# **Auto Insurance Fraud Detection**



Insurance fraud is a significant and costly problem for both policyholders and insurance companies in all sectors of the insurance industry. In recent years, fraud detection has attracted a great deal of concern and attention. The Oxford English Dictionary defines fraud as “wrongful or criminal deception intended to result in financial or personal gain”. Fraud occurs in a wide variety of forms and is ever changing as new technologies and new economic and social systems provide new opportunities for fraudulent activity. The total extent of business losses due to fraudulent activities is difficult to define.

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, 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, 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.

Background of Insurance Fraud

Auto insurance fraud ranges from misrepresenting facts on insurance applications and inflating insurance claims to staging accidents and submitting claim forms for injuries or damage that never occurred, to false reports of stolen vehicles.

Fraud accounted for between 15 percent and 17 percent of total claims payments for auto insurance bodily injury in 2012, according to an Insurance Research Council (IRC) study. The [study](https://www.insurance-research.org/sites/default/files/downloads/IRC%20Fraud%20News%20Release.pdf) estimated that between $5.6 billion and $7.7 billion was fraudulently added to paid claims for auto insurance bodily injury payments in 2012, compared with a range of $4.3 billion to $5.8 billion in 2002.

[No-fault auto insurance](https://www.iii.org/issues_updates/no-fault-auto-insurance.html) is a system that allows policyholders to recover financial losses from their own insurance company, regardless of who was at fault in a motor vehicle accident. However, in many no-fault states, unscrupulous medical providers, attorneys and others perpetrate fraud by padding costs associated with a legitimate claim, such as billing an insurer for a medical procedure that was not performed.

**Salvage fraud**: Another common auto fraud involves vehicles damaged by storm flooding that later appear in used car lots and auction sales. In some states, vehicles that have been flooded bear the words “salvage only” on their titles, usually after damage to the vehicle has reached about 75 percent of its value. Unscrupulous sellers may switch or clone manufacturers’ serial number plates and put them on a flooded vehicle that has been repaired. They may also resell a car that has a salvage title in a state that has more lax title standards. This practice is called “title washing.”

Standardized state rules for titling vehicles are necessary to combat salvage fraud. In recent years, some states in the hurricane-prone parts of the United States have adopted rules that require that the words “flood vehicle” be included on the titles of vehicles that have been water damaged and rebuilt. Before such a vehicle can be sold, the buyer must be notified in writing of the vehicle’s past flood damage. However, if one state in the region does not have such strict laws it can become a dumping ground for undeclared flooded vehicles

# **Data Preprocessing**

Input Features:

months\_as\_customer

age

policy\_number

policy\_bind\_date

policy\_state

policy\_csl

policy\_deductable

policy\_annual\_premium

umbrella\_limit

insured\_zip

insured\_sex

insured\_education\_level

insured\_occupation

insured\_hobbies

insured\_relationship

capital-gains

capital-loss

incident\_date

incident\_type

collision\_type

incident\_severity

authorities\_contacted

incident\_state

incident\_city

incident\_location

incident\_hour\_of\_the\_day

number\_of\_vehicles\_involved

property\_damage

bodily\_injuries

witnesses

police\_report\_available

total\_claim\_amount

injury\_claim

property\_claim

vehicle\_claim

auto\_make

auto\_model

auto\_year

\_c39

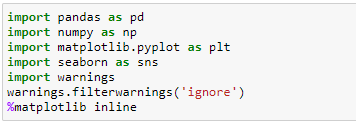
Output Variable:

fraud\_reported

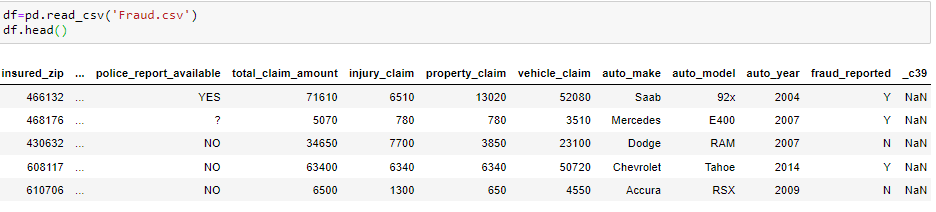
This current dataset as n=1000 samples. It means the data set consists of 1000 auto incidents and 40 features. It is not stated if this data is from multiple insurance companies or just one company. However, throughout the report, “the insurance company” will be used to refer to the origin of this data.

