**Insurance Claims- Fraud Detection**

Submitted by:

DEEPTHI MALLU



**INTRODUCTION**

Insurance fraud isan illegal act on the part of either the buyer or seller of an insurance contract**.**  Fraud has become a major issue for insurance industries and most commonly seen activity. Each year, insurers are subject to 1,40,000 dishonest claims, with a total value of more than £1 billion. 

This article focuses on claim data of an Automobile insurance company. Because of fraudulent claims the insurance companies are losing huge amounts of money, which indirectly affects the public. Therefore, it is important to know which claims are genuine and which are fraud. All the time, the insurance companies cannot check on them personally weather the applied insurance is correct or not, since that is a time taking process and directly which result in the cost.

Spotting a fraudulent one among hundreds of claims is tough. But with predictive analysis, we can build the algorithms and be able to identify fraudulent claims. The insights drawn from the data are faster because they are the combination of the old rules and new tools. The tools run on real-time data mining, testing for the search and exception scenarios.

In this article, we will see how we can draw insights from the attributes provided by the insurer and build machine learning models to predict which claims are likely to be fraudulent. This information can narrow down the list of claims that need a further check. It enables an insurer to detect more fraudulent claims.

Problem Definition

The main aim of this project is to build a predictive model that can detect auto insurance fraud. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims.

Insurance fraud detection is a challenging problem, given the variety of fraud patterns and relatively small ratio of known frauds in typical samples. While building detection models, the savings from loss prevention needs to be balanced with the cost of false alerts. Machine learning techniques allow for improving predictive accuracy, enabling loss control units to achieve higher coverage with low false positive rates.

Insurance frauds cover the range of improper activities which an individual may commit in order to achieve a favourable outcome from the insurance company. This could range from staging the incident, misrepresenting the situation including the relevant actors and the cause of incident and finally the extent of damage caused.

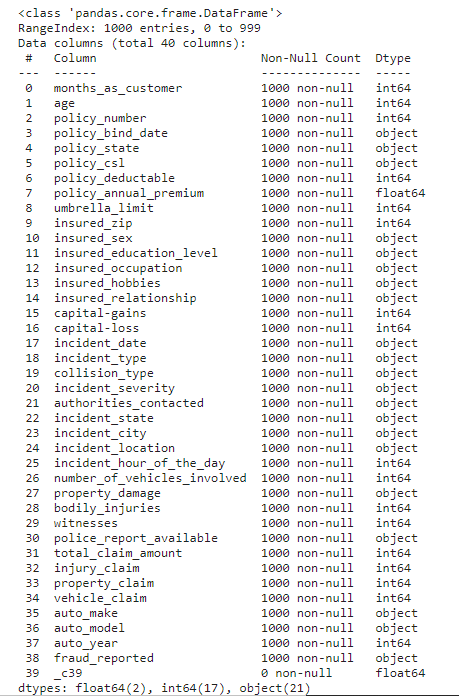
# Data Analysis

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.

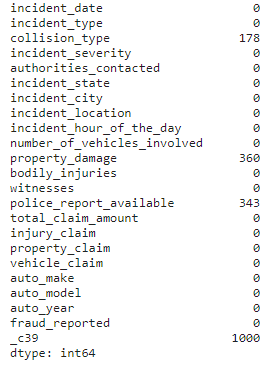
The given dataset contains 1000 rows and 40 columns. The column names like policy number, policy bind date, policy annual premium, incident severity, incident location, auto model, etc.

Data Analytics Especially useful when fraud is hidden in large data volumes and manual checks are insufficient Can be used reactively or proactively. The obvious con of this data set is the small sample size. However, there are still many companies who do not have big data sets. The ability to work with what is available is crucial for any company looking to transition into leveraging data science.

Checking for the type of attributes present in the dataset using info method. The info method gives us the type of columns and the memory used by the columns.

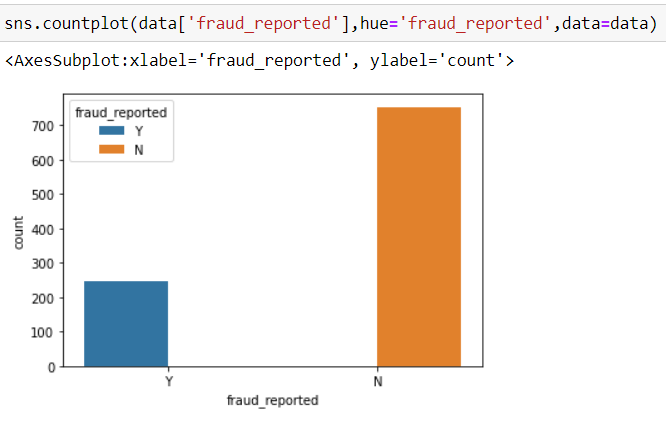


There are some variables which contain the null values given as character ‘?’. The number of null values present in the columns are given below.

