# ****Insurance claims — Fraud detection****

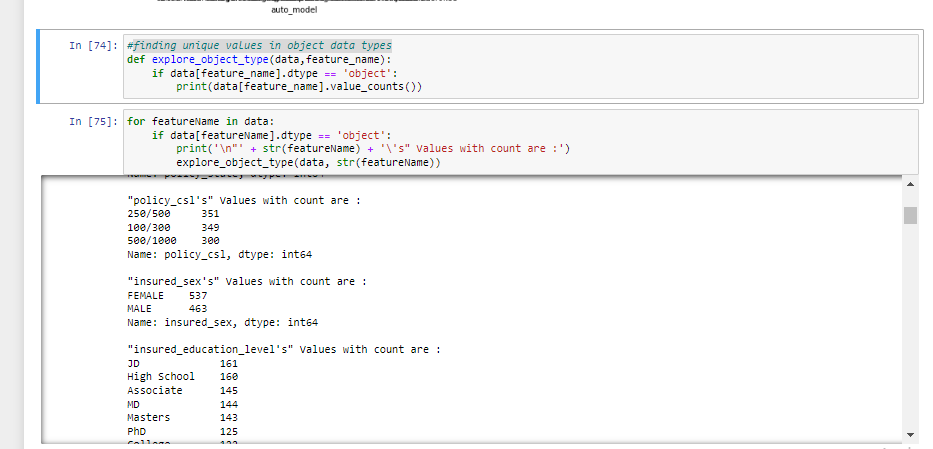
Insurance fraud detection is a challenging problem, all insurers have to face given the variety of fraud patterns and relatively small share of detected frauds in typical samples. It becomes a challenge to detect fraud , when most claims are genuine. However, even a small amount of fraudulent cases can cost the company a lot. Hence, it is essential for companies to be aware about which claims are false and which are genuine.

In this dataset, various attributes that may be linked to fraud and other circumstances are included. Some of the fraud could include staging the incident, misrepresenting the situation and information on the extent of damage caused.

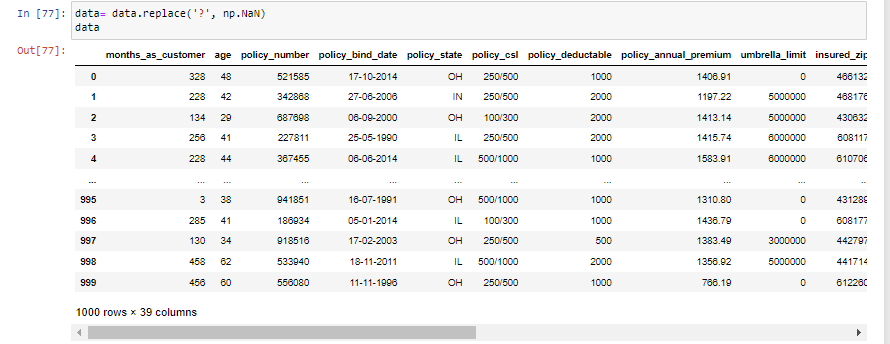
The goal of this assignment is to build an effective model that can detect auto insurance fraud. The savings from loss prevention needs to be balanced with cost of creating a false model. This is the objective that is kept in mind before building a machine learning model.

**Data Analysis**

* I have observed that the data set is small. The shape of the data is 1000, 40. The feature columns are too many and the important ones needs to be selected and the others to be eliminated for accuracy
* It is observed that ‘\_c39’ feature column has null values. This column is dropped
* I have separated ‘object ‘ data type columns and checked for their unique values and composition to find relationships



