**Insurance Claims- Fraud Detection**

**Problem Statement:**

**Business case:**

Fraud is amongst the biggest and one of the most well-known issues that insurers face. This Blog/Article sheds light on this particular topic and information about some of the automobile insurance company faces. Fallacious claiming are often extremely costly for every insurance company. Hence, it is vital to understand that which of the claims are correct and which of them are not. It’s not realizable or even feasible for any insurance firms to see all claims in person particularly since this can consume just too much of time and cost. Hence, Insurance fraud is a huge problem in the industry. It’s difficult to identify fraud claims. Machine Learning is a very unique tool which can help the Auto Insurance industry with this particular problem.

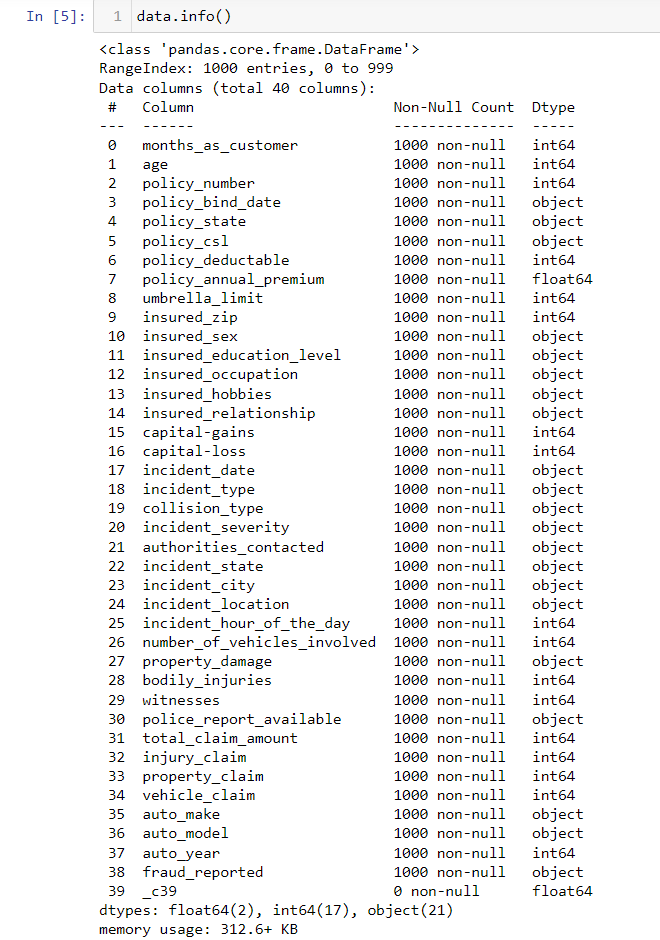
In this article, we are provided a dataset which has the all the details of the insurance policy including the customer details. It also has the details of the accident on the basis of which the claims have been made.

In this article, we will be working with some auto insurance data to demonstrate how we can create a predictive model that predicts if an insurance claim is fraudulent or not.

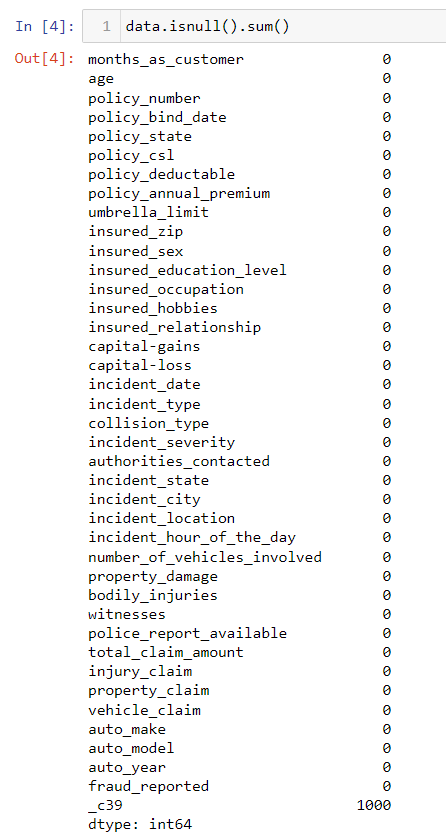
**Data Analysis:**

In this project, we’ve a dataset that has the main points of the policy together with the client details. It additionally has the main points of the accident on the premise of that the claims are created.

This dataset have 1000 Rows with 40 Columns which obviously is small but is workable amongst the most other datasets which have much more less data than we have already.

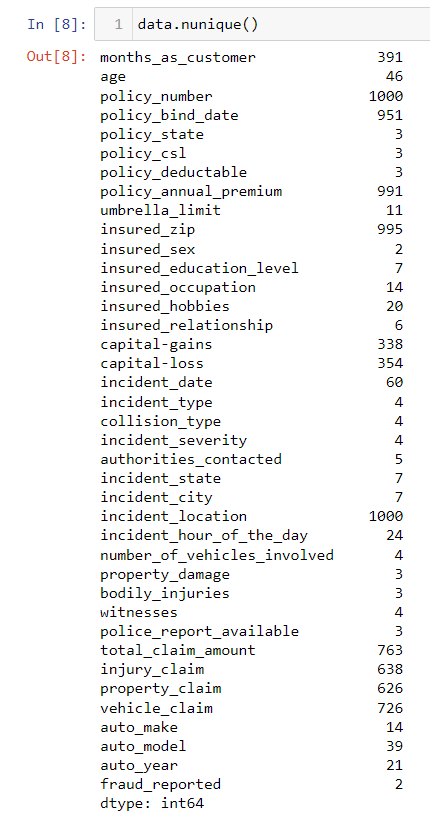
Dataset Descriptions

Also after seeing the dataset and viewing null values in in dataset we see that the data has one column i.e. (“\_c39”) with all of its rows as Null.