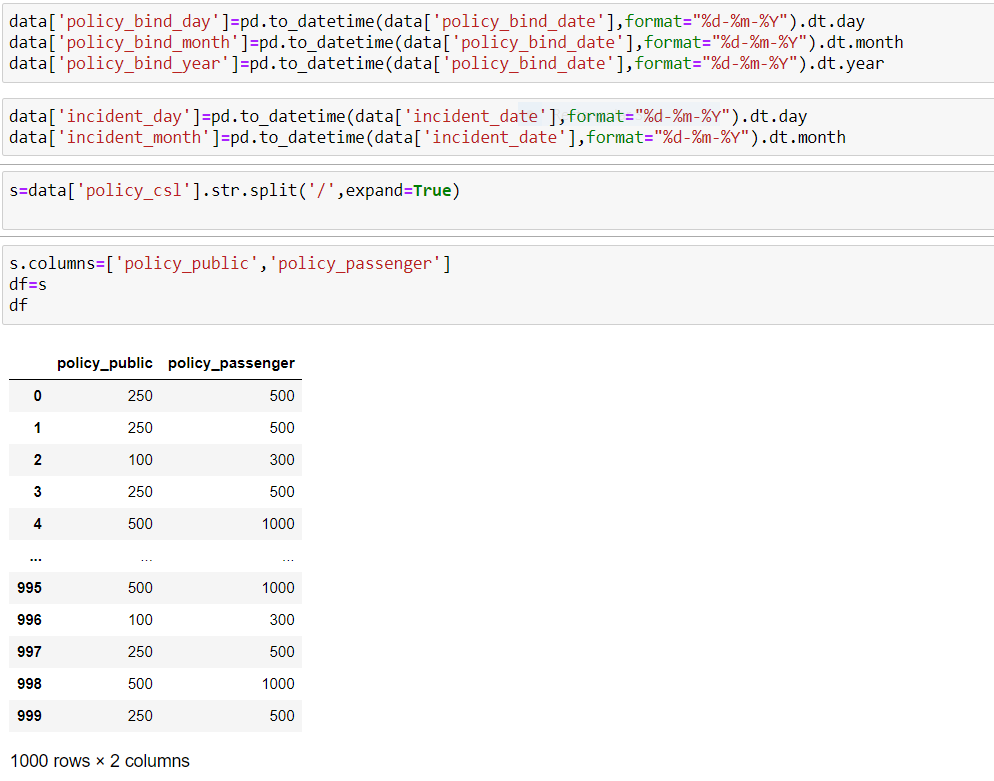


# Exploratory data analysis

Exploratory data analysis was conducted on dependent variable, Fraud-reported. There were 247 frauds and 753 non-frauds. The frauds are recorded as 24.7% and genuine are recorded as 75.3%.

as 24.7% and 75.3% are recoreded

There is a column named policy bind date which means the date on which the policy was made, that was given as object type which cannot be understand by our machine learning models. So that need to be converted as integer, so I have applied string split function as shown below.

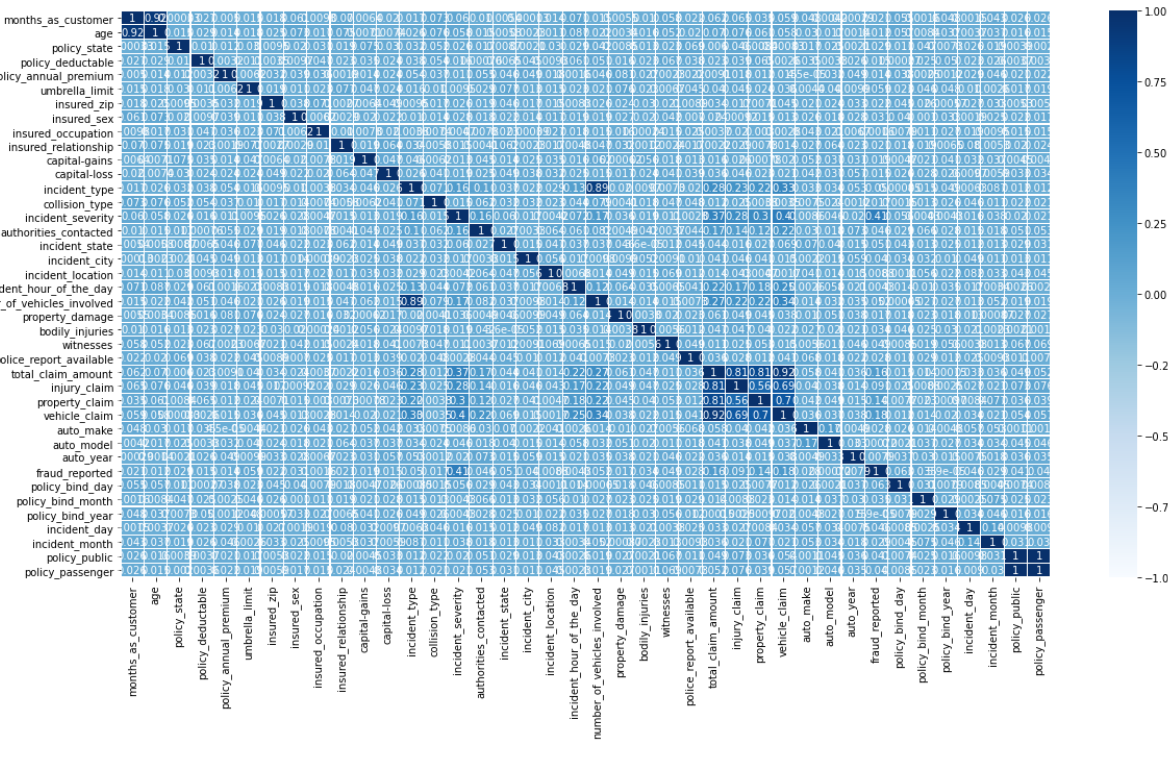


**Correlations among variables:**

1.The Dark blues gives the highest correlation and light blue gives the less correlation.