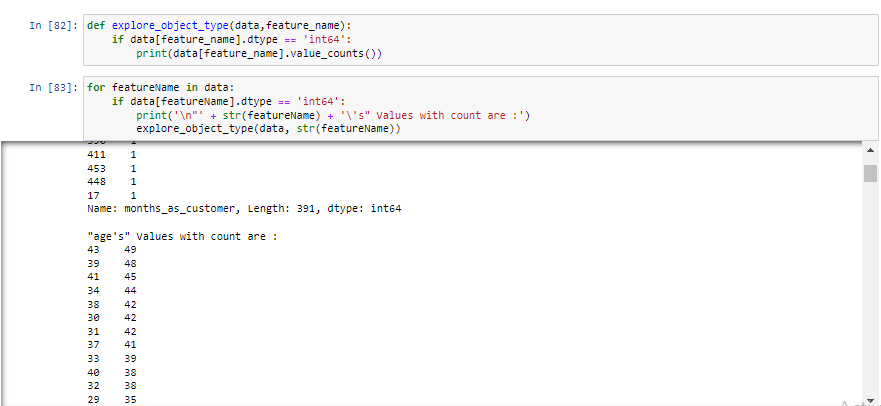
* there are '?' unique values for features columns: property\_damage's, police\_report\_available's, collision\_type's. I have treated these columns with the panda replace function



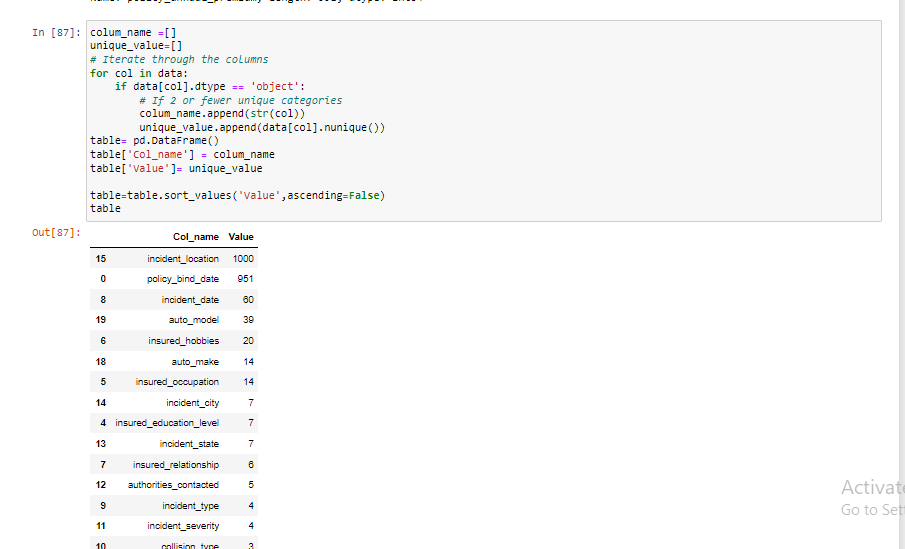
* I have used the ‘ mode’ function to fill these columns and replace nan. I have also checked if ‘?’ is removed via value counts function in sns countplot



* Similarly, I have separated ‘int 64’ and ‘float 64’ data type feature columns to check for values and remove any unique ones



* The column umbrella\_limit's, capital-gains's, capital-loss's, property\_claim's, injury\_claim's is filled with zeros
* I have checked the unique values with the following code. As yoi can see below, the unique values for Dropping incident\_location and policy\_bind\_date are high, I have dropped these feature columns

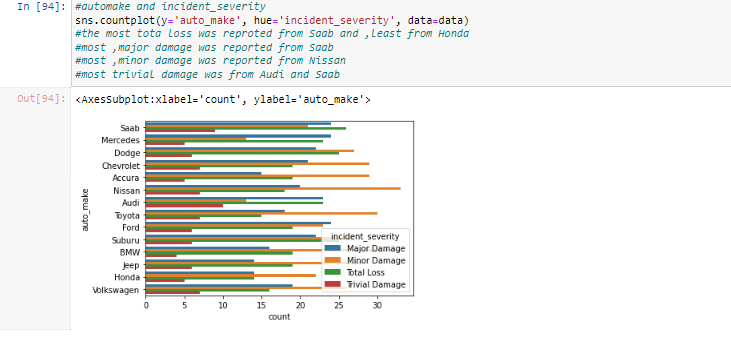


**Exploratory Data Analysis**

* Following observations were found after finding correlations between target columns and different features
* Minor damage cases were the most prevalent while filing claims
* People who played chess as a hobby, reported the most fraud
* Most fraud reports were from Airlington and least fraud reported was from Northbrook
* Northbrook city had least amount of total loss. The most minor damage was from Springfield. The maximum total loss was from Columbus and Northbend.