Null Values in Dataset

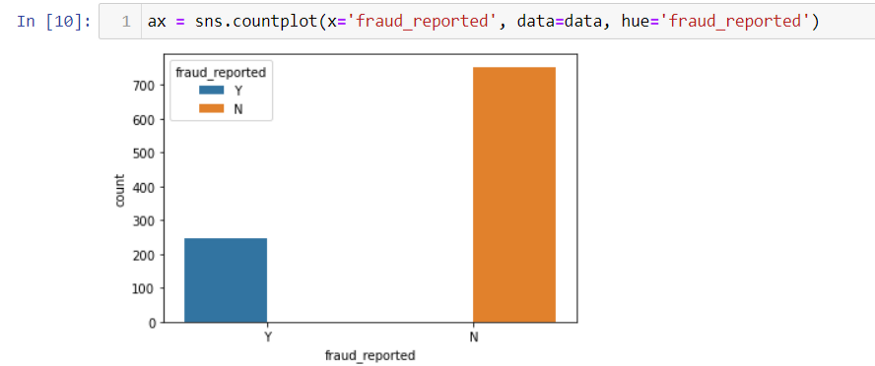
Also viewing number of unique values in the dataset we see multiple columns which will be useful in our prediction and there are some columns which might require Data Pre-Processing.



Unique objects in every Columns

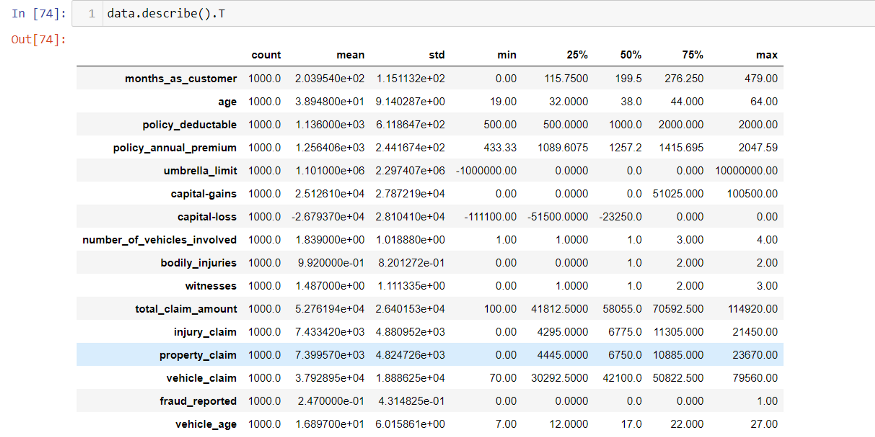
**Exploratory Data Analysis:**

We first did EDA of our Dependent Variable i.e. **‘fraud\_reported’** and we see that there are 247 frauds and 753 non-frauds.



As we can see we have here quite Imbalanced as nearly 75.3% of total data were Non-Fraudulent claims and only 24.7% were claimed as Fraud.

Now if we see the data description in the dataset we see some interesting facts in while observing,



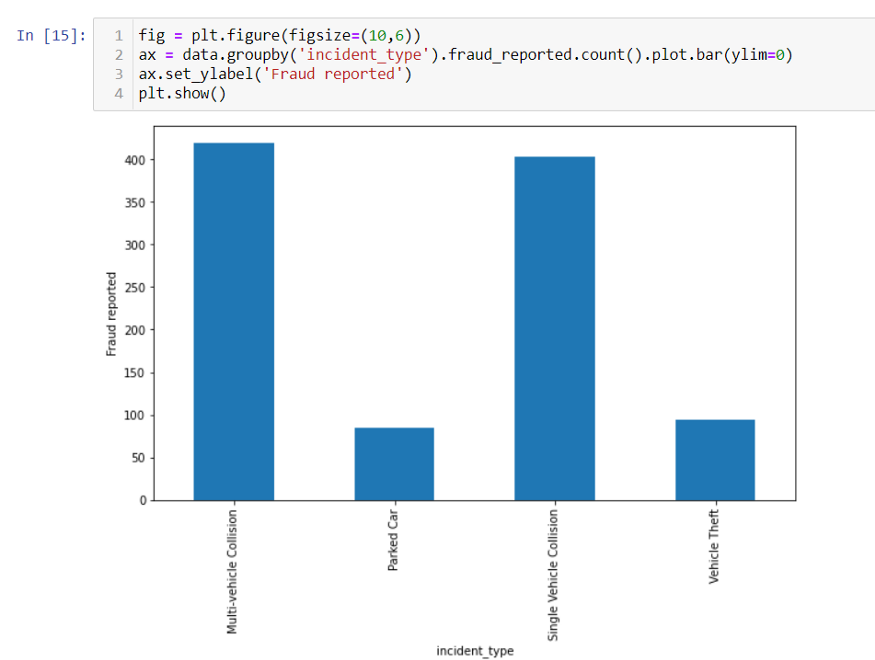
Data describtion in detail

We can observe from the above image demonstrating the table that:-

Some independent variable or columns such as **‘policy\_bind\_date’ , ‘incident\_date’ , ‘incident\_location’ and ‘insured\_zip’** contain very high number of level. We will drop these columns for our prediction.