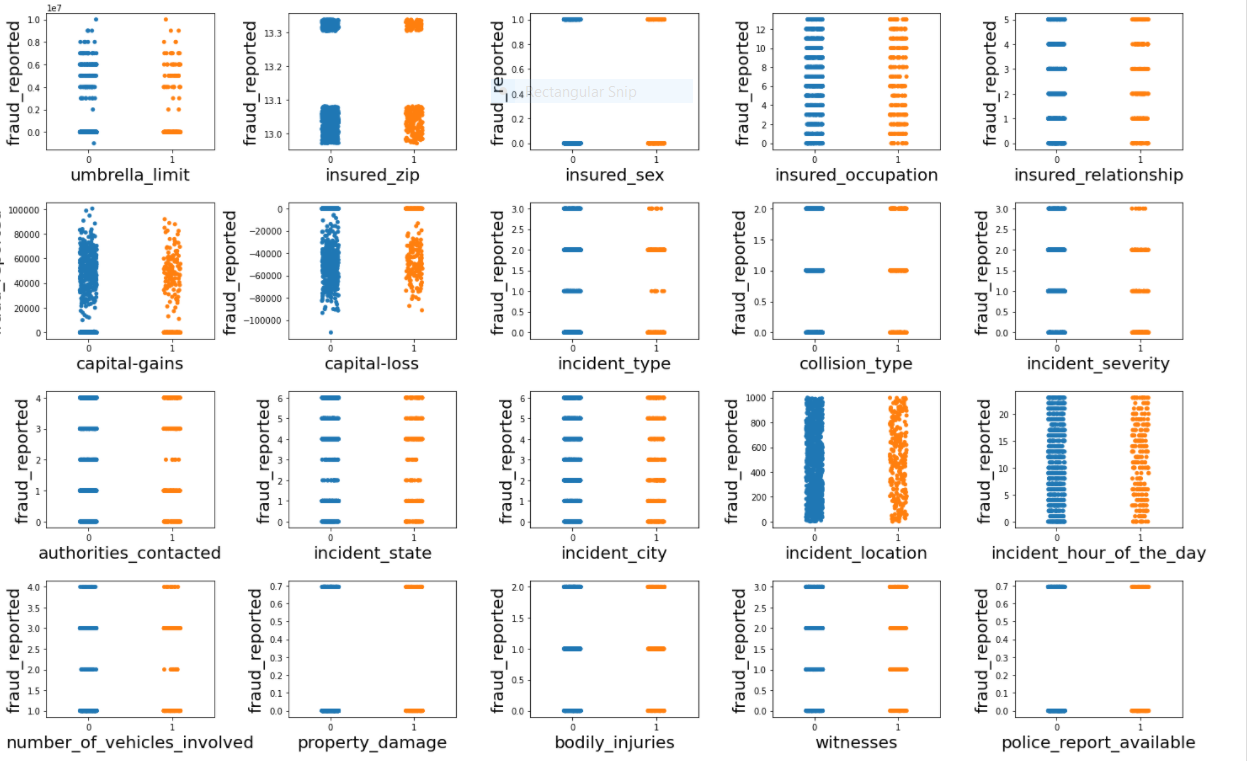
2.total\_claim\_amount is highly correlated with different amount\_claims. No of vehicles involved is highly correlated with incident type.

3.policy pubic and policy passenger are highly correlated.



**Visualizing variables:**

I have plotted strip plot to check the relation between the fraud reported and the features only the incident severity is positively correlated with our label.



## I have plotted count plots on all the categorical columns and visualised how the data is distributed among them.

## 

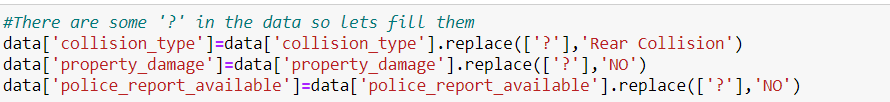
## **Pre-processing Pipeline**

 Data Pre-processing is a technique that is used to convert the raw data into a clean data set. It a predominant step in machine learning to draw insights and achieve better results. The quality of the data results better performance of the models. In General datasets are Incomplete, noisy, and inconsistent in nature of real-world. By employing Data pre-processing we can maintain quality of data by cleaning.

**Treating null values**

Most of the cases the null values are filled by NaN’s. But, in this case the null values are filled by ‘?’ which need to be filled. In our dataset the null values are present in collision\_type, property\_damage, police\_report\_available, and \_c39 with 178, 360, 343 and 1000 number of null values.

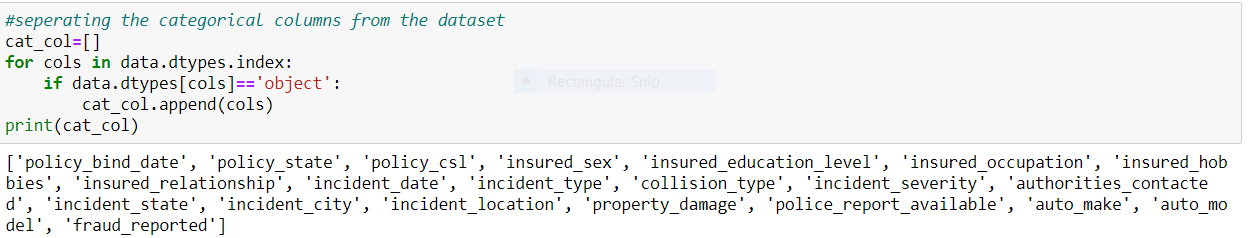
Imputing techniques are employed to fill null values, in this I have used replace method to fill null values for collision\_type, property\_damage, police\_report\_available, but \_c39 has completely null values so I have dropped that column since there is no data present in it.