Importing libraries and dataset: -

Here we are importing Pandas, NumPy. We are also importing Matplotlib and Seaborn for data visualization.



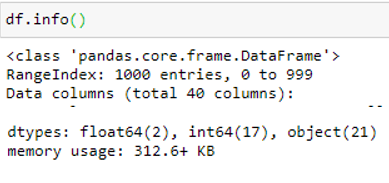
Now let’s import the dataset. Here the dataset is named as Fraud.csv file.



Let’s check the shape of the dataset.

We can observe that there is 1000 rows and 40 columns in the dataset

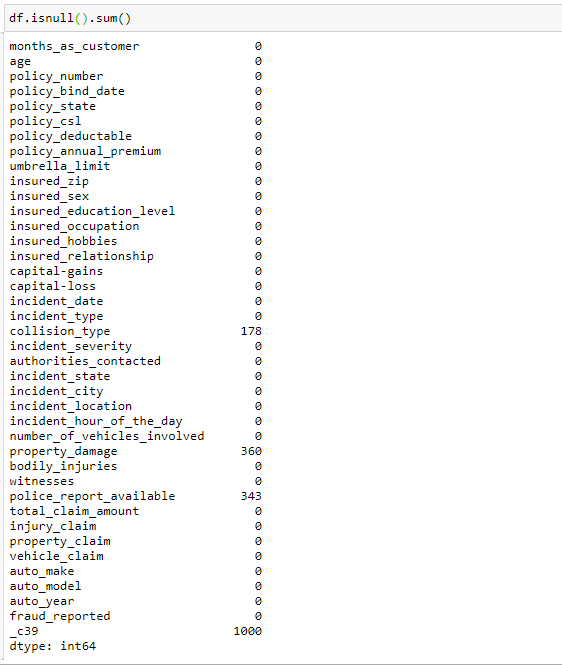




As shown above we have 19 numerical features and 21 non-numerical features.

In the dataset we can identify some missing values denoted by ‘?’. We shall replace it with Nan. So that it will be easy for further computation.

Now we shall check for null values/missing values in the dataset.

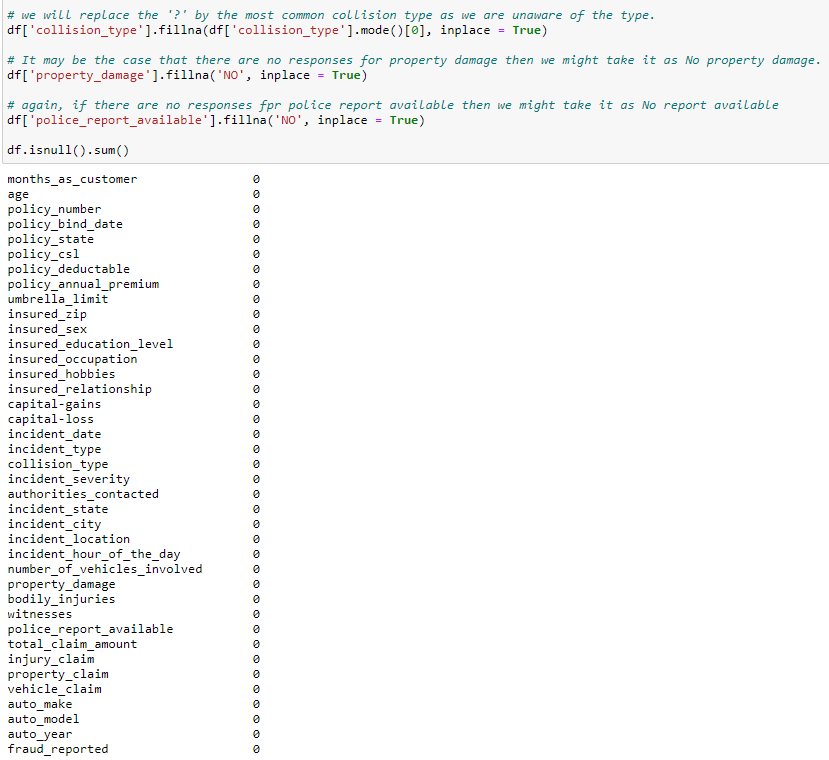


From the above code we can observe that values are missing in four features. Collision\_type has 178 values missing, property damage has 360 values missing, police\_report\_available has 343 missing values.

