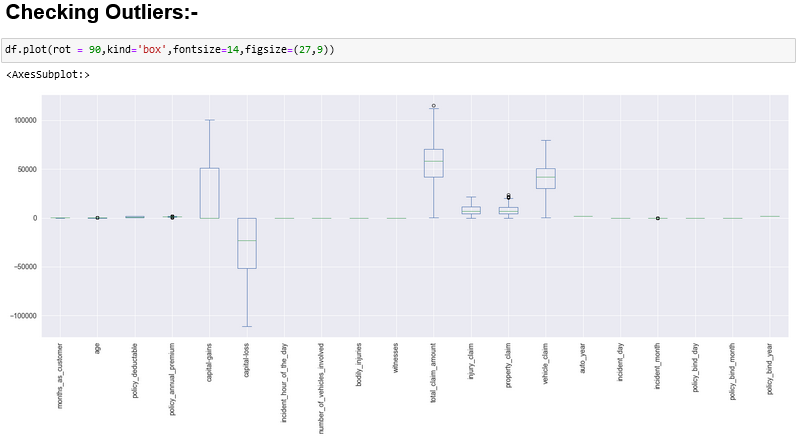
* In this count plot we can see the count is high for single and multi vehicle collision. The fraud reported and not reported is almost same for these two types of collision.
* 
* From above countplot we can see the count is high for Rear collision and in that case the fraud detected is less compared to fraud not reported.



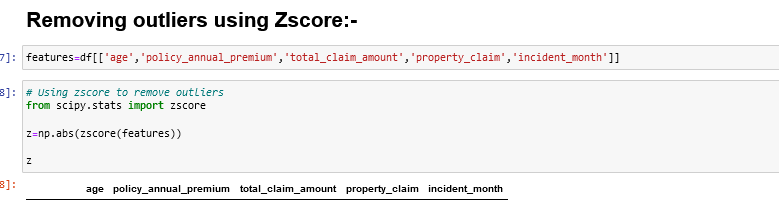
* This plot is for incident\_severity. If a person is climbing insurance on the condition of major damage then the chance of fraud reporting is very high as compare to any other cases.
* 
* From above observation we can see that the most of the people pay 800-1600 per annum.

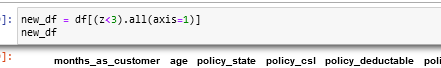


* As we know our problem was classification. And the target column fraud\_reported is imbalanced. The count of Y is less compared to N. We have to balance these counts to get a good model.
* **EDA Concluding Remark:-**
* Here I have checked for NaN values and I found there was no missing values in the dataset.
* I have extracted the necessary features from existing features to get better accuracy and dropped the old columns to avoid multicolinearity.
* If I keep the old columns as it is then they will act as duplicates in the model.
* I have also dropped the unnecessary columns. And also I replaced the ‘?’ entries with there suitable values.
* I have used both matplotlib and seaborn to visualize the data.
* To get better insight on the features I have used distplot, barplot, scatterplot and boxplot since most of my columns were categorical I have used all categorical plots.
* **Checking Outliers:-**

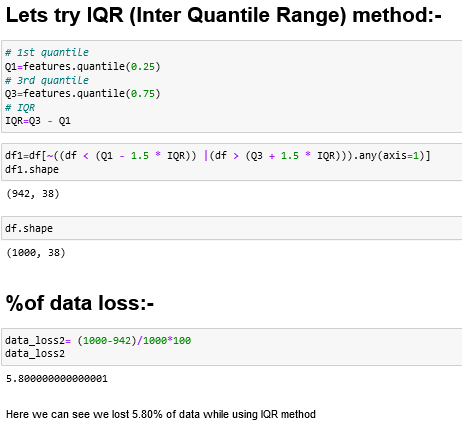


* Here, I have used box plot to check outliers. I found outliers in age, policy\_annual\_premium, total\_claim\_amount, property\_claim and incident\_month.
* Now I have to remove those outliers.