**Encoding Techniques**

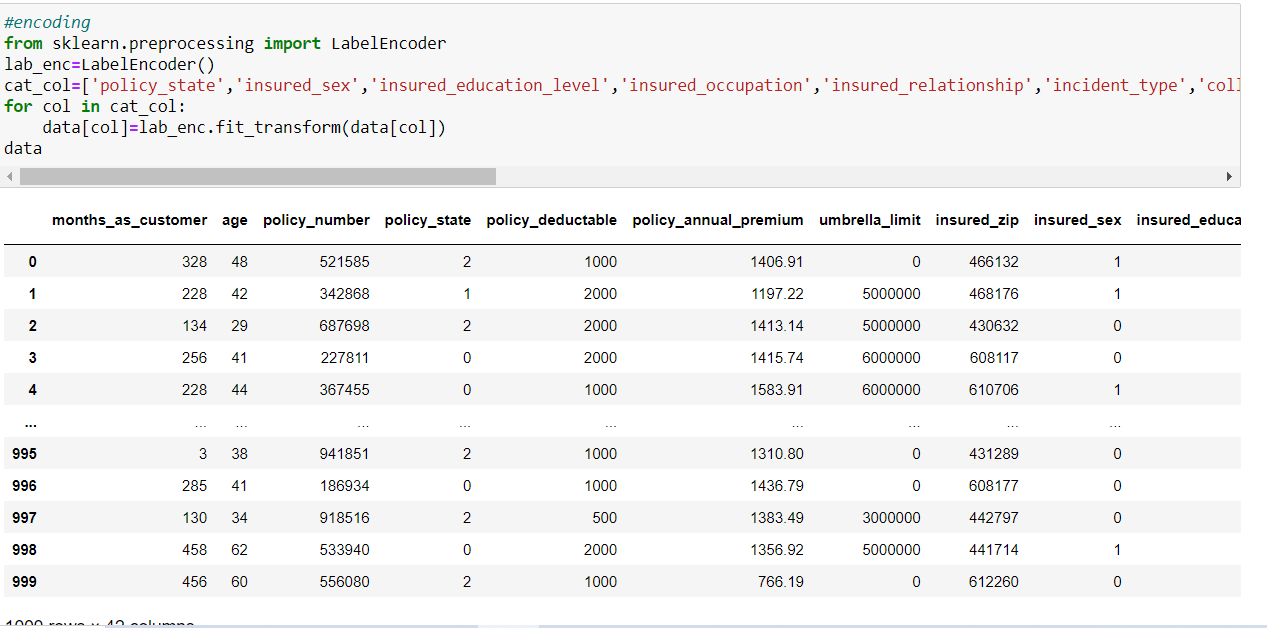
In machine learning, we usually deal with datasets with categorical columns in one or more in number. These labels can be in the form of words or numbers. or in human readable form, that type of data need to be converted to machine understandable form.

In our data there are columns with categorical values. The columns like incident\_severity, incident\_state, incident\_type, insured\_hobbies, authorities\_contacted, incident\_city, police\_report\_available, auto\_make, collision\_type, auto\_model, insured\_occupation, insured\_education\_level, property\_damage, insured\_relationship, policy\_state, insured\_sex, fraud\_reported. These columns can be treated using encoding techniques.



**Label Encoding**

Label Encoder encodes the labels into numeric form, so as to convert it into machine readable form we use encoding techniques. It is an important pre-processing step for the structured dataset in supervised learning.

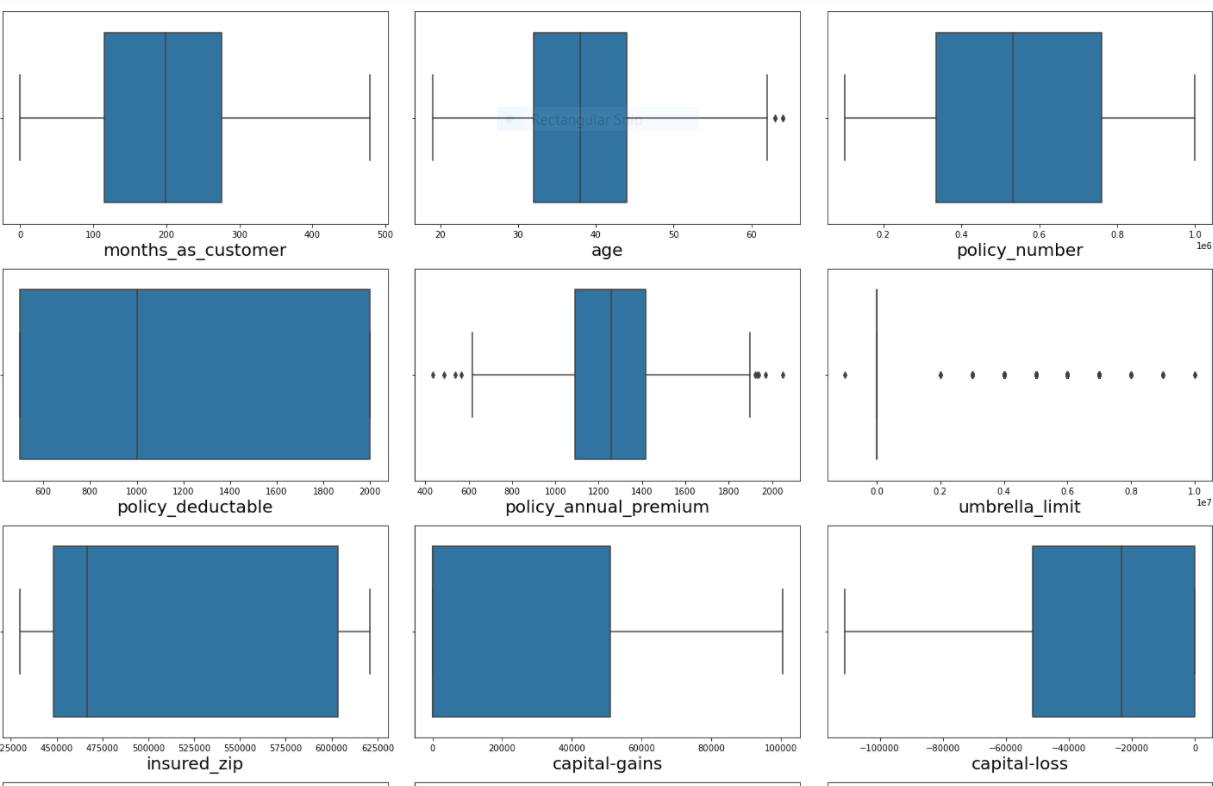


**Outliers**

Outliers are the data points, in an observation that lies an abnormal distance from other values in a random sample. Sometimes, Outliers also might be the errors which we want to exclude from the data during our analysis.