All the features with missing values are categorical / object datatype. So, let’s replace the missing values with mode of that feature i.e., most repeated value of that feature in that dataset.

As for \_c39, there are 1000 missing values. If u recall, the dataset consists of 1000 rows and 40 columns. So, when there are 1000 missing values, it means all the values of that particular feature is missing from the dataset. Hence we can drop that particular column.



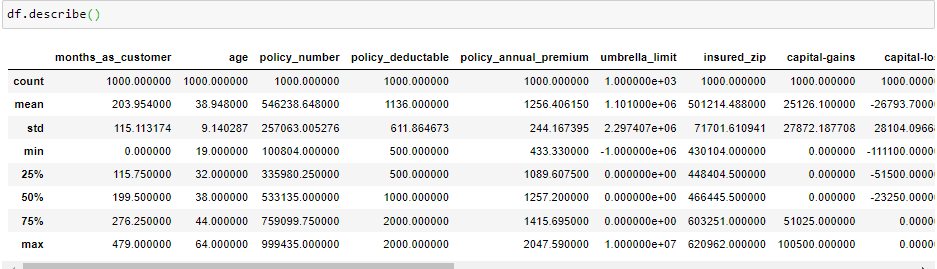


## **Exploratory Data Analysis**

## Features distribution:

## Standard deviations are important here because the shape of a normal curve is determined by its mean and standard deviation. The mean tells you where the middle, highest part of the curve should go.

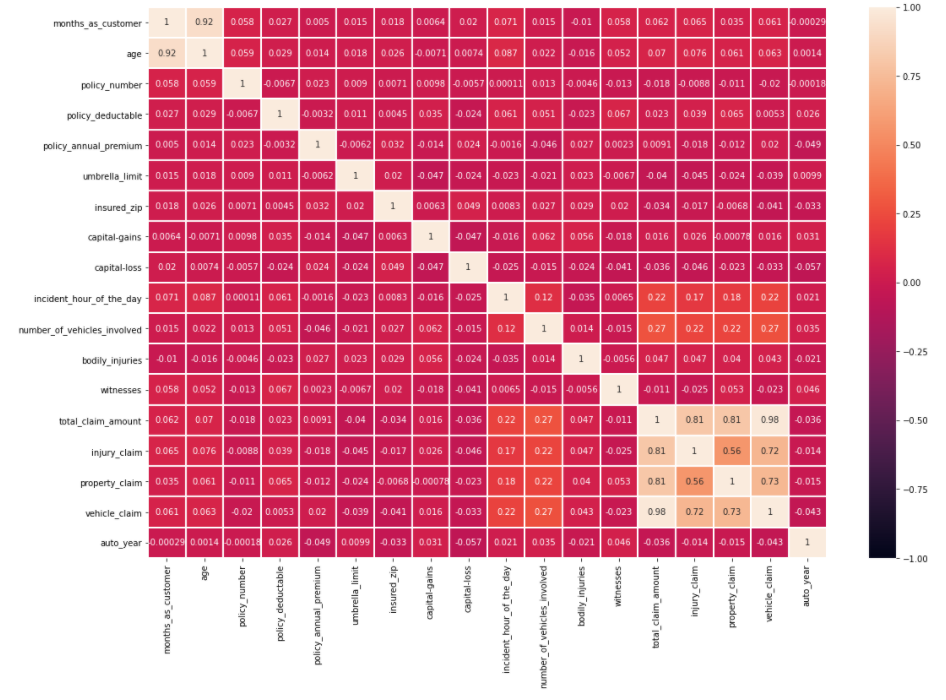
At first glance, there aren’t any outliers in the data. No data point is disconnected from distribution or too far from the mean value. To confirm that we would need to plot the data.