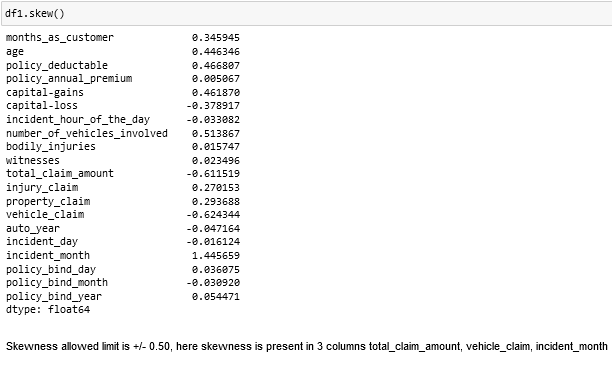


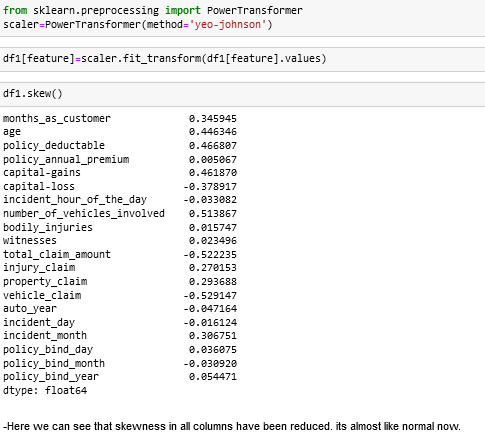
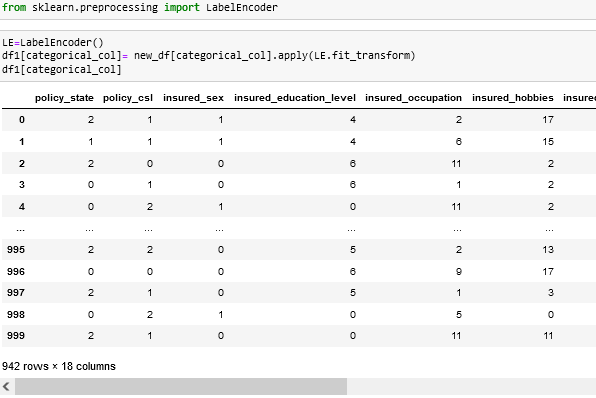
* By using Zscore here only 0.4% of data lost.
* So here we have to try another method which is IQR method

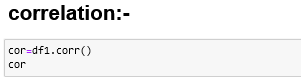


* In IQR method we lost 5.85 of data which is below 10% soo from now we consider df1 as our dataset.

**Now we have to check skewness:-**

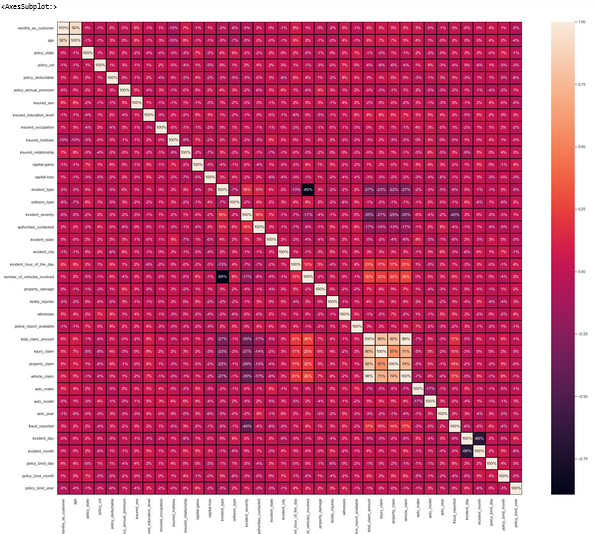
****

* Here I can notice skewness in total\_claim\_amount, vehicle\_claim, and incident month.
* 
* To remove skewness I have used Yeo-johson method.
* After used Yeo-Johnson se can see the skewness is reduced to +/- .5 hich is good .
* Now it’s time to encode our categorical columns
* 
* Here I am using LabelEncoder to encode our categorical dataset.and here is the result we can see.
* And we have created a new dataframe as df1