If we focus some of the variable as visualization aspect and could concur some details regarding datasets :-

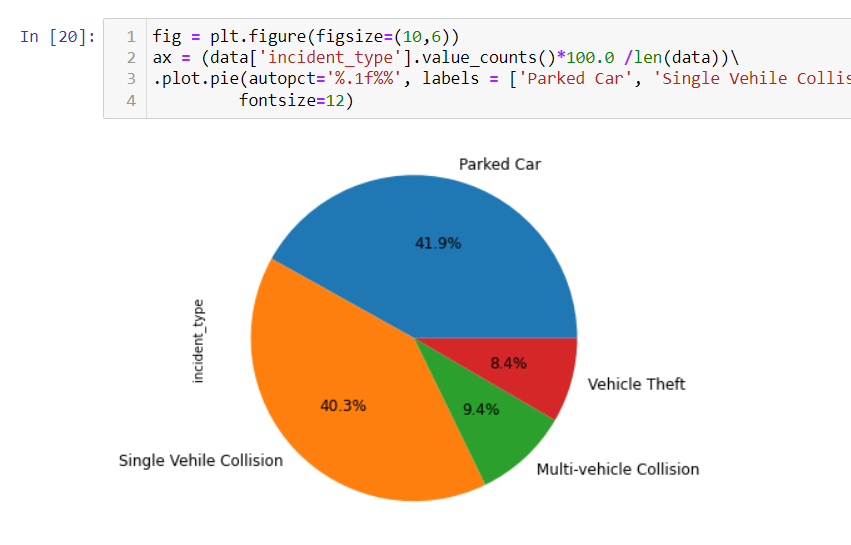
If we see details of **Incident types** in our dataset:-



incident\_type vs fraud reported

We can see that the most of the incidents which are reported are of **Multi-Vehicle Collision** and **Single-Vehicle Collision** and very less of Parked car and Vehicle Theft.

But if we see the total number of major Incidents in of totality,

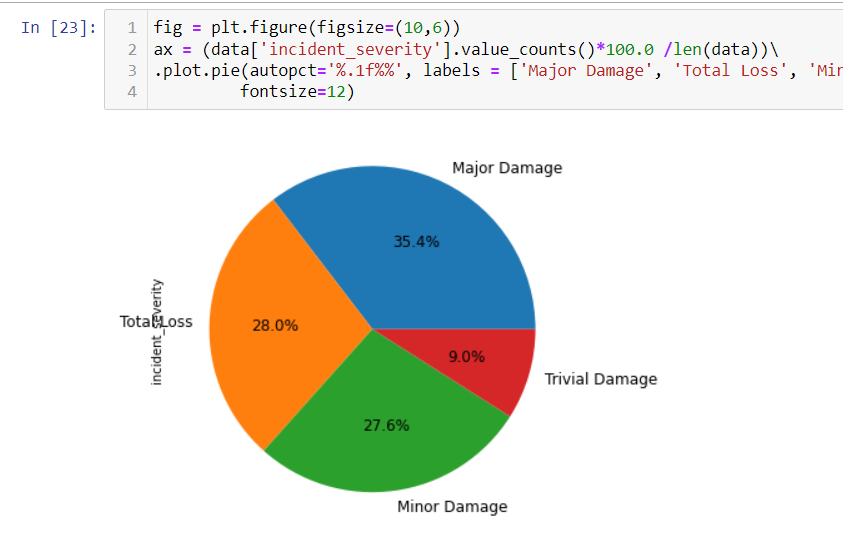


Number of Incident type

We can see that most incidents type are of Parked Car and of Single Vehicle Collision.

*If we pay any attention on the above two aspects we can see some anomaly as the major Incidents is of Parked Car but Parked car is is one of the least reported for fraud .*

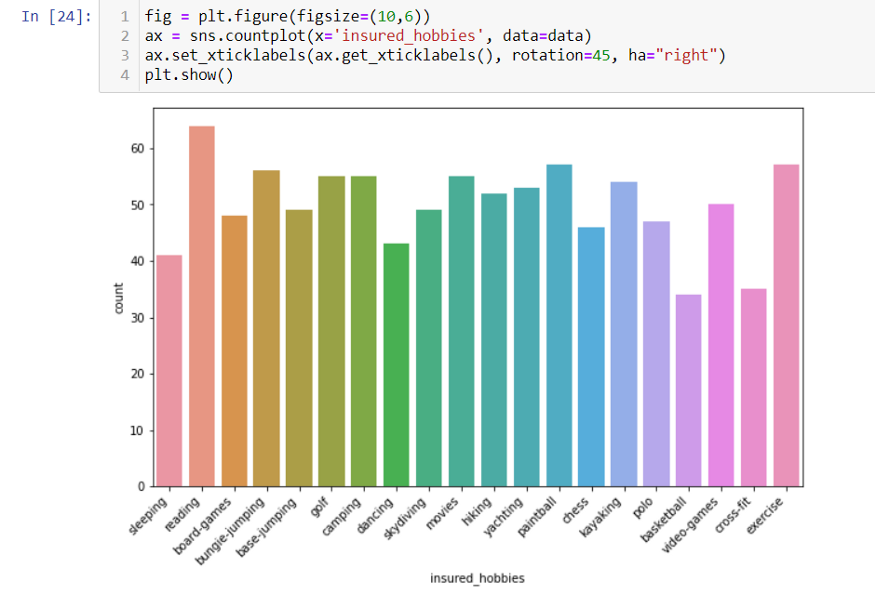
Now let’s take a look at **‘incident\_severity’** in the dataset



Incident severity in dataset

We can see that number of Major damage is the most and Trivial Damage is the lowest.