**Methods to remove outliers:**

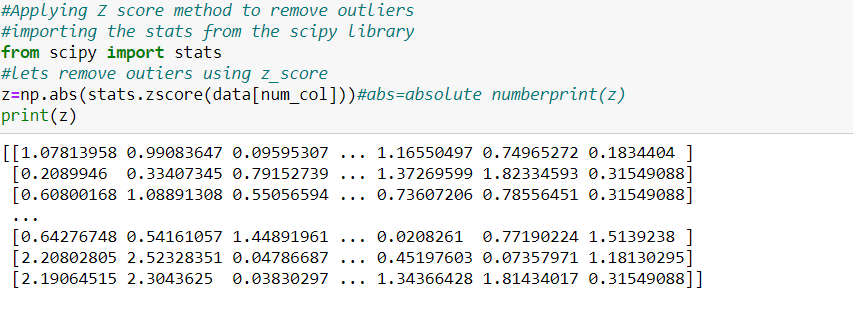
In order to check the outliers in the dataset, I have plotted Box plots to visualise the outliers as shown in screenshot. We can plot the box plots only on numerical columns.



After plotting box plots, I came to know that some of the columns having outliers those are age, policy\_annual\_premium, umbrella\_limit, property\_claim. So that outliers can be removed by using many techniques but here I have used Z-Score method which Is imported from scipy.stats to remove outliers.

**Z-score**

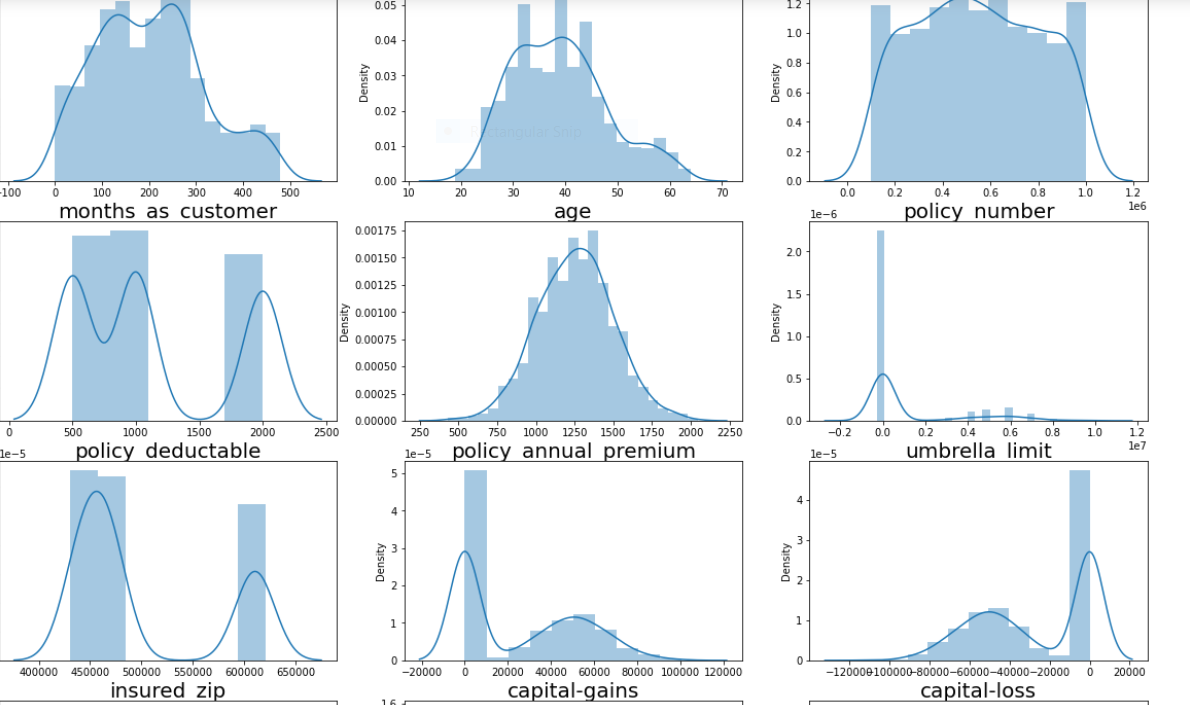
Z score is a statistical method. It is also called as standard score. This score helps to understand if a data value is greater or smaller than mean and how far away it is from the mean.



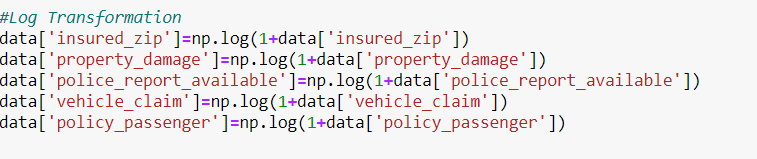
After getting the data points with outliers we will be dropping the entire rows having outliers. So, the shape of the data set changes as the rows dropped.

**Skewness**

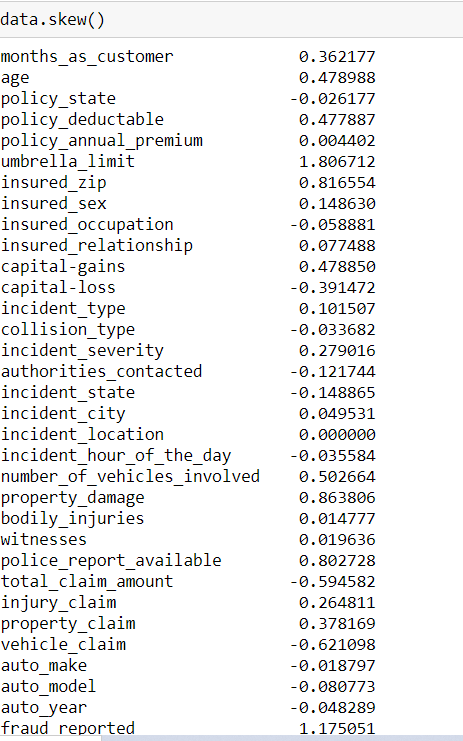
The skewness of the columns will be checked by plotting the distribution plot as shown below



In order to remove skewness from the data, I have applied log transformation to remove skewness.



Even I can check the skewness of the columns by writing code as data.skew()



**Standardization:**

Standard Scaler follows Standard Normal Distribution, which makes mean = 0 and scales the data to unit variance. Since our data in the columns are of different scales with which our models cannot give good patterns and performance on unequal scales, in order to bring all the values to standard form we use standard scaler.