Correlation: -

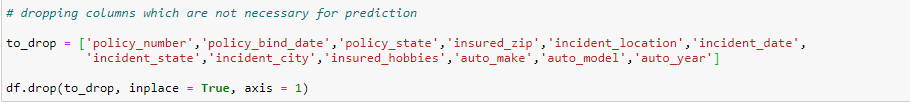
Correlation coefficients are used to measure the strength of the relationship between two variables. This measures the strength and direction of a linear relationship between two variables. Values always range between -1 (strong negative relationship) and +1 (strong positive relationship).

We shall check the correlation of the variables. From the heatmap we can make some observations as such.

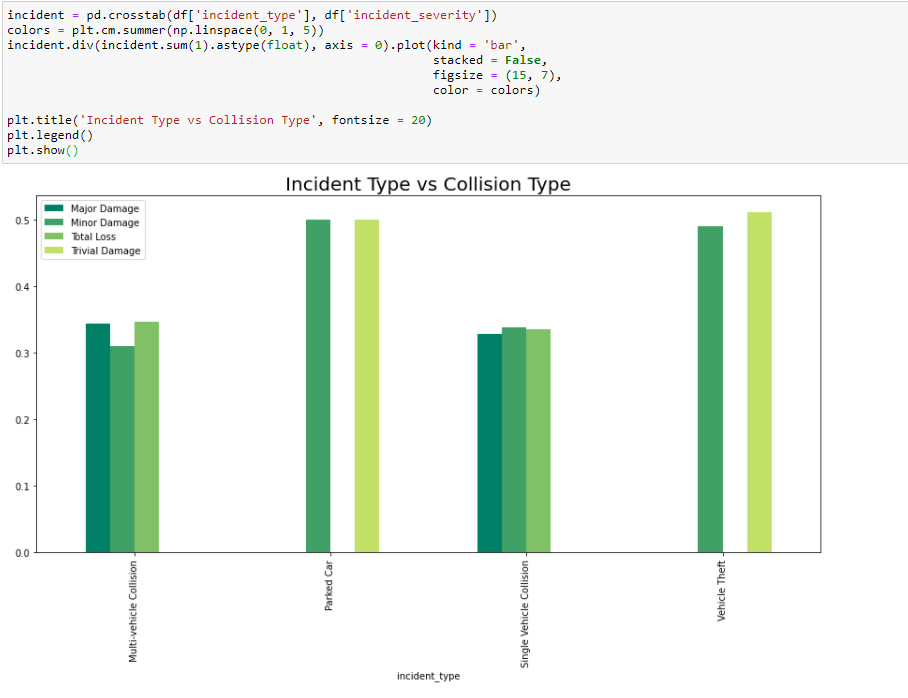


1. We can observe that month as customer and age had a correlation of 0.92. Probably because drivers buy auto insurance when they own a car and this time measure only increases with age.
2. Total\_claim\_amount is highly correlated with injury\_claim, property\_claim and vechicle\_claim

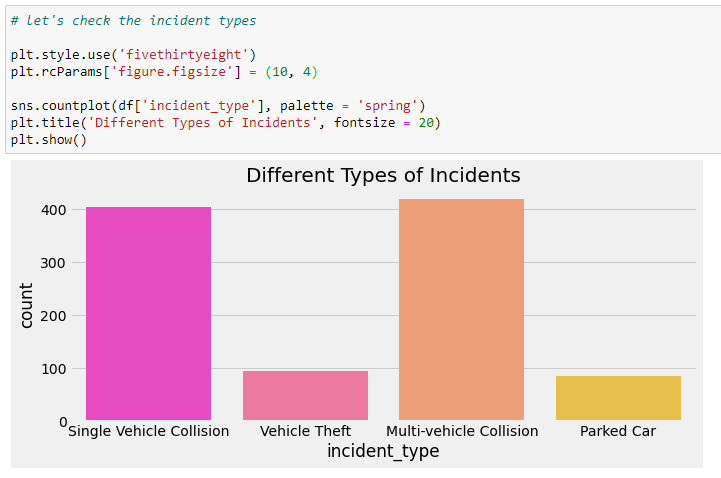
Let’s drop unnecessary columns.