* Automake and vehicle claims: Audi has the highest claims and Nissan has shown least claim
* Automake & Incident severity

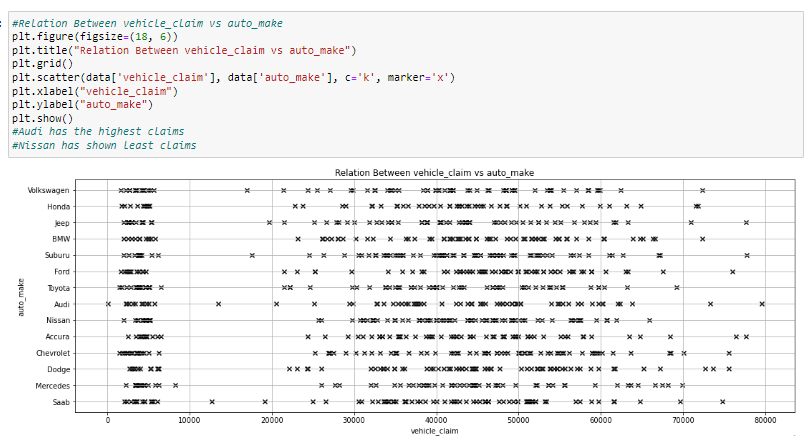


* The most fraud cases were reported were for one vehicle as compared to two and three vehicles
* Number\_of\_vehicles\_involved and incident\_city: Airlington and Hillsdale had maximum no of 1 vehicle involved, Hillsdale had maximum no of 2 vehicle involved, Columbus and Springfiled had maximum no of 3 vehicle involved, Northblend & Arlington had maximum no of 4 vehicle involved

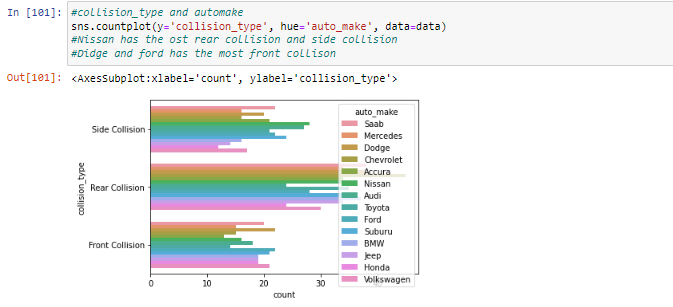
As we can see Airlington and Hillsdale are the cities with max car involvement