**Balancing our imbalanced data**

 Imbalance means that the number of data points available for different classes are different. We commonly encounter with this type of problems in most of the classification problems, In order to make the classes balance we use many techniques one among is SMOTE.

SMOTE is a popular over sampling technique, which aims to balance class distribution by randomly increasing minority class samples by replicating them.



**Building machine learning models**

Apart from achieving highly accurate models, one of the most important aspect of building machine learning models is to obtain actionable insights and be able to select a subset of important features from the vast number.

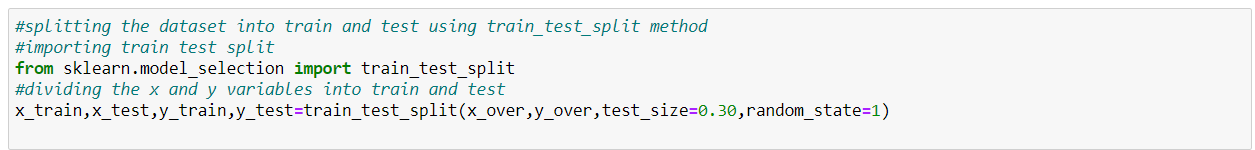
In order to build Models, we should be in a position to find out the type of problem, either the problem is a Regression problem or classification problem.

Since our label has two classes it comes under Binary classification. Once the problem is known we will be splitting the dataset.

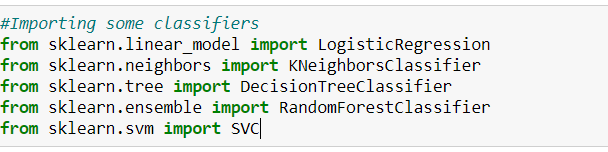
But before fitting our dataset to its model first we have to separate the predictor variable and the target variable, then we pass this variable to the train test split method to create a random test and train subset.

In train test split method will be passing the values of x and y along with them we will be dividing the train data and test data into some portions as the train data is divided into 70% and test data is divided into 30%. Which we pass as test size by passing 0.30. and assigning value for random\_state.

random\_state is used for reproducing the same value every time as it is run. If we do not use random\_state in train\_test\_split, every time we make the split and different values will be taken by the train data and test data so if we get any issue we cannot debug so for train\_test\_split we have to give a random\_state value.



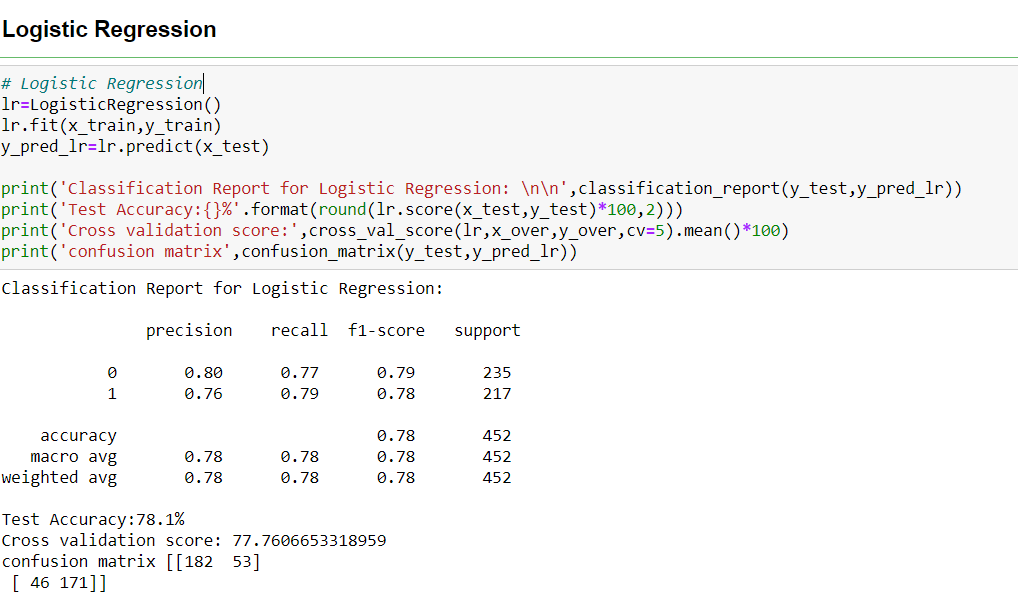
After train\_test\_split I have imported 5 models from sklearn to predict the fraud\_reported.



The 5 models for fraud prediction are:

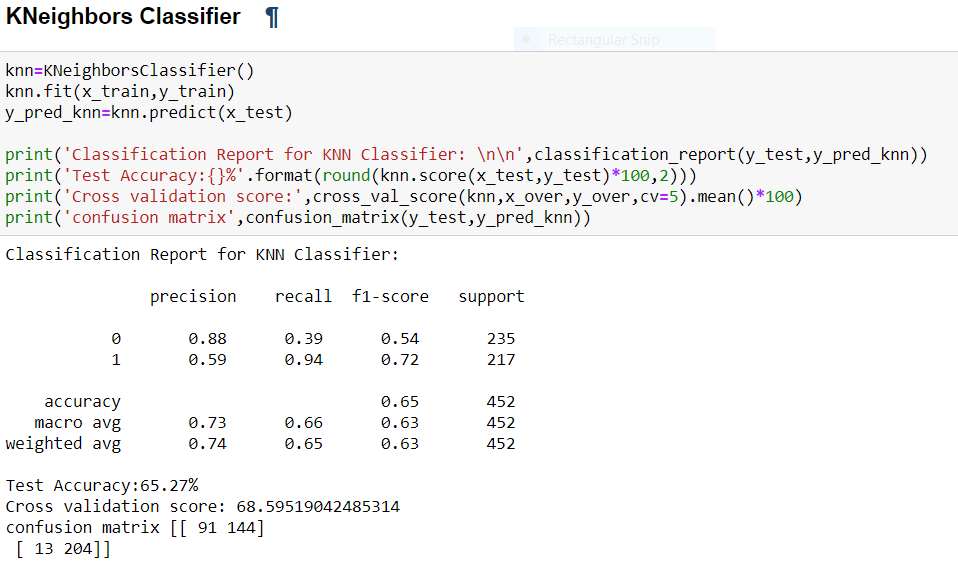
**Logistic Regression**

Logistic regression is a classification algorithm used to predict the probability of a target variable. In basic form it uses a logistic function to model a binary dependent variable.