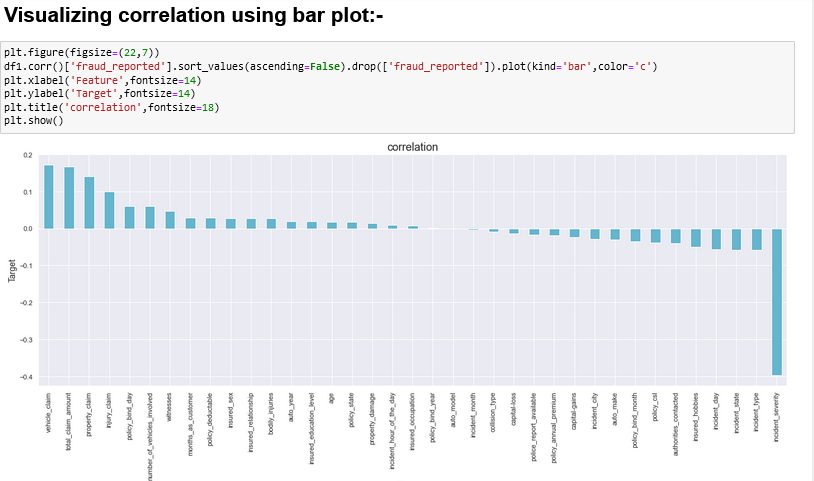


* After checking the correlation, to get better insight on the corr values I have plotted heat map below.





* From above correlation sns plot we can see:-
* -Here we can observe multicolinearity issue between incident type and number\_of\_vehicles\_invovled.
* -There is multicolinearity issue between incident\_month and incident\_day.
* -There is multicolinearity issue between total\_claim\_amount and injury\_claim.
* -There is multicolinearity issue between total\_claim\_amount and property\_claim.
* -There is multicolinearity issue between injury\_claim and vehicle\_claim.
* -There is multicolinearity issue between total\_claim\_amount and vehicle\_claim.
* -There is multicolinearity issue between vehicle\_claim and property\_claim.
* -There is multicolinearityissue between total\_claim\_amount and property\_claim.
* Looking into the heat map I can say that there is multicolinearity issue and to get better insight on targets correlation with other features I have ploted bar graph.

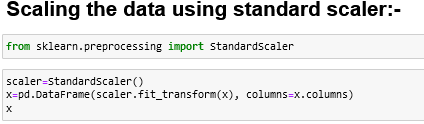
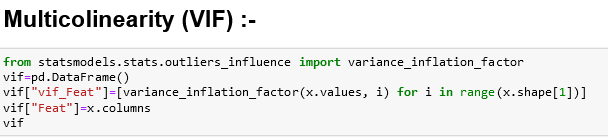


* Policy\_bind\_year, auto\_model, insured\_occupation, auto\_age and incident\_hour\_of\_the\_day are very less correlation with target but I keep those columns and build the model. Since I don’t want to loose any data so first keeping all the columns let me build the model. After looking into the accuracy if I feel I can increase the accuracy by deleting these less correlated columns then again let me come back and delete these columns.
* Now my dataset is ready for preprocessing.

**Pre-processing:-**

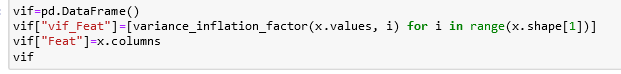
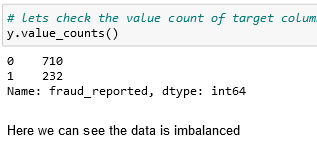
1st I have to separate the both the dependent and independent features.