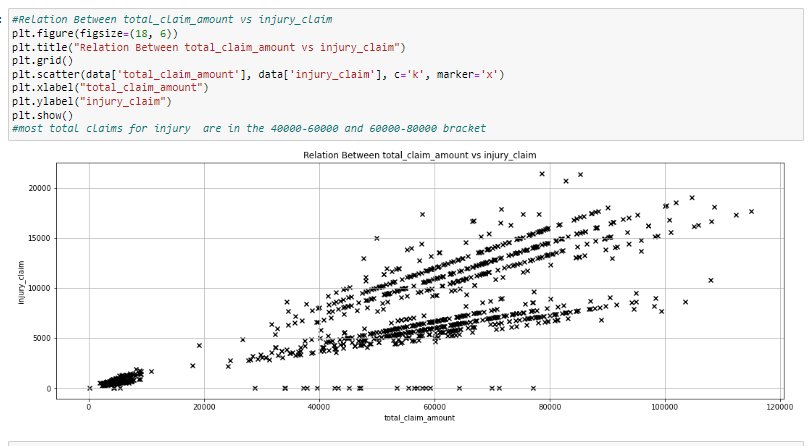


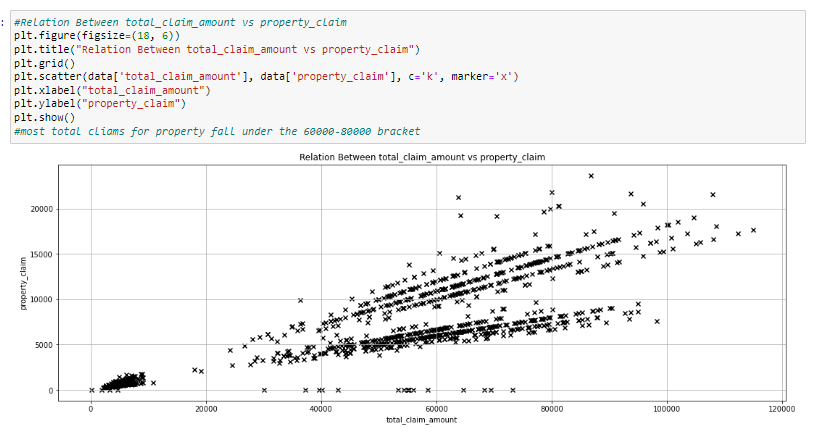
* There ae more claims that are left unreported for fraud for property claims than the ones that are reported

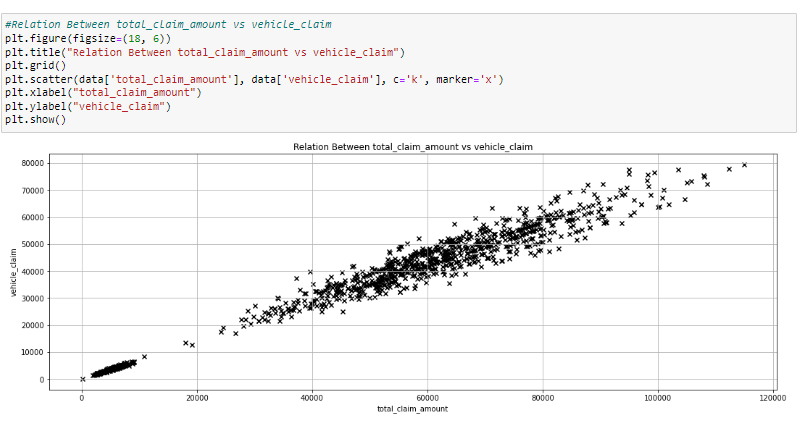


* Rear collision( out of the other collision types – dront and side) is the most reported fraud

