## Project on

## Insurance Claims- Fraud Detection Using Machine Learning



**Submitted by:**

**Arpita Rai**

**Introduction**

Insurance is one of the largest industries after banking and plays a major role in any countries economy. It’s a tool whereby losses of few is borne by a pool contributed by many.

Insurance fraud is one aspect which not only eats the margins of Insurers but also increases the cost of claim leading higher premium for customers. Though there are many stages where frauds are detected in insurance business cycle, but fraud in claims is the most common ones.

In United States market Insurance fraud are in tune of 80 Bln $ including all lines of business but most commons ones are:

* Motor Fraud Claims.
* Health Fraud Claims.
* Marine Fraud claims.
* Liability Fraud Claims.
* Staged Fire Claims.

To detect such frauds Insurers use different tools, process check etc.

In this project we are trying to develop a model using **Machine Learning** which can predict whether the claim is fraudulent or genuine.

**Problem Statement:-**

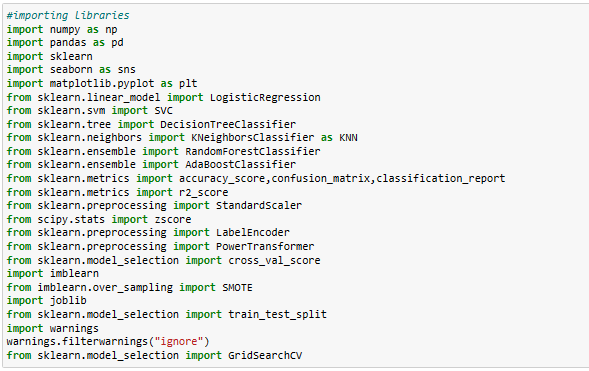
Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem.

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.

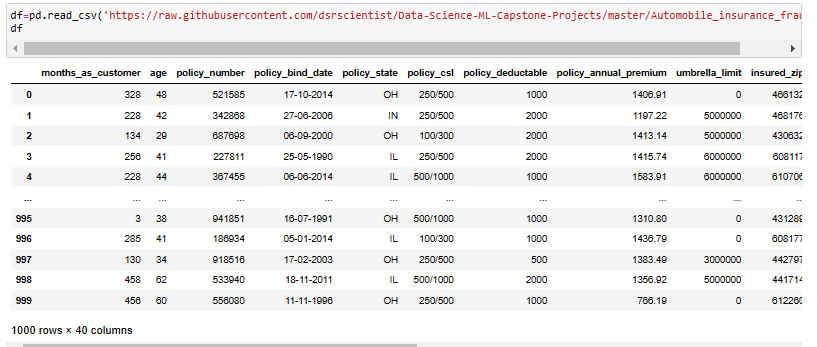
In this example, you will be working with some auto insurance data to demonstrate how you can create a predictive model that predicts if an insurance claim is fraudulent or not.

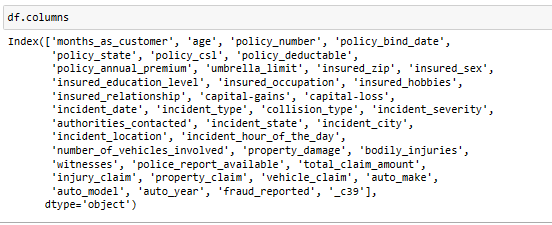
**Data Analysis:-**

We have imported the library and the dataset which contains all the detail like customer detail and insurance policy and claims details. The dataset has 1000 rows and 40 columns, it has categorical as well as numerical values. We can see all the column names.



This is how the dataset looks like.





**Exploratory Data Analysis (EDA)**

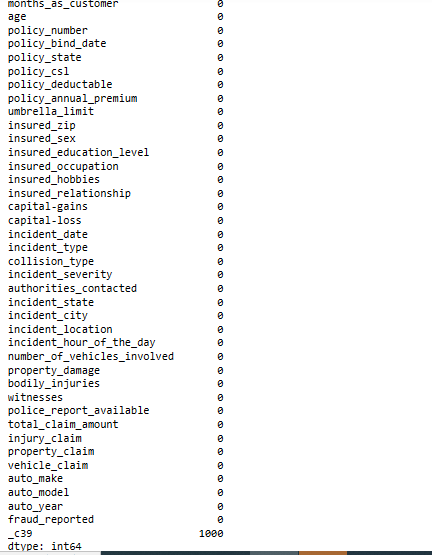
The main purpose of EDA is **to help look at data before making any assumptions**. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

There are many null/? values and unique values in the dataset. So by EDA we can check and fill or rectify these small things to make the dataset clean and balanced.