DataVisualization: -



From the above barplot we can notice that Parked Car has more major and trivial damages to it. Except that we have more Vehicle thefts.



Multi-vehicle and Single vehicle collision has occurred more than vehicle theft and parked car collision.



Insured\_sex is almost balanced data. Here women have got auto-insured comparatively more than men.

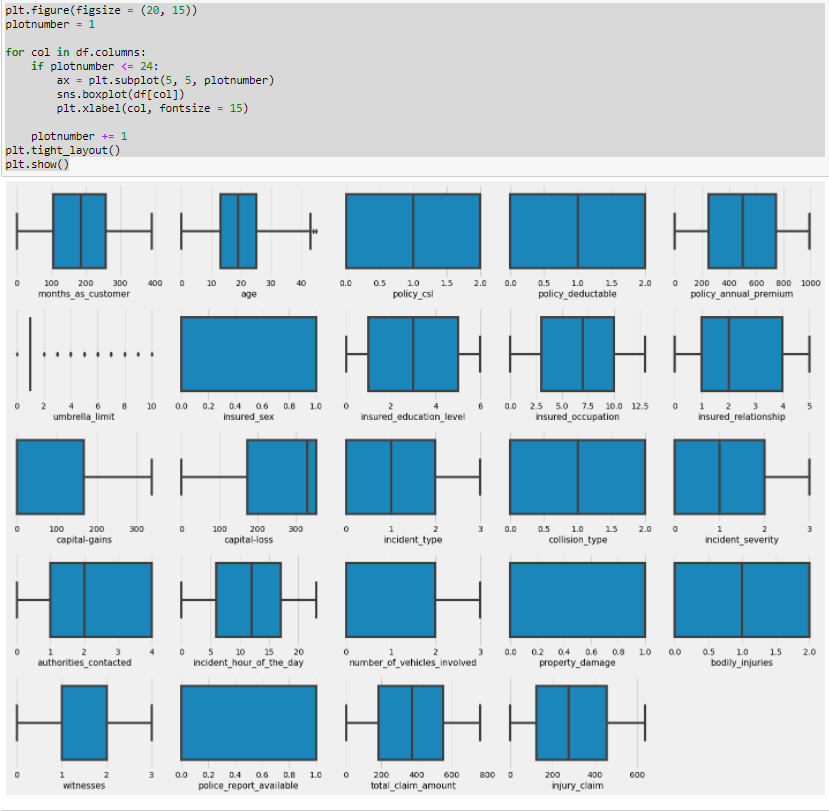
Since most of the variables are categorical let’s change it into numerical using Label encoder.

**Label Encoding** refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.



All the features datatype is now integer. Let’s check for the outliers by plotting box plot of all features.

Often there exist data objects that do not comply with the general behaviour or model of the data. Such data objects, which are grossly different from or inconsistent with the remaining set of data, are called outliers. Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This, however, could result in the loss of important hidden information. In other words, the outliers may be of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity



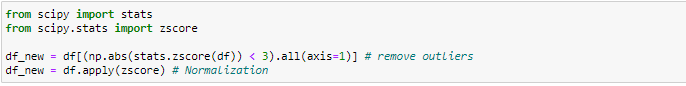
We can observe that there are no much outliers expect in age and umbrella limit. We shall remove the outliers using zscore.

**A z-score (also called a**standard score**) gives you an idea of how far from the**[mean](https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-median-mode/)**a data point is.** But more technically it’s a measure of how many [standard deviations](https://www.statisticshowto.com/probability-and-statistics/standard-deviation/) below or above the [population mean](https://www.statisticshowto.com/population-mean/)a [raw score](https://www.statisticshowto.com/raw-score/) is.

A z-score can be placed on a [**normal distribution**](https://www.statisticshowto.com/probability-and-statistics/normal-distributions/) curve. Z-scores range from -3 standard deviations (which would fall to the far left of the normal distribution curve) up to +3 standard deviations (which would fall to the far right of the normal distribution curve). In order to use a z-score, you need to know the [mean](https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-median-mode/#mean) μ and also the population standard deviation σ.