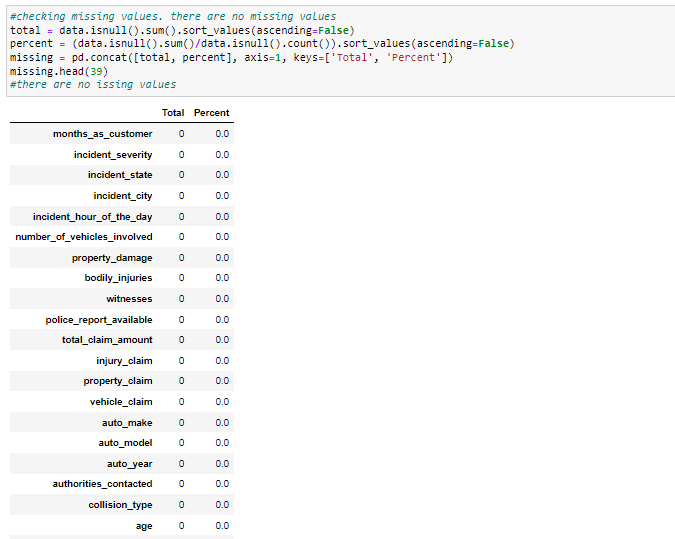




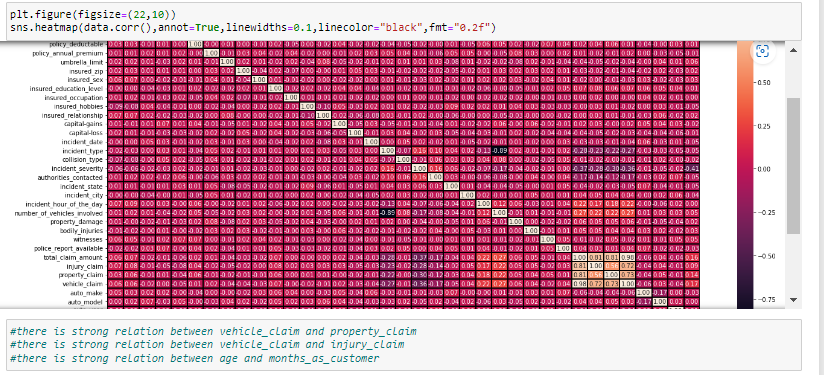


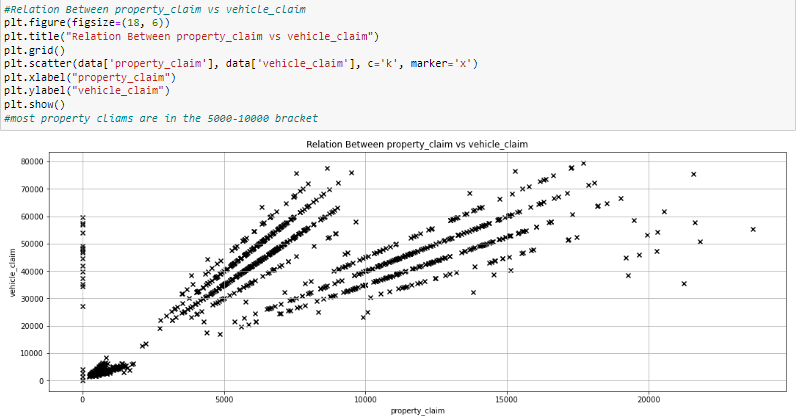


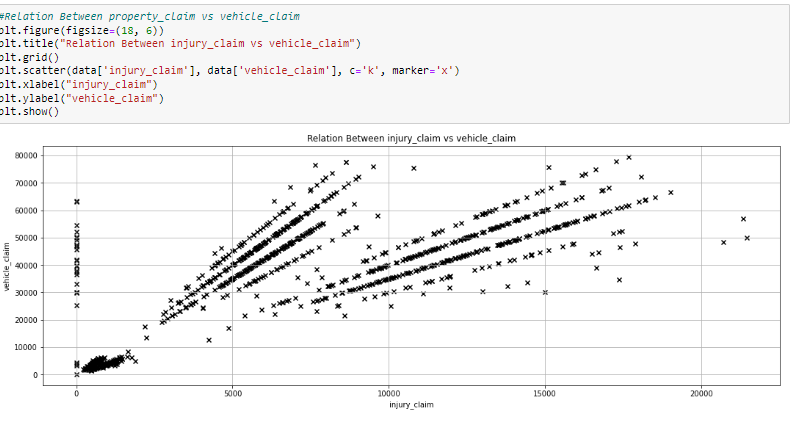
* Double checked for missing values and there were none