Let’s check the shape of the dataset, unique values and null values present in the dataset.

pic.png

pic.png

****

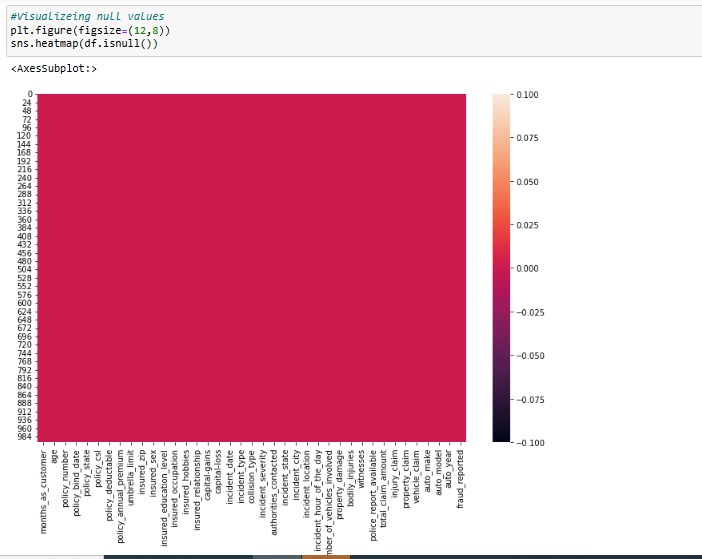
As we can see that there are 1000 null/missing values in the column \_C39 in the dataset, so we try fixing it either by dropping the column or by filling the it mean and mode.

**Handling Null/Missing values**

**pic.png**

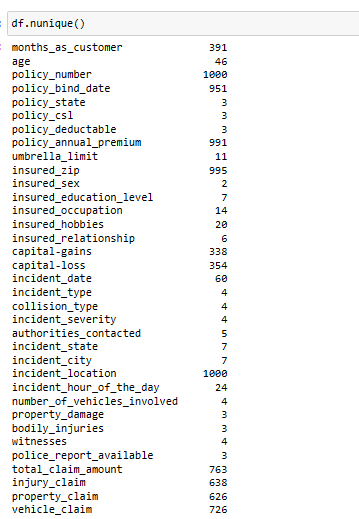
As the column \_C39 is not very important or will not affect the data so we can drop this column.

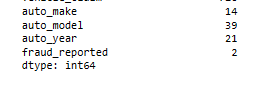
**Checking and Plotting Null values on Heatmap**

****

In the above Heatmap we cannot see any other color line which confirms that this data is without any null values.

**Checking for Unique values**

****

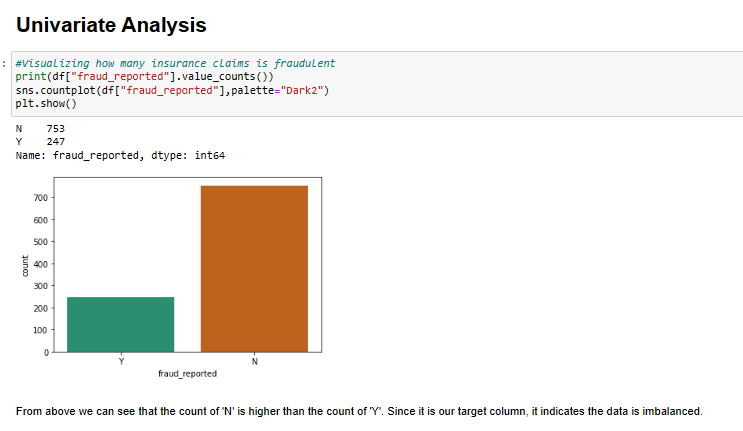
****

We can see that there are unique values in every column.