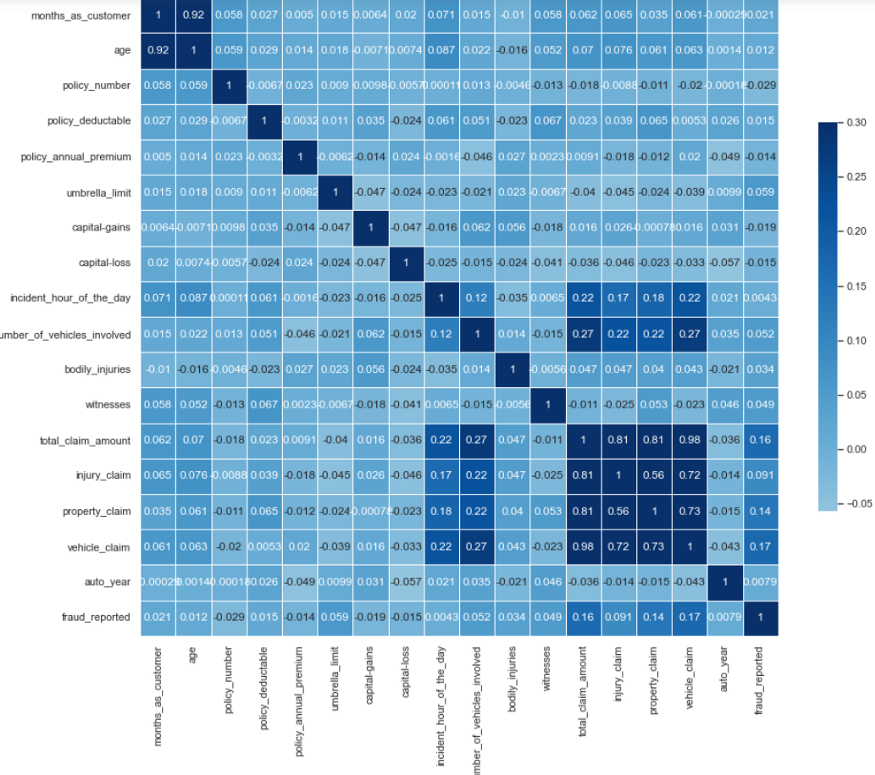
If we look into the the hobbies of the customers with aspect to fraud committed in the dataset



Hobbies of customers with respect to frauds committed

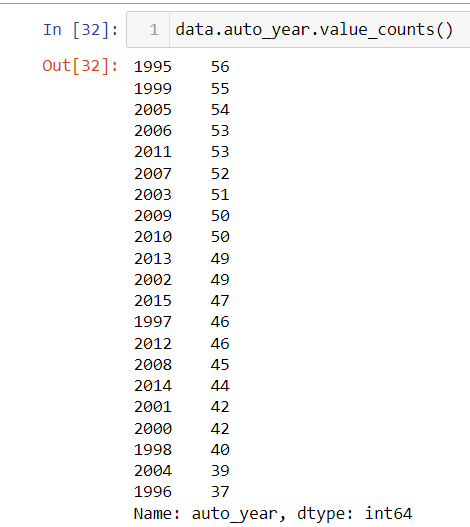
We would see that the likes of persons with hobbies with Reading have higher tendencies to fraud. Also some with hobbies of Basketball and Cross-Fit have low tendencies to fraud.

**Correlations among variables:**



Heatmap of Correlation

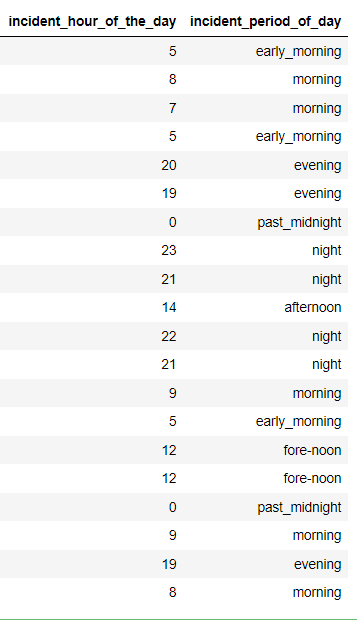
Spread of years to decide on further action.



Spread of Years

Column “auto\_year” contains 21 levels, and the number of records for each of these levels are quite significant considering data size is not as large as we would like . We could do some feature engineering using this variable considering , the year of manufacturing of vehicles indicates that the age of particular vehicle and might contain some valuable information for insurance premium or fraud is concerned.

Considering factoring in the various hours of the day.



Time Period of the Day

**Pre-processing Pipeline**

Data pre-processing plays a very important role in a data analytics process. It includes a wide range of tasks, from fixing faults to choosing the most pertinent features for the analysis stage.