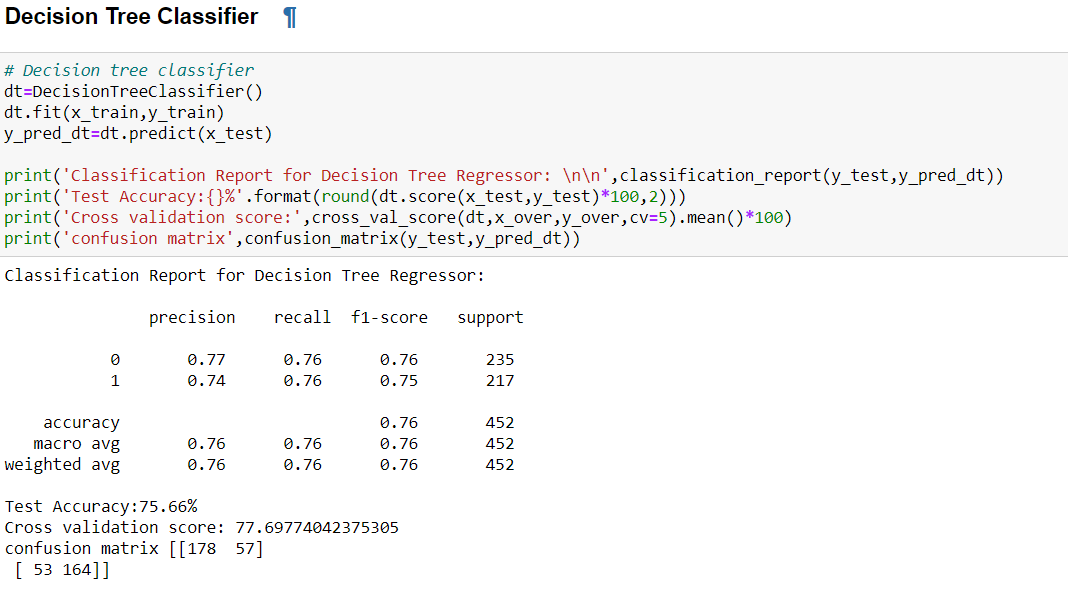


**KNeighbors Classifier**

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

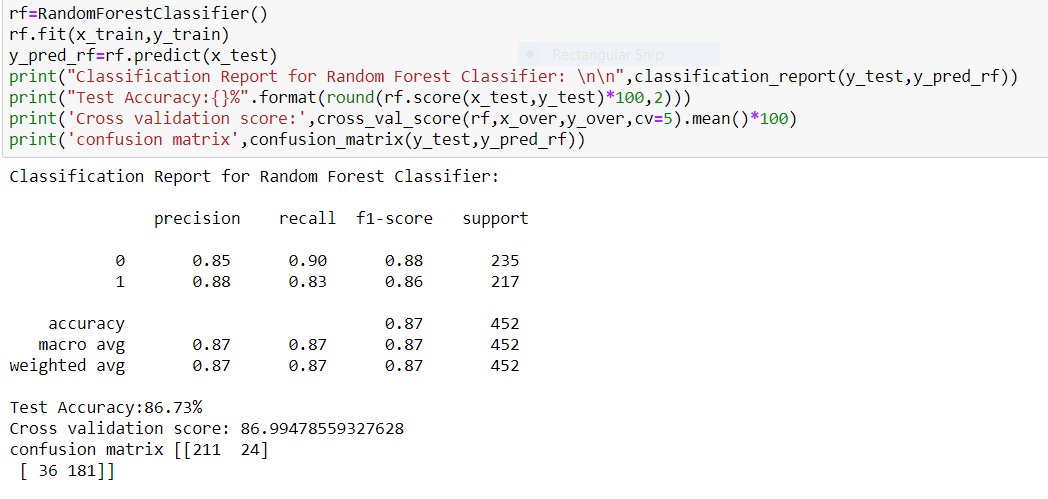
**Decision Tree Classifier**

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node. We make some assumptions while implementing the Decision-Tree algorithm.



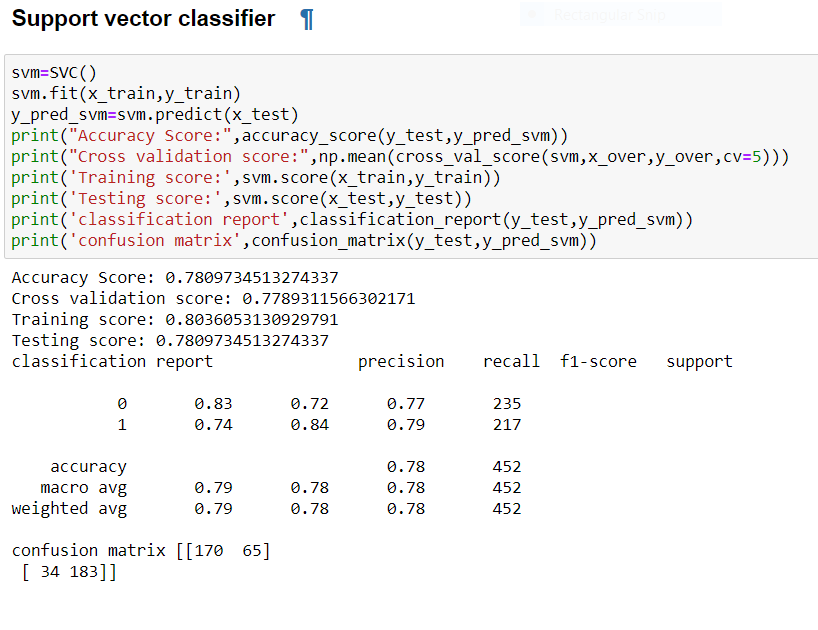
**Random Forest Classifier**

Random forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. It creates a set of decision trees from a randomly selected subset of the training set.