**Separating Categorical and Numerical columns:-**

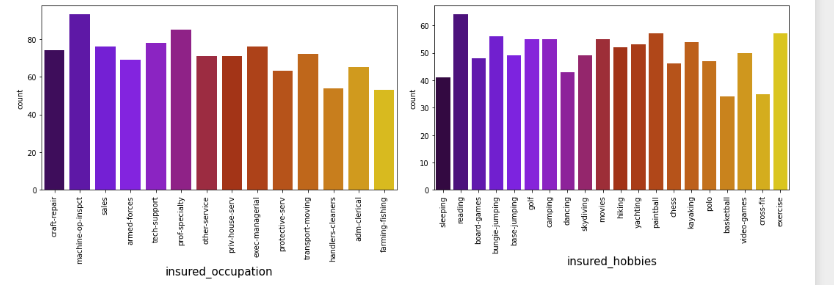
****

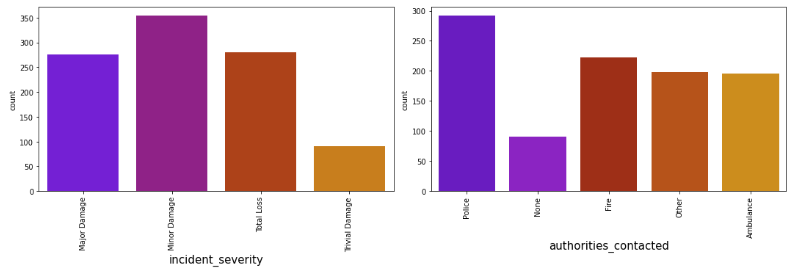
**Data Visualization:-**

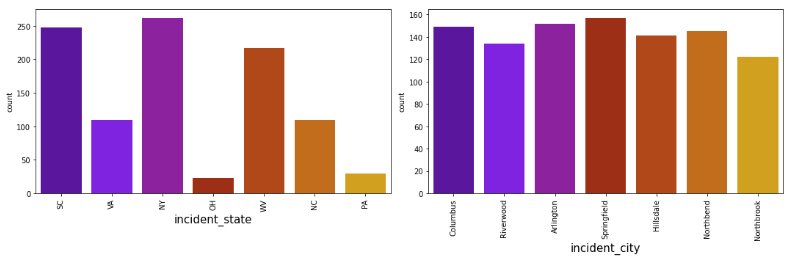
****

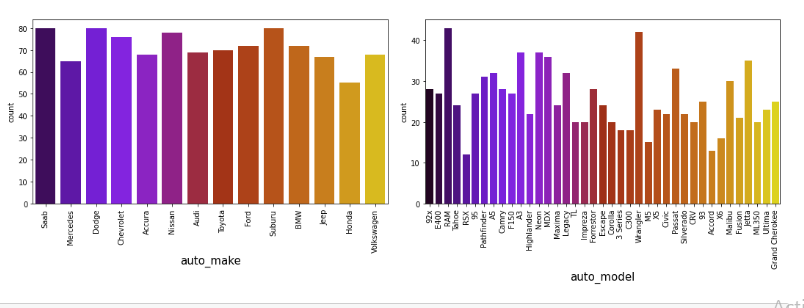
From the above graph we can see that the data is imbalanced