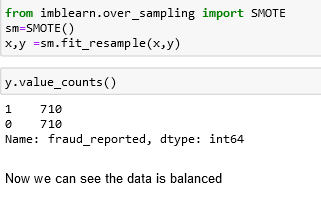


* I have taken x as all independent features and y as dependent/target feature.
* Then I have to scale my independent features to get the same range in all the columns. If I don’t scale my independent columns then there is a chance that my model may get baised. So In this perticular case I have used Standard scaling as I have removed all outliers and skewness from the dataset it is good to use standard scaling else we have use MinMax scaler.
* 
* Now I have done scaling. But I have left out with multicolinearity. I have to check VIF(variance inflation factor) now.
* 

I can notice a high VIF for total\_claim\_amount, so I have dropped this column first.



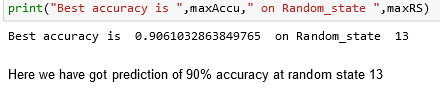
* After that again I checked for VIF
* 
* Now we can see that all the vif value is below 10 soo the multicolinearity issue is solved.
* Now I have to check the value counts of our target column or label.
* 
* There is oversampling issue soo I have to import SMOTE for balancing

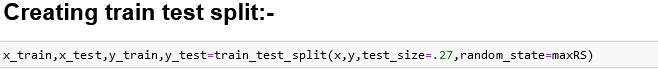


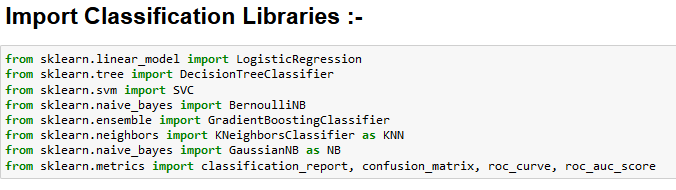
* My data is all set for model building. Let’s go ahead with classification algorithms.
* **Building Machine Learning Models:-**

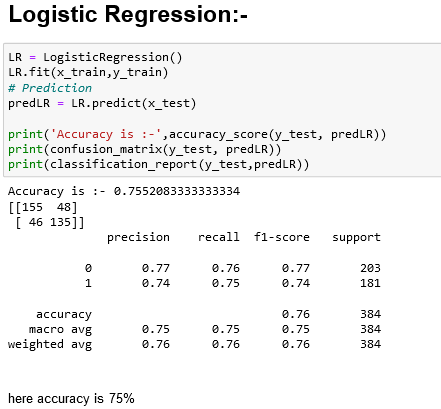
# Finding Best Random State and Accuracy:-