The basic z score formula for a [sample](https://www.statisticshowto.com/sample/)is:

**z = (x – μ) / σ**



## **Modelling**

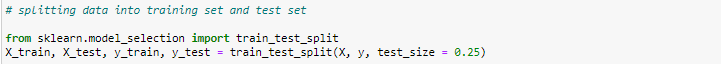
Let’s split the data. Here X is independent variables used to predict dependent variables y.



Here fraud reported is dependent variable and rest of the features are independent variables.

Train-Test split: -

**Train-Test split** is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. By default, Sklearn Train-Test split will make random partitions for the two subsets. However, you can also specify a random state for the operation.

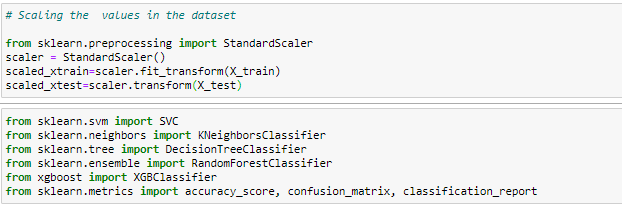


The test size=0.25 inside the function indicates the percentage of the data that should be held over for testing.

Let’s scale the data using StandardScaler library.

StandardScaler removes the mean and scales each feature/variable to unit variance. This operation is performed feature-wise in an independent way. StandardScaler can be influenced by outliers (if they exist in the dataset) since it involves the estimation of the empirical mean and standard deviation of each feature.

We shall also import few Machine learning models.



Here we are scaling independent variables of both train and test set.

Five different classifiers are used in this project:

-SVC

-K-nearest neighbours

-Random Forest

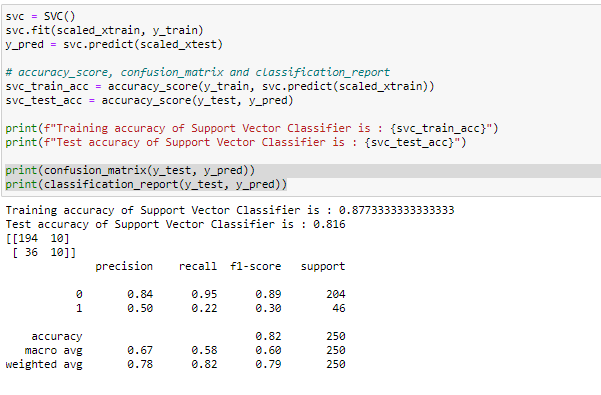
-XGBoost

-Decision Tree

Classification is defined as the act or process of putting things into groups based on ways that they are alike. Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown.

**SVC**

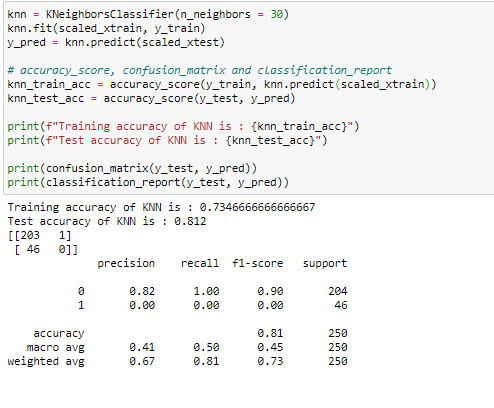
The objective of a Linear SVC (Support Vector Classifier) is to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data.



We can see that training accuracy is 87% and test accuracy is 81%.

**KNN**

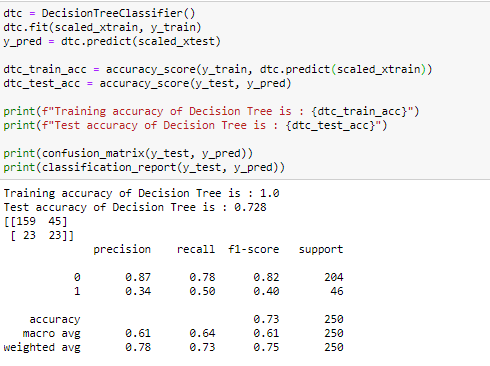
K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K-NN algorithm.