**Visualizing other Varaibles:-**

****

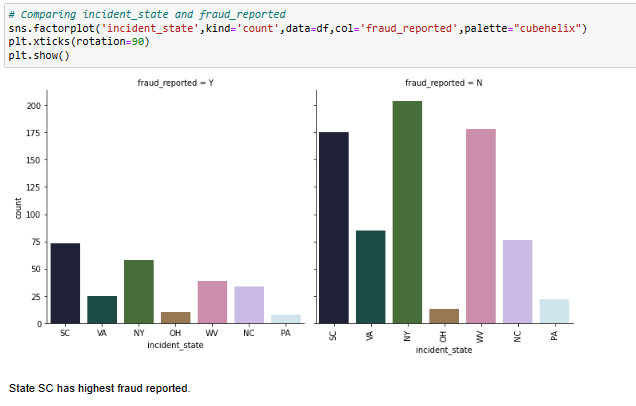
****

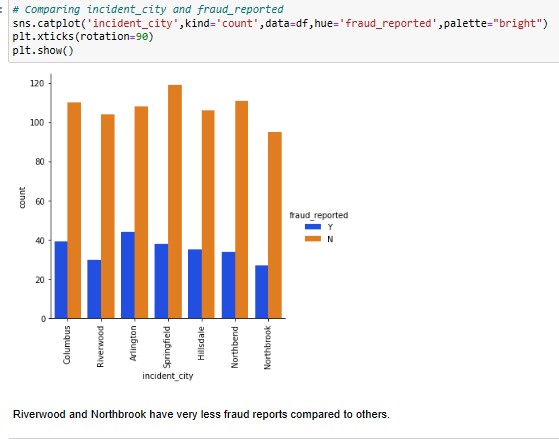
****

****

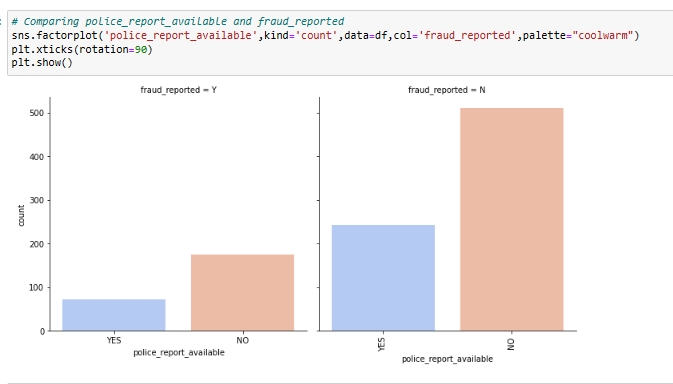
In the given data, ‘fraud\_reported’ is the Target variable. The unique values of this feature are Y and N (Yes and No), which means it has only two classes.

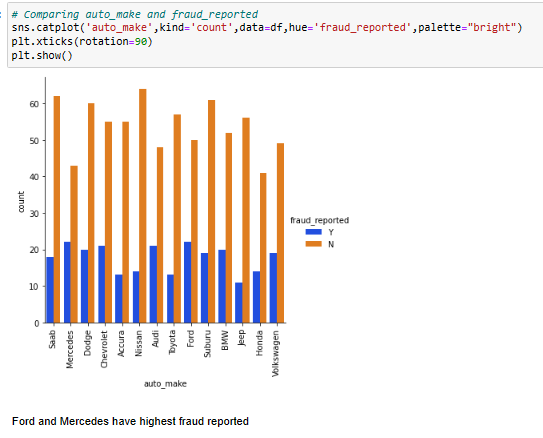
**Now we will compare The Target variable with other variables.**

****

****

****

****

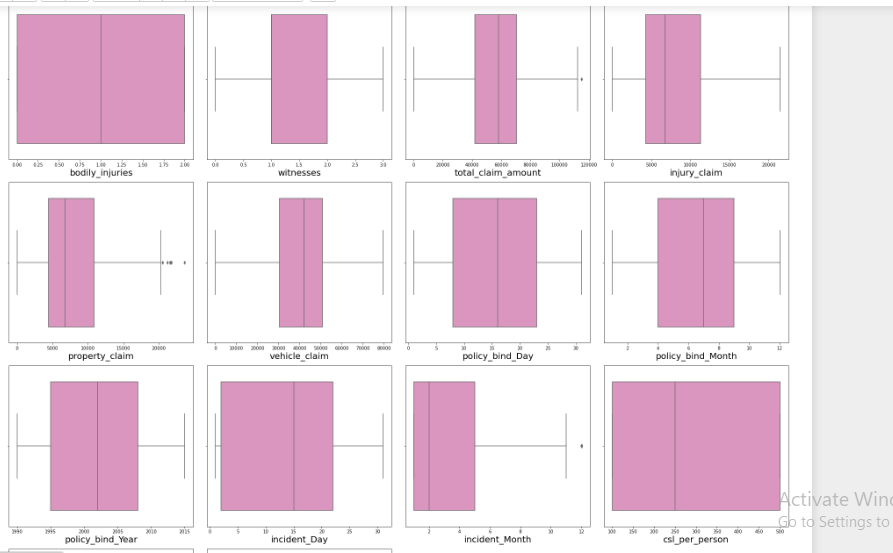
****

**Checking for Outliers:-**

**Outliers** are data that do not conform to the usual trends in your dataset. They are often too small or too big and generally rebels from following a pattern.

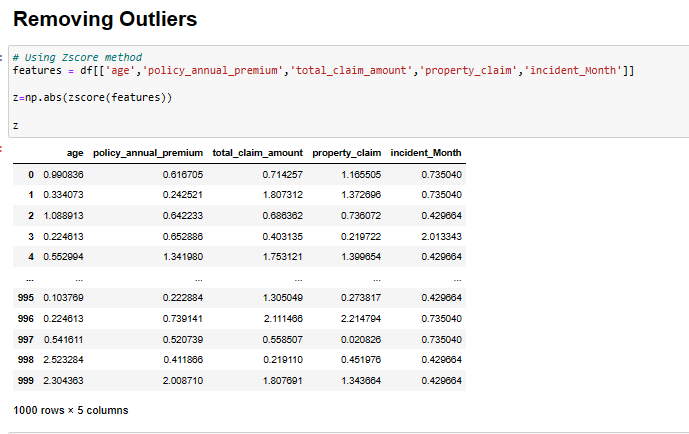
Presence of outliers can be due to various reasons but it is important to identify valid outliers and remove them. Presence of outliers can lead to noise in the data model. But this should be seen in a positive light as model being grounded in reality.