So to perform Pre-processing we follow following process:-

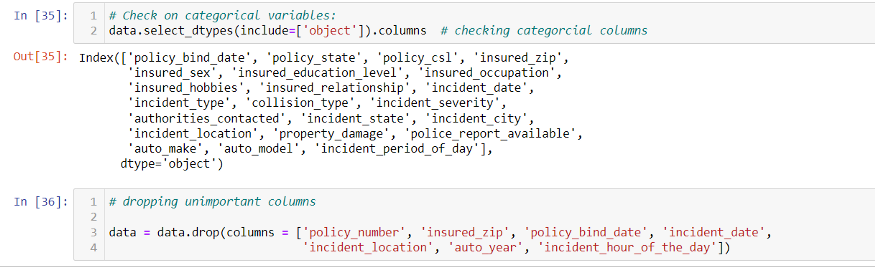
1- **Data Cleaning**

2- **Data integration**

3- **Data transformation**

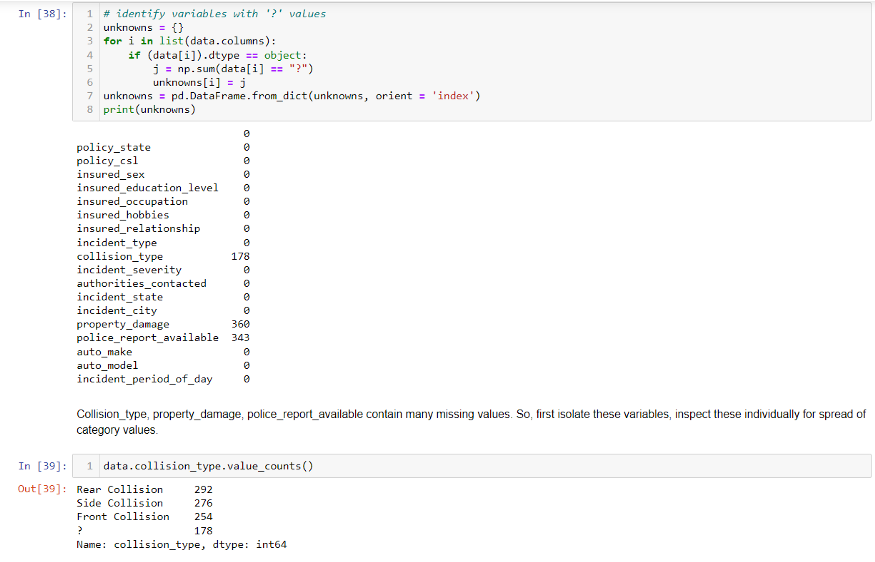
4- **Data reduction**

Checking to the categorical variable and viewing the importance for the prediction



Dropping Un-Important columns

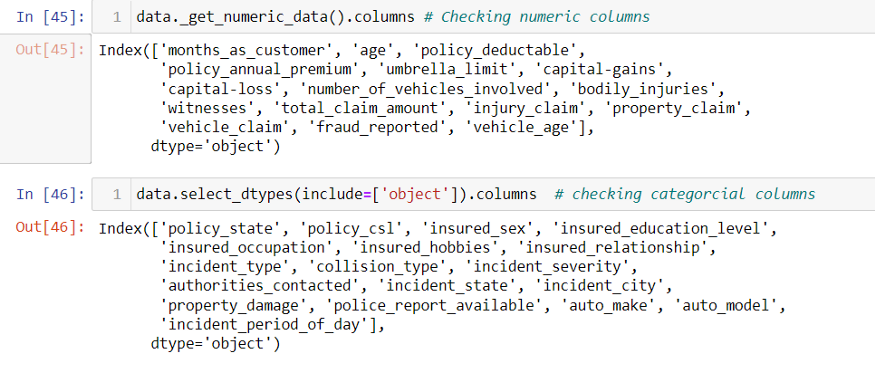
Viewing the data with **“?”** in the Dataset



Identify variables with ‘?’ values

Collision\_type, property\_damage, police\_report\_available contain many missing values. So, we will first isolate these variables, and also inspect these aspects individually.

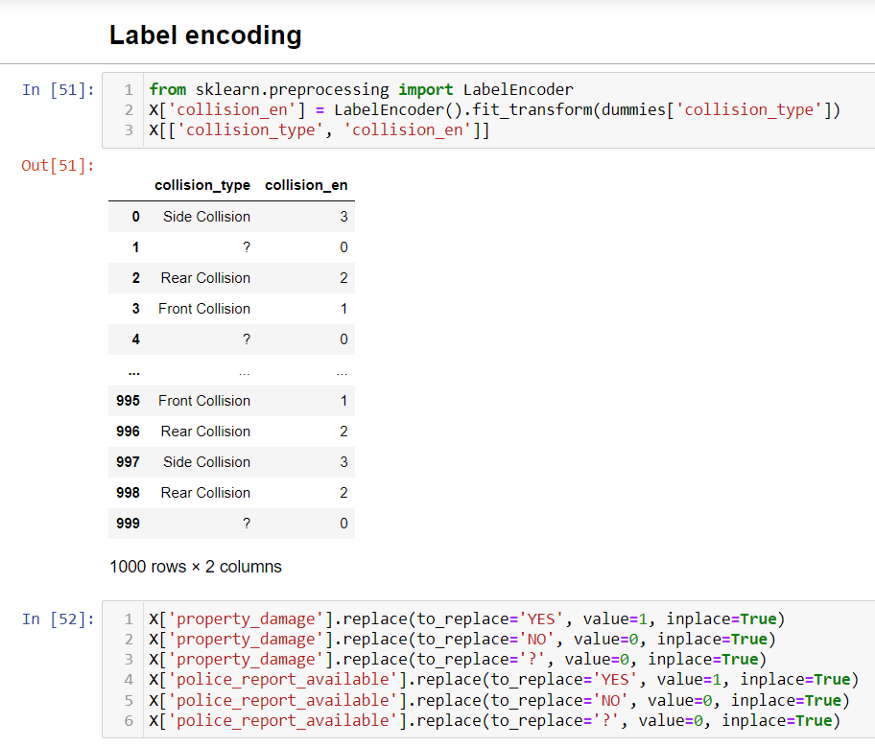
Distinguishing the Data into Numerical and Categorical columns



And Applying one-hot encoding to convert all categorical variables except out target variables.

**Label Encoder**

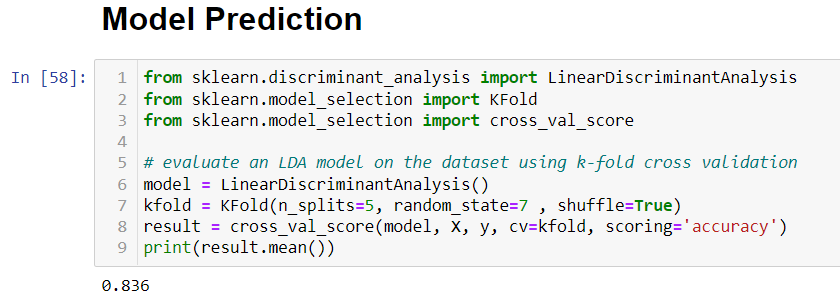
Converting the labels into numeric form so as to convert it into the machine readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is a very important Pre-Processing part for structured dataset in supervised learning.