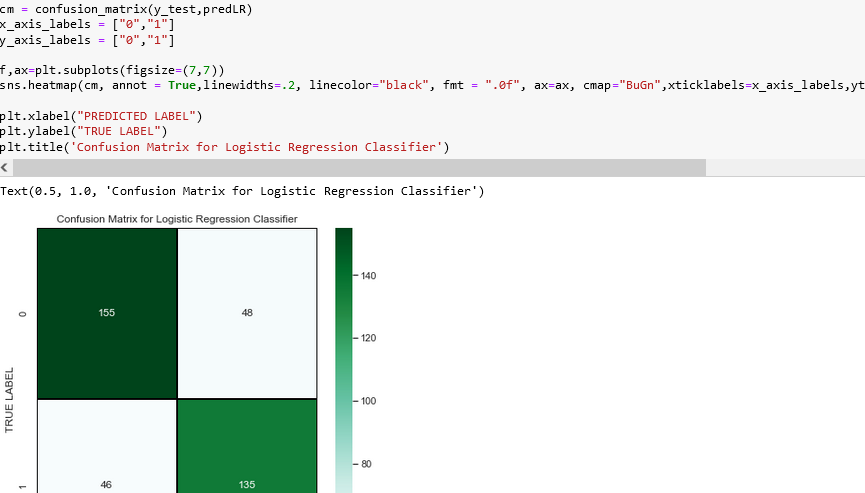
# 

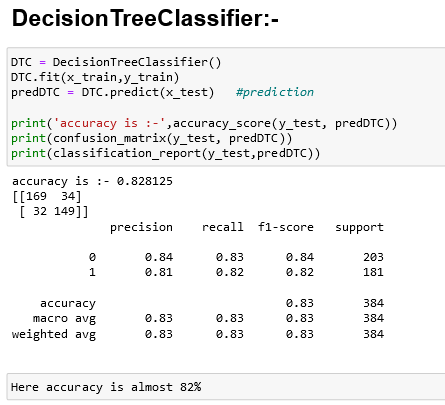


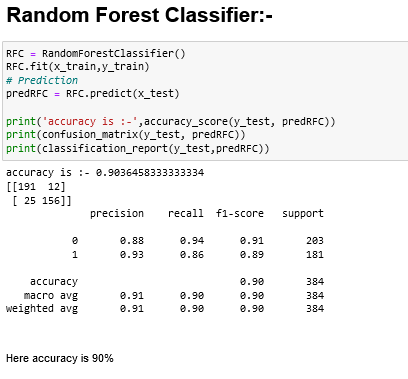


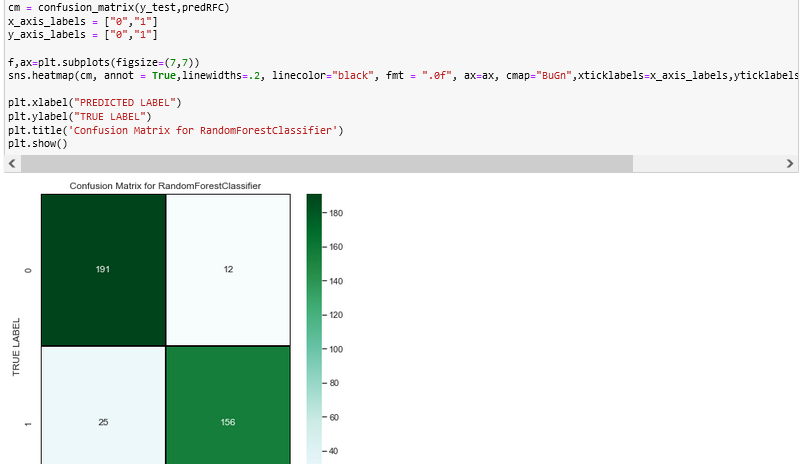
* Here I Created train and test data as x\_train, x\_test and y\_train, y\_test.
* 
* I have used Cross validation as model evaluation metrics for all the algorithms. And I have used accuracy\_score, Confusion metrics as metrics in model building.

****

****

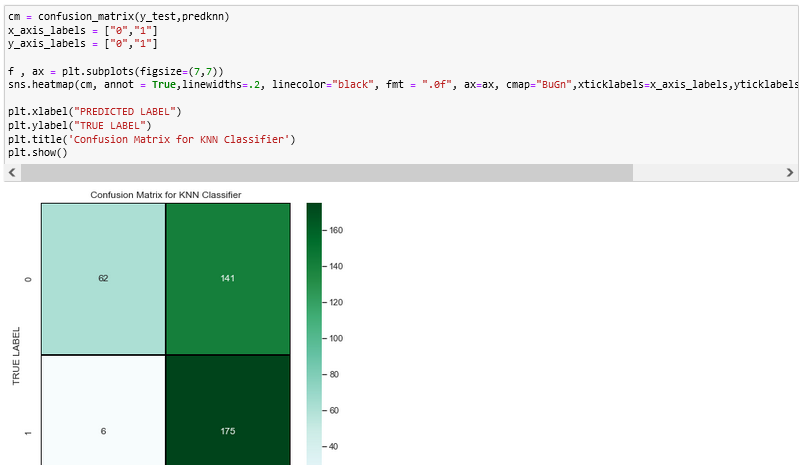
* Here Logistic regression is giving me 755 of accuracy which is preety good and here is the confusion matrix for logistic regression.
* 
* 
* Here Decision tree classifier also give a good accuracy of 82% and it is the confusion matics for DTC.