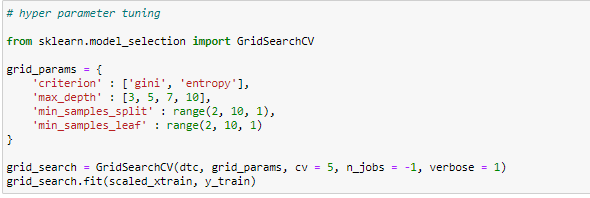
Here training accuracy is 73%, test accuracy is 81%.

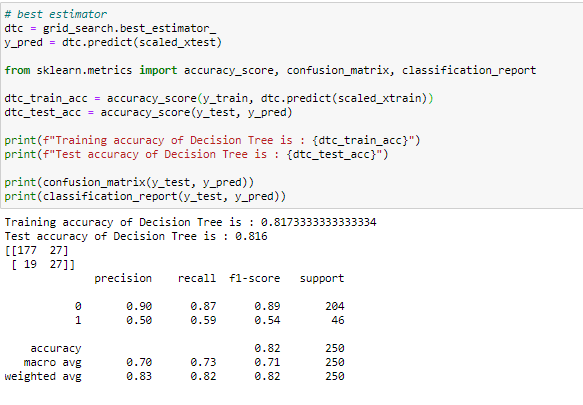
**Decision Tree**

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem. The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.



In decision tree, training accuracy is 100 % whereas testing accuracy after applying the model is 72%. We shall tune the parameters using GridSearchCV and predict the outcome.

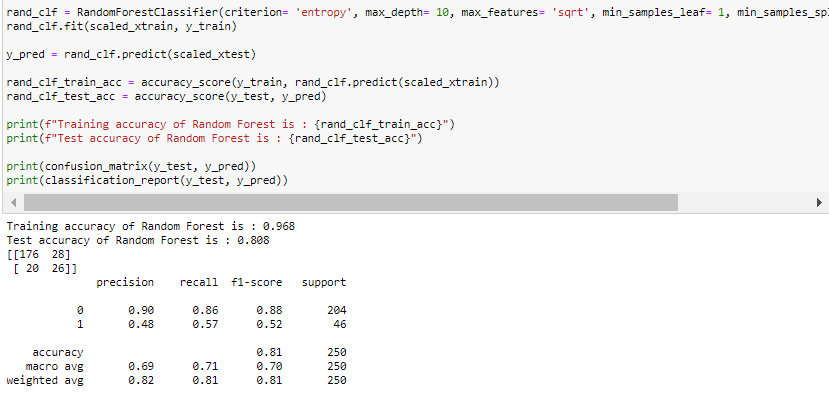




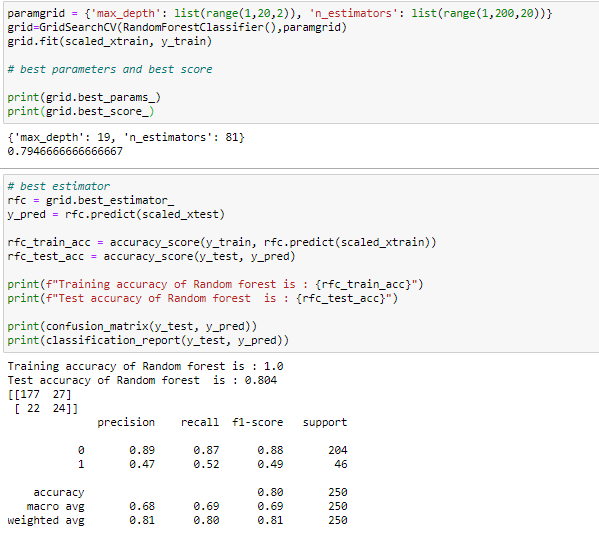
After hyper parameter tuning we can see that accuracy is 81% now.

**Random Forest**

Random forests (RF) are basically a bag containing n Decision Trees (DT) having a different set of hyper-parameters and trained on different subsets of data. Let’s say I have 100 decision trees in my Random Forest. These decision trees have a different set of hyper-parameters and a different subset of training data, so the decision or the prediction given by these trees can vary a lot. Let’s consider that I have somehow trained all these 100 trees with their respective subset of data. Now I will ask all the hundred trees in my bag that what is their prediction on my test data. Now we need to take only one decision on one example or one test data, we do it by taking a simple vote. We go with what the majority of the trees have predicted for that example.