Label Encoding

Model Prediction using LDA

LDA(**Linear Discriminant Analysis**) is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling different in various groups i.e. separating two or more classes. The features in a higher dimension space are projected into a lower dimension space using this technique.



Model Prediction using LDA

83.6% accuracy without standardizing the data. Going with the Random Forest Classification approach seems to be a good idea. Random Forest is a tree-based model and therefore it does not need any feature scaling. The convergence and numerical precision issues , which can even seldom trip up the algorithms used in logistic regression as well as linear regression, as well as even neural networks , are not so important in case of random forest.

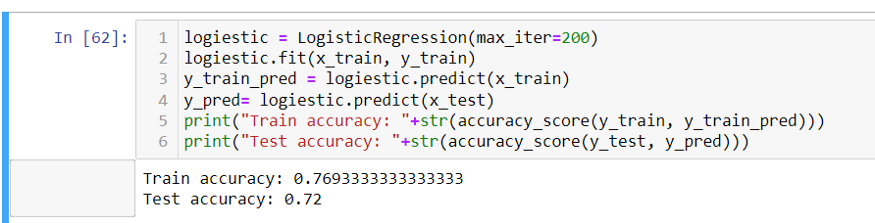
**Building machine learning models**

After performing Test Train split , I have used 4 ML models :

**Linear Regression**

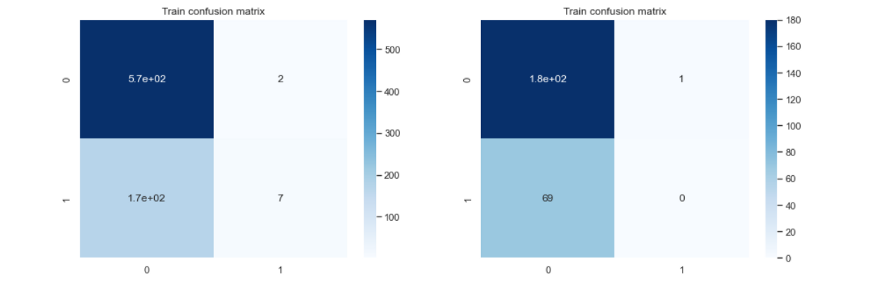
**Linear Regression** is a ML algorithm which is mostly based on **supervised learning**. It specifically performs a **regression task**. Linear regression just attempts to model the basic relationship between 2 variables by fitting on a linear equation to observed data . One of that variable is considered to be an explanatory variable , and the other variable is considered to be a dependent variable .

A linear regression is a line in an graph and has an equation of the general form of ***Y = mX + c***, where ***X*** is the explanatory variable and ***Y*** is the dependent variable. The slope of the line is ***m*** , and ***c*** is the intercept .