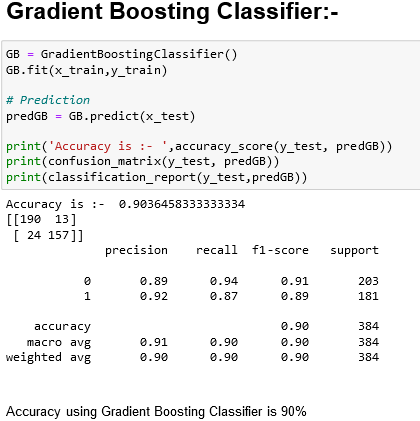


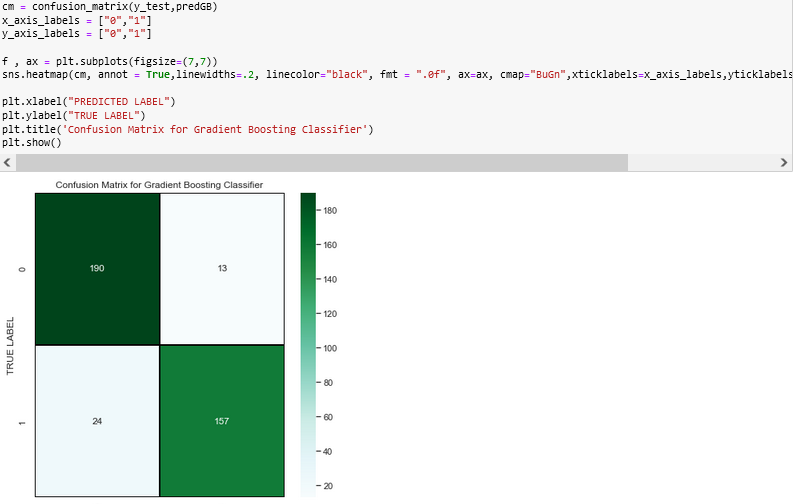
* Here we can see that our model RFC is giving us a very good accuracy score of 90%.and this is the confusion matrics for RFC.



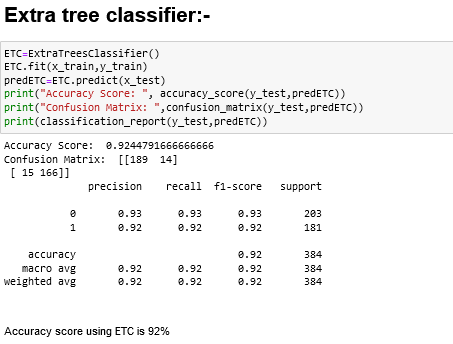


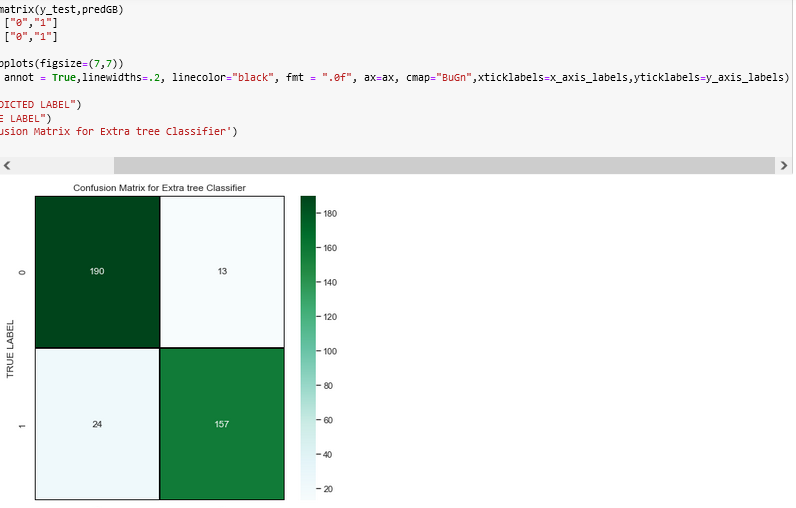
* Here the accuracy of KNN is 61% which is not soo good as compare to previous models.
* And this is the confusion matrics for KNN.