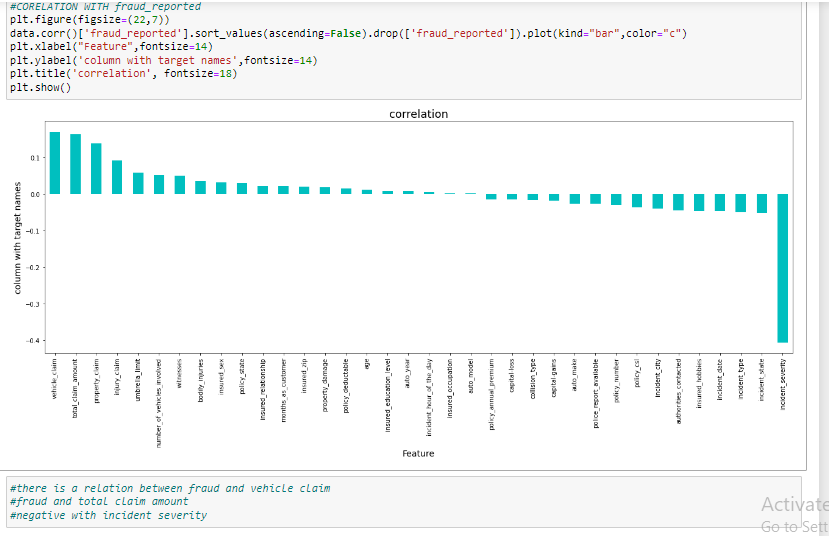


* I have used label encoding to turn object data type into numerical data columns
* Used density plot to determine the skewness of all columns
* I have used correlation matrix to further determine any relationships







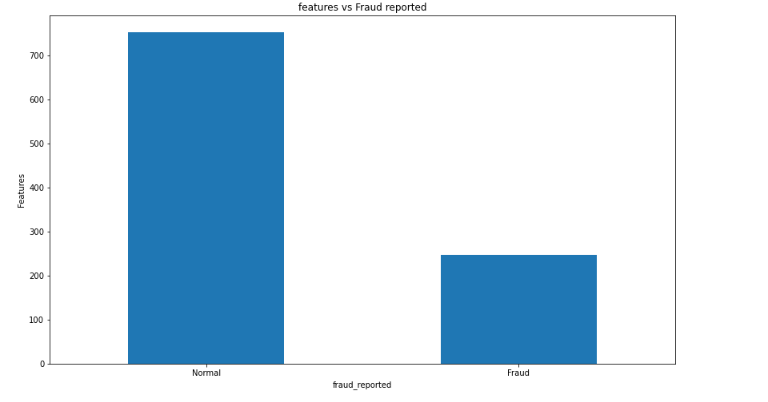


* Used ‘data.skew().sort\_values(ascending=False) ‘ to find the skewed columns. umbrella\_limit column is heavily skewed

**Data transformation**

After analysis , I have transformed the data with the following technique:

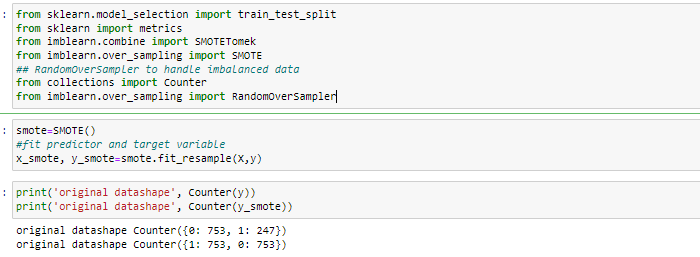
* After performing analysis, I used the ‘SMOTE’ technique as the data was imbalanced