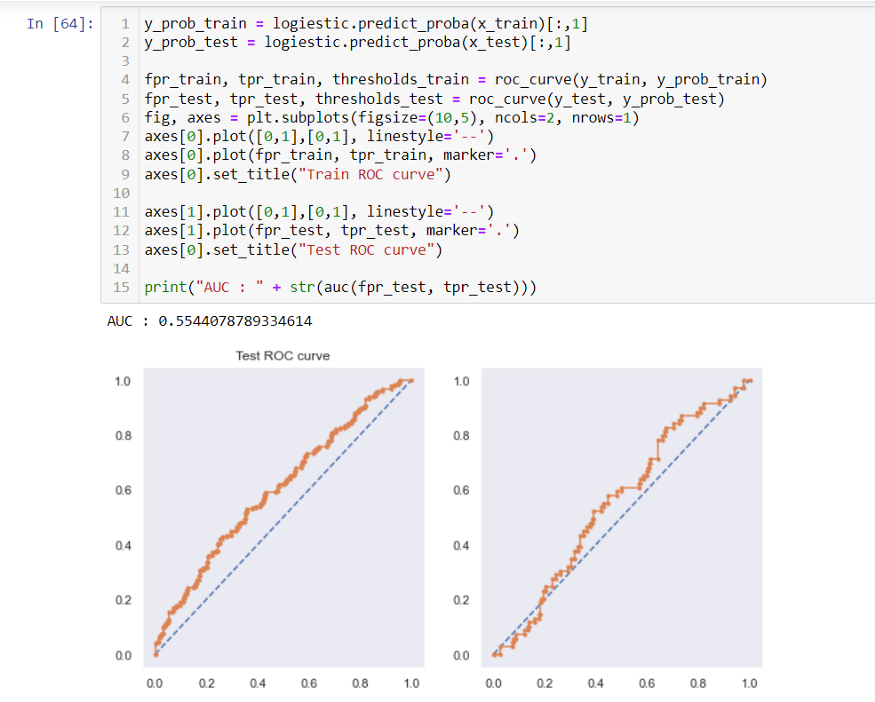
Linear test train prediction

**Confusion Matrix of Linear Regression**



Confusion Matrix

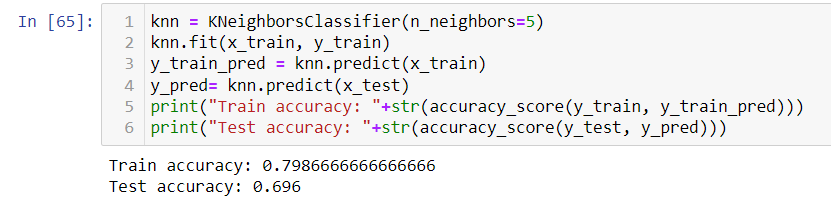
**ROC Curve in Linear Regression**



ROC Curve for Linear Regression

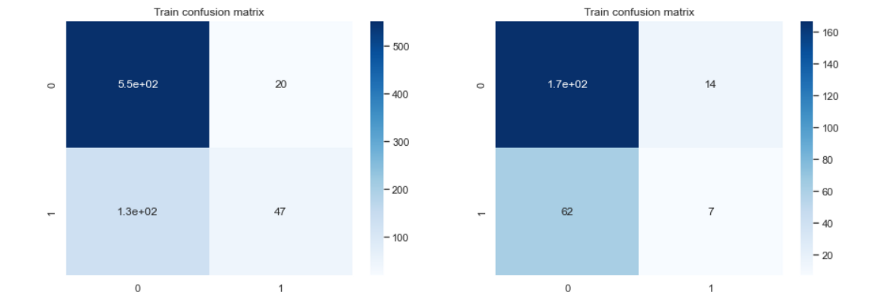
**KNN**

KNN is an easy-to-use, supervised machine learning (ML) technique that is frequently used in missing value imputation. It can be utilized for classification or regression problems. For both classification and regression issues, the KNN algorithm is used. KNN algorithm based on feature similarity approach. KNN is generally based on the idea that the observations closest to a given data point are the most “similar” observations in a data set, and we can therefore classify unforeseen points based on the values of the closest existing points. By choosing *KNN*, the user can select the number of nearby observations to use in the algorithm.



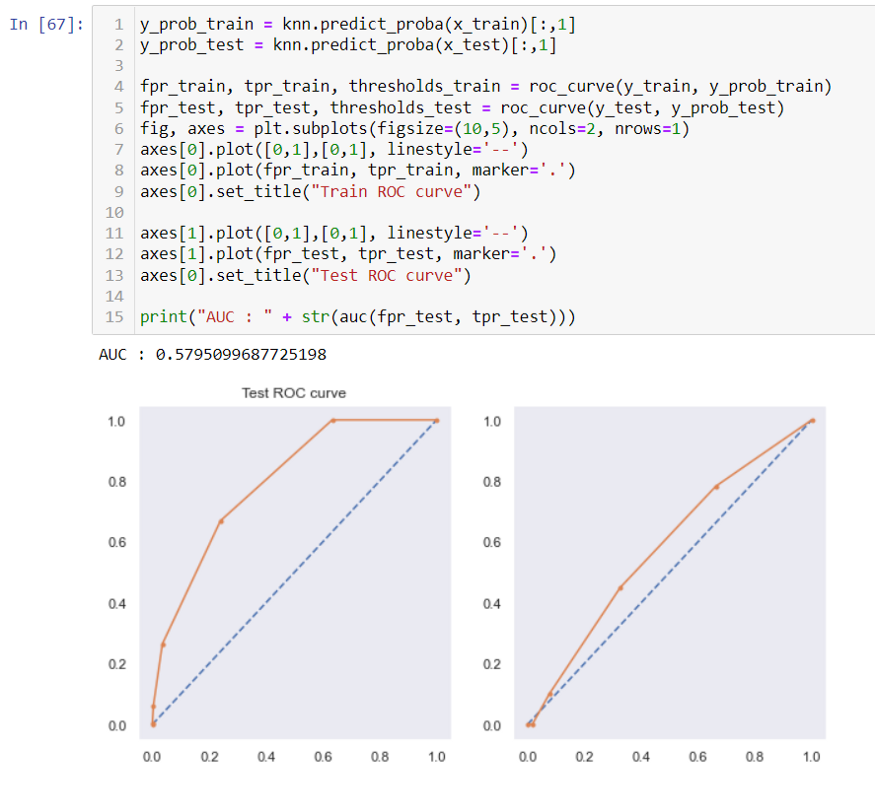
KNN test train prediction

**Confusion Matrix of KNN**



Confusion Matrix for KNN

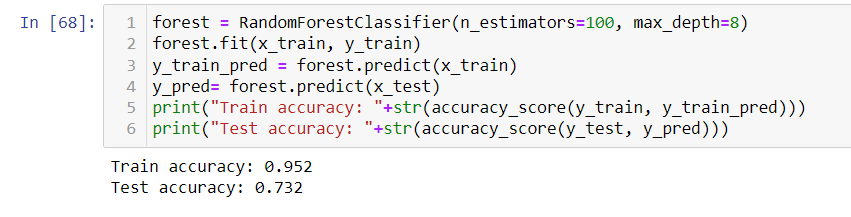
**ROC Curve in KNN**



ROC graph in KNN

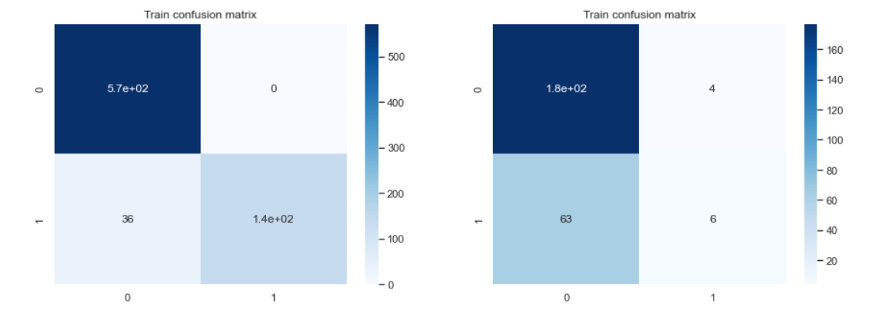
**Random Forest**

Random Forest is a robust machine learning algorithm that can be used for a variety of tasks including regression and classification. It is an ensemble method, meaning that a random forest model is made up of a large number of small decision tree , called estimators , which each produce their own predictions. The random forest model combines the predictions of the estimators to produce a more accurate prediction.