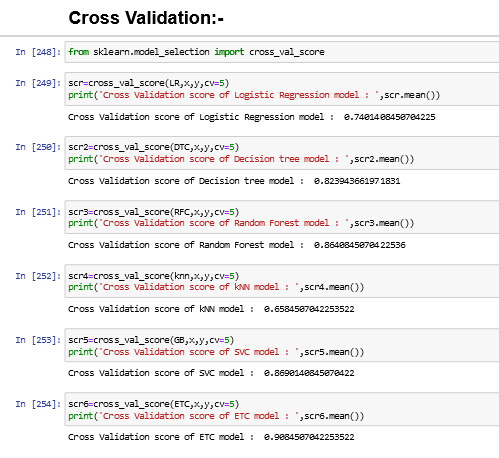


* The accuracy of the model gradient Boosting classifier is 90% which is very good.
* This is the confusion matrix for GBC.

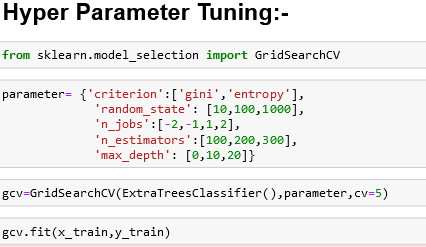




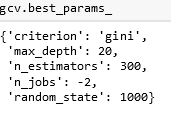
* In this model ETC the accuracy score is 92% which is a very good accuracy.
* And here is the confusion matrix for ETC.



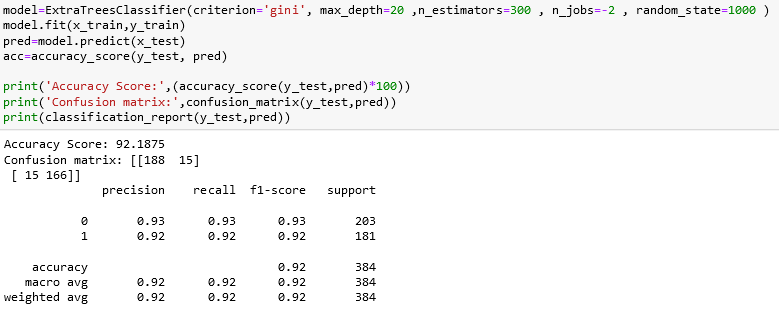
* From above we can see the cross\_validation scores of all the models.here we can see the Least difference between model accuracy and cross\_val\_score is ExtraTreesClassifier.
* so i found ETC as my best model here.

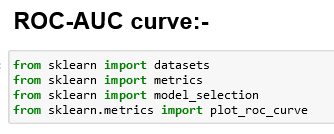


* Using the above parameters list I’m tuning my best model i.e., Extra Trees Classifier. And I have to choose the best parameters in above parameter list, with those parameters I have to build the best model.