We can see that the outliers are present in the columns: **age ,policy\_annual\_premium ,total\_claim\_amount ,property\_claim and incident\_month.**

So now we will try to remove them by using ZSCORE Method.

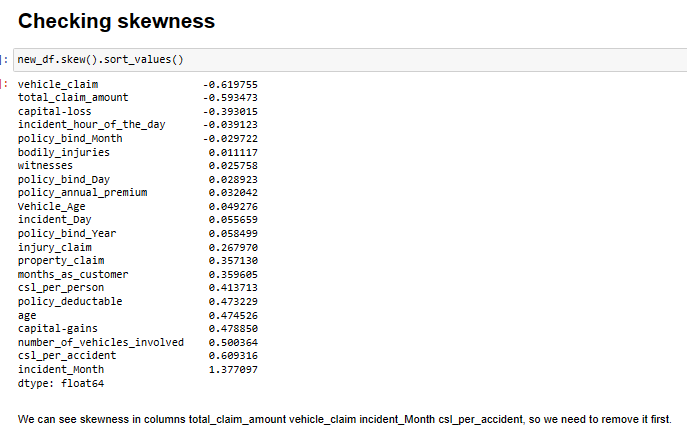


**Skewness:-**

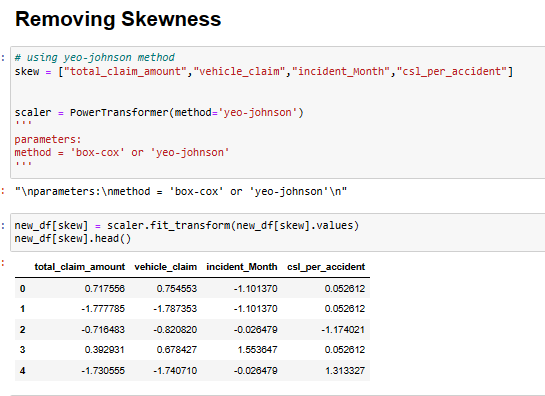
Skewness is **a measure of the asymmetry of a distribution**. A distribution is asymmetrical when its left and right side are not mirror images. A distribution can have right (or positive), left (or negative), or zero skewness.

Skewness **gives the direction of the outliers** if it is right-skewed, most of the outliers are present on the right side of the distribution while if it is left-skewed, most of the outliers will present on the left side of the distribution.

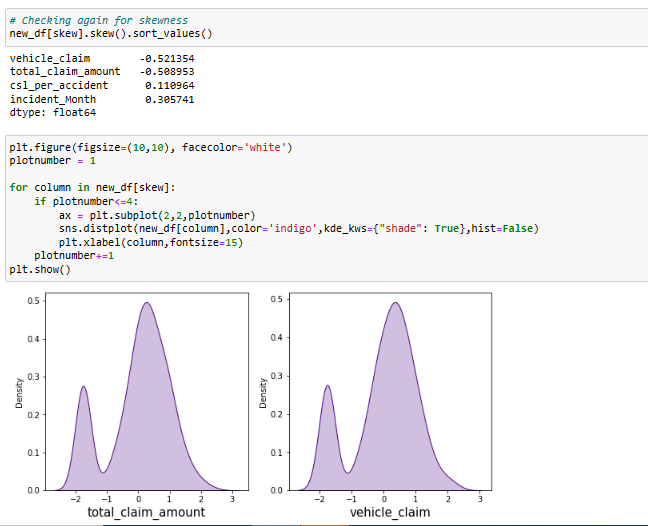
Now let’s check Skewness in our dataset.

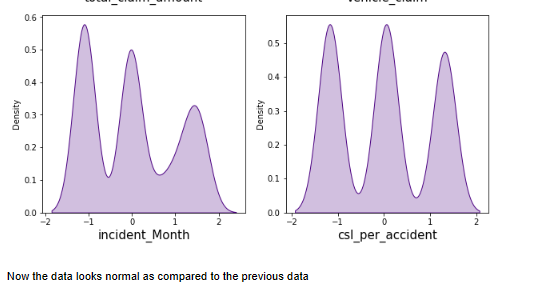


**Now we will try removing Skewness using Yeo-Johson Method**



Again checking and Visualizing Skewness to confirm that it removed.