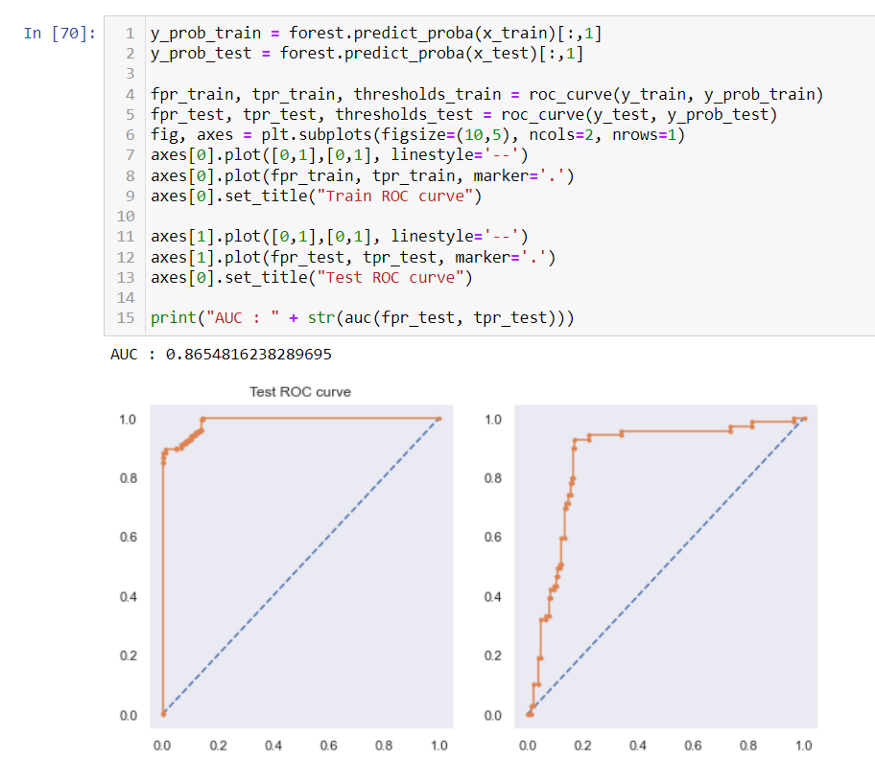
Random Forest Test Train prediction

**Confusion Matrix of Random Forest**



Confusion Matrix of Random Forest

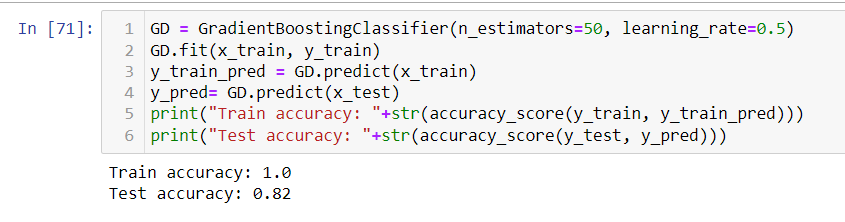
**ROC Curve of Random Forest**



ROC Curve for Random Forest

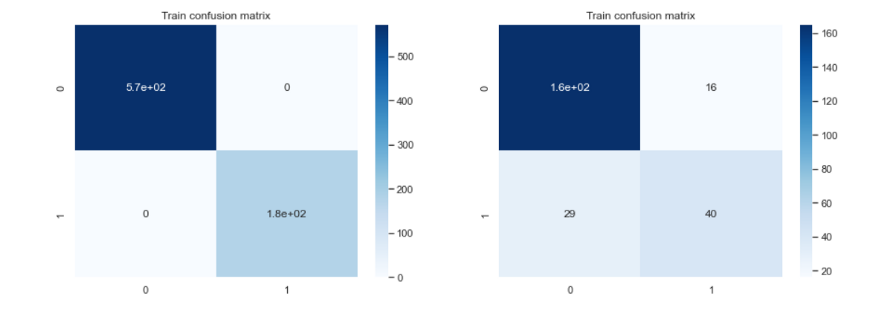
**Gradient Boosting**

Gradient boosting is a machine learning technique for regression and classification problems that produce a prediction model in the form of an ensemble of weak prediction models. This technique builds a model in a stage-wise fashion and hypothesize the model by allowing development of an random differentiable loss function. Gradient boosting basically mixed weak learners into a unit strong learner in a continual fashion. A new model is fitted for each additional weak learner to give a more precise estimate of the response variable. The negative gradient of the loss function linked to the entire ensemble is maximally correlated with the new weak learners. By combining a number of relatively weak prediction models, gradient boosting seeks to create a stronger prediction model.



Gradient Boosting Test-Train Prediction

**Confusion Matrix of Gradient Boosting**



Confusion Matrix for Gradient Boosting

**ROC Curve of Gradient Boosting**



ROC Curve for Gradient Boosting

**Conclusion:**

So here “**Gradient Boosting Model”** is the best model out of all model tested above and by looking this we can conclude that our model is predicting around Train Accuracy of 100% and Test Accuracy of 82% .

Depending on the accuracy and ROC AUC graphs with AUC Accuracy of 88.237% on Gradient Boosting Model we can assume that Gradient boosting model is the best model that fit this data.