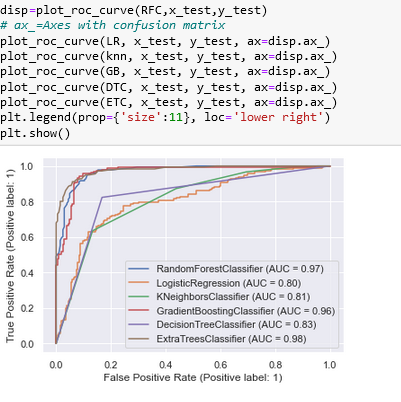


* These are the best parameters we found.
* After knowing the above best parameters I have to run for improving model accuracy.



* We can see even after tunning the model accuracy is same which means the default parameters used by the model were giving the best accuracy.
* And the model is now ready with 92.44% accuracy which looks good.
* 

The ROC-AUC curve for all the above model is as shown below:-



* The AUC value is high for Extra trees and also I found the least difference of model accuracy and cross validation score for Extra Tree Classifier.
* So I’m choosing Extra Tree Classifier as the best model.

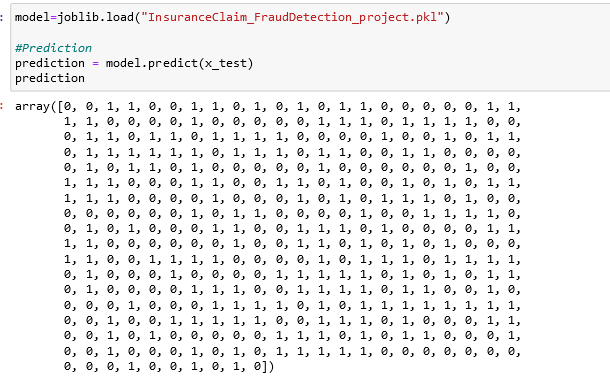
# Saving the model:-

# 

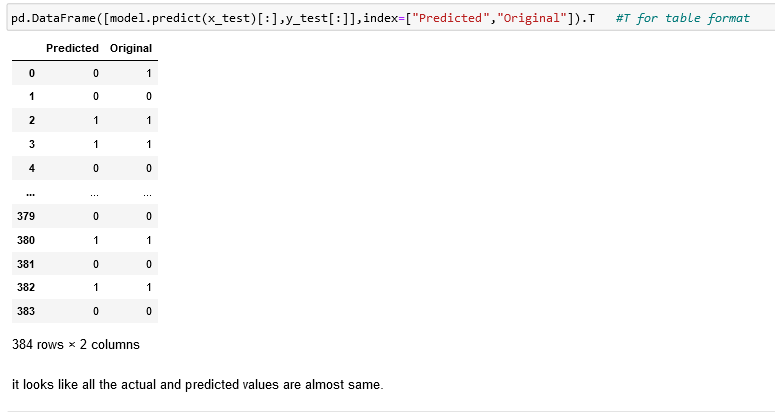
* I have saved it using .pkl. As InsuranceClaim\_FraudDetection.pkl

# Lets Predicting the saved model:-

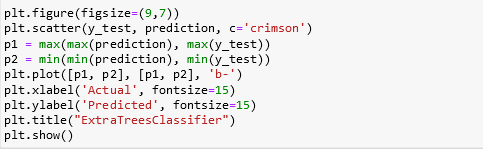
* Now using the saved model I am going to predict wheather the insurance claim is fraudulent or not.



These are the predicted values which I am predict using our best model.



* Here I am comparing both the actual and predicted values.
* And we can see that it seems like actual an predicted values are almost same.
* So here we can say that thee model which we created is pretty much good with accuracy in prediction.



# 

# In this scatter plot Blue line is the actual values & and red dots are predicted values.

# *Concludeing Remarks:-*

* This is the problem which needs a good vision on data, and in this problem Feature Engineering is the most crucial thing.
* You can see how we have handled numerical and categorical data and also how we build different machine learning models on the same dataset.
* Here we use multiple models as well as cross validation then after came to know our best model.
* Using this machine Learning Model we can can easily predict the insurance claim is fraudulent or not.

# we could reject those application which will be considered as fraud claims.

# 