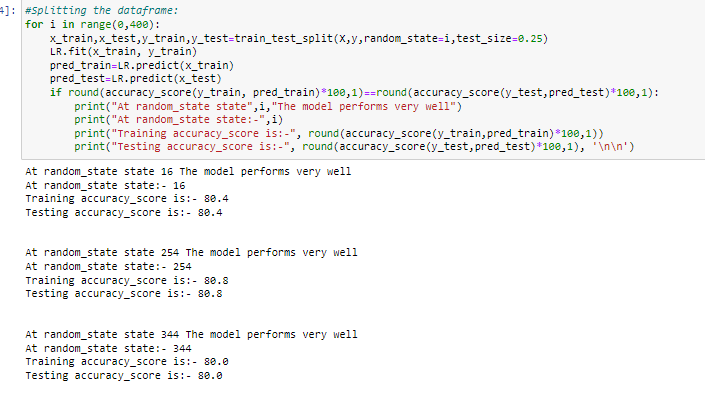


Data Transformation

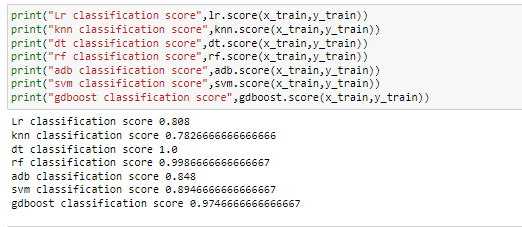
* I have used power transform and standard scalar to remove outliers
* Data is balanced after using SMOTE

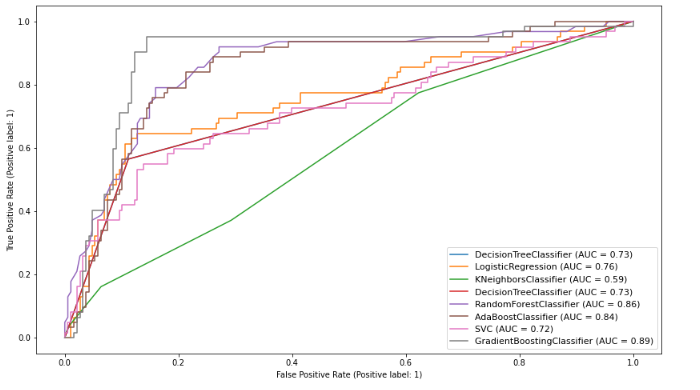


* The next step was to split the data and select the best random state

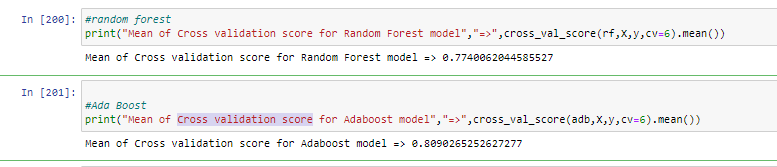


* The model is trained with Logistic Regression, KNearest Neighbour, Decision Tree, Random Forest, Adaboost Classifier , Support vector classifier and Gradient Boosting Classifier. The moel is then trained.
* After establishing the classifications score, the ROC curve is built to check the scores for the best alternative

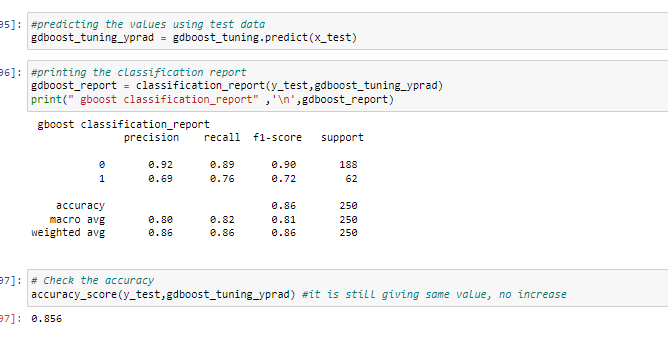




* Based on the results, Gradient boosting, Adaboost and Random forest for modelling were selected.
* With SMOTE scores the classification increased dramatically, however the accuracy increased by little.
* I tool out eman with Cross validation score fr the above three techniques



* I carried out hyper tuning with grid search CV. There was not a dramatic increase in the scores, they were almost the same.
* GD boost had the best accuracy with 86%



Finally, the model is saved with the pickle library.

I could have worked better with the smote method and used it to get a better accuracy score.

The challenge with this data set was handling the small and imbalanced data.

The method we have used for this data set is Predictive analytics. It observes the historical data as well as existing external data to find patterns and behaviors.