. 

**Support Vector Classifier**

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyse the data for classification and regression analysis.

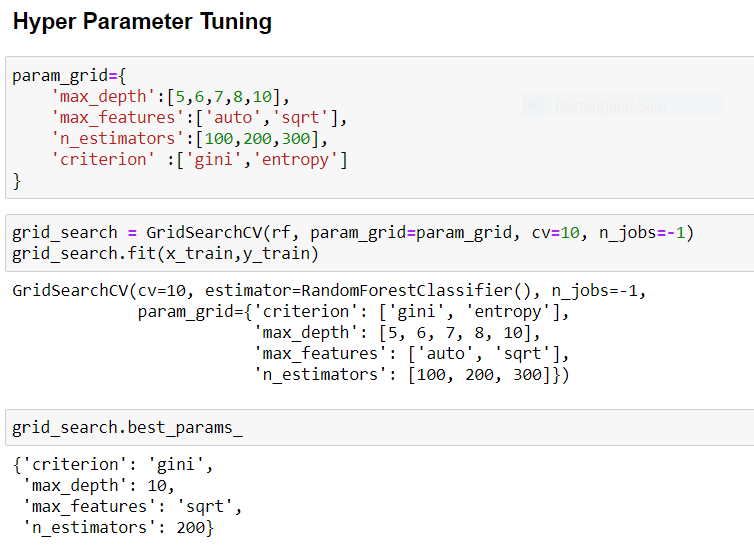
**Conclusion from models**

Random Forest Classifier is the best model to detect fraud with an accuracy score of 87%. Here our model predicts 204 fraud cases and 248 non fraudulent cases, among them 208 are true positive cases and 27 false positive cases and 184 are true negative cases and 33 are false negative cases. The F1 score for Random Forest Classifier is 87% F1 score is the combination of precision and Recall.

**Hyper parameter tuning**

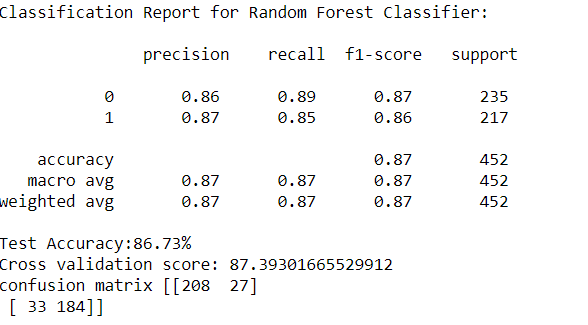
Hyper parameter tuning optimization is anessential aspect of machine learning process which can really make a model succeed in meeting desired metric value. model has its own sets of parameters that need to be tuned to get optimal output. Setting parameters maximizes the performance of the model on a validation set. Machine learning models frequently require to fine-tuning of model hyper parameters. So, we use Grid Search CV to tune the parameters.

**Grid Search CV**

Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time- 

**Evaluation Metrics**

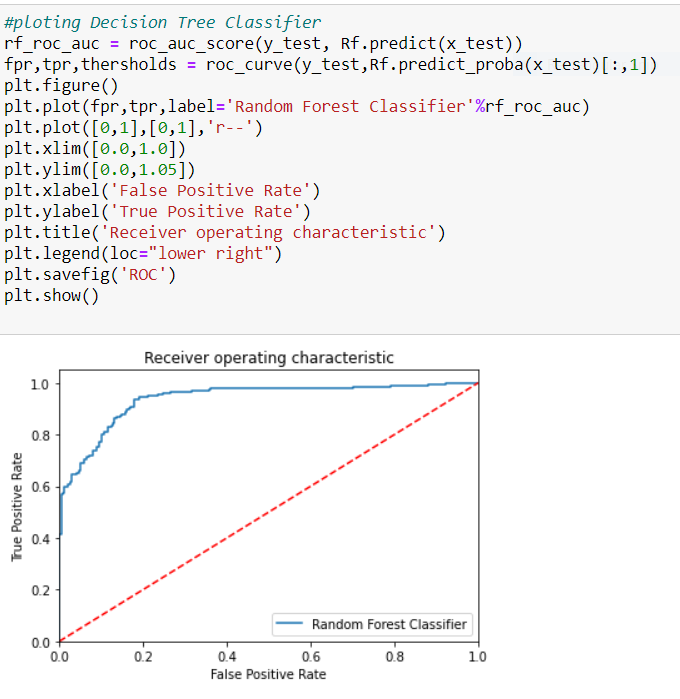
The evaluation metrics are important for a model to decide weather a model is good or bad which will be decided on the values of metrics such as accuracy score, cross validation score, precision, recall, F1 score, sensitivity, ROC AUC score, confusion matrix. The values for evaluation are shown below.



**ROC curve**

It is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher is the AUC score, the better is the model performance. The score must be more than 90% then only we say that, the model is performing well.

The ROC curve is plotted between the True Positive Rate (TPR) and the False Positive Rate (FPR) where TPR is on the y-axis and FPR is on the x-axis.



**Remarks**

Choosing the right machine learning method depends on the problem type, size of a dataset, resources, etc. A good practice is to use several models to both streamline assessment and achieve higher accuracy.

As of today, antifraud systems should meet the following standards:

* Detect fraud in real-time
* Improve data credibility
* Analyse user behaviour
* Uncover hidden correlations

While these qualities can be offered by machine learning algorithms, even they require carefully prepared datasets for training and still need some features like checking legal limitations for cash transactions. Statistical models are more stable when data sets are larger. It also generalises better as it takes a bigger proportion of the actual population. So, If the dataset is big and the smooth data is provided, we can achieve better patterns and high performance of the models.