Training accuracy is 97%, Testing accuracy is 80%. Let’s tune the parameters.



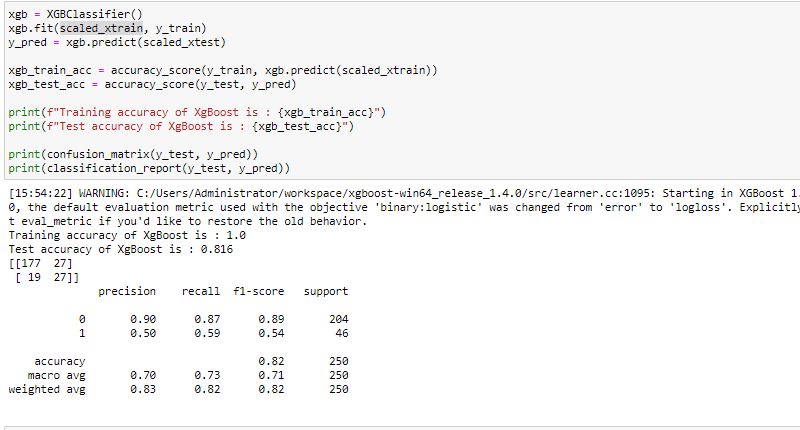
We can see that testing accuracy is the same 80%.

**XGBoost**

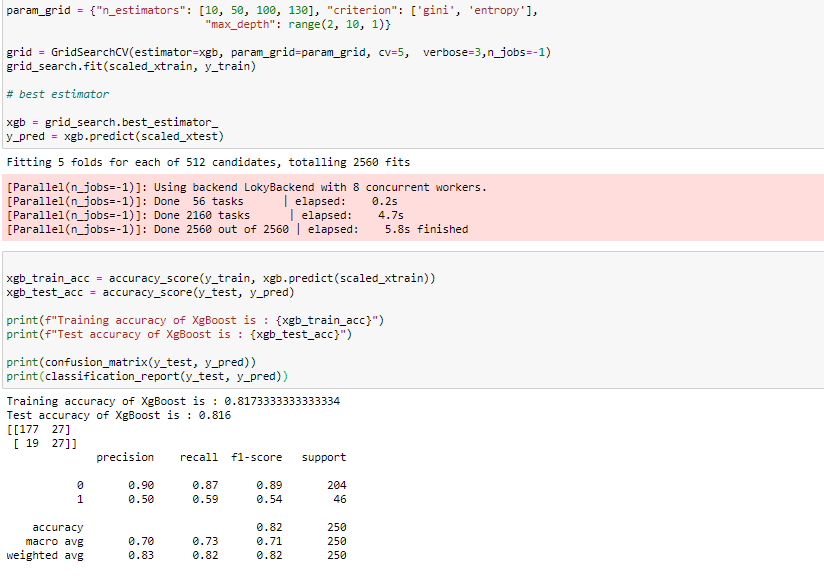
It is also known as “Extreme Gradient Boosting”. It carries out the gradient boosting decision tree algorithm. It has several different names like gradient boosting, gradient boosting machine, etc.

Boosting is nothing but ensemble techniques where previous model errors are resolved in the new models. These models are added straight until no other improvement is seen. One of the best examples of

such an algorithm is the AdaBoost algorithm. Gradient boosting is a method where the new models are created that computes the error in the previous model and then leftover is added to make the final prediction



XGBoost testing accuracy is 81%. Let’s check after hyper parameter tuning.



Accuracy is the same even after tuning the parameters.

**Conclusion**

We studied the existing fraud detection systems. To predict and present fraud we used KNN classifier and Decision Tree-Based algorithms. We looked at model performance metrics derived from the confusion matrix. Performance metrics such as accuracy, recall, and precision are derived from the confusion matrix. It is strong with respect to class skew, making it a reliable performance metric in many important fraud detection application areas.