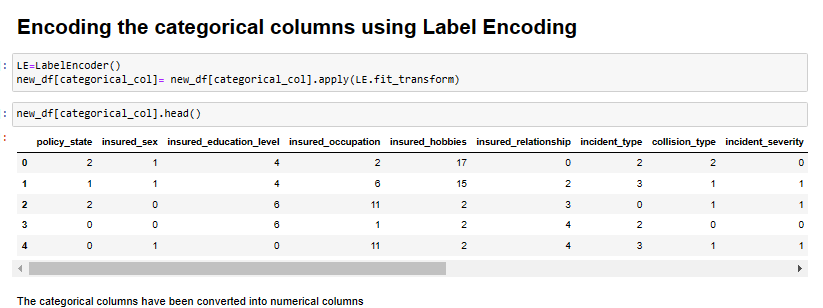




**Label Encoding:-**

Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated.

Machine learning models can only work with numerical values. For this reason, it is necessary **to transform the categorical values of the relevant features into numerical ones**. This process is called feature encoding.

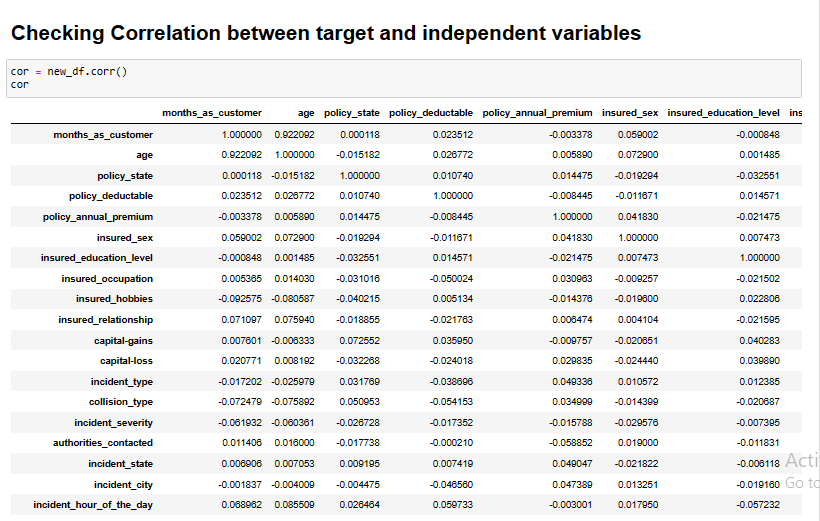


**Checking the Correlation:-**

Correlation is **a statistical measure (expressed as a number) that describes the size and direction of a relationship between two or more variables**. A correlation between variables, however, does not automatically mean that the change in one variable is the cause of the change in the values of the other variable.

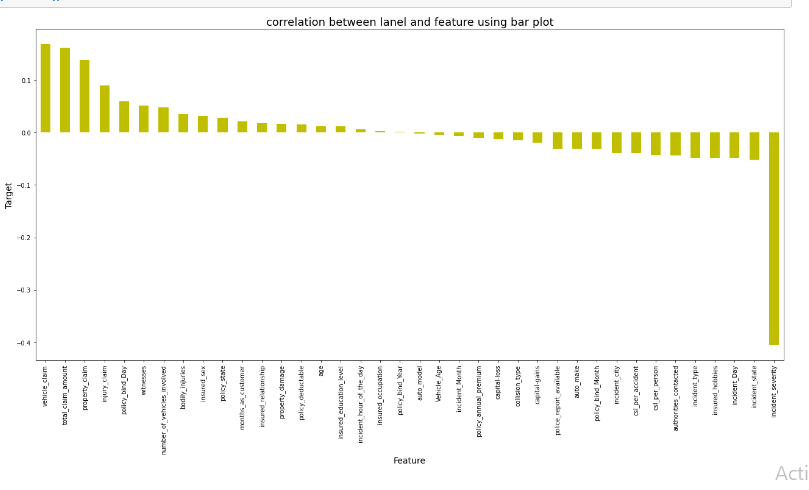
Correlations have three important characterstics. They can tell us about **the direction of the relationship, the form (shape) of the relationship, and the degree (strength) of the relationship between two variables**.

A correlation coefficient greater than zero indicates a positive relationship while a value less than zero signifies a negative relationship.

****

****

**Visualizing Correlation between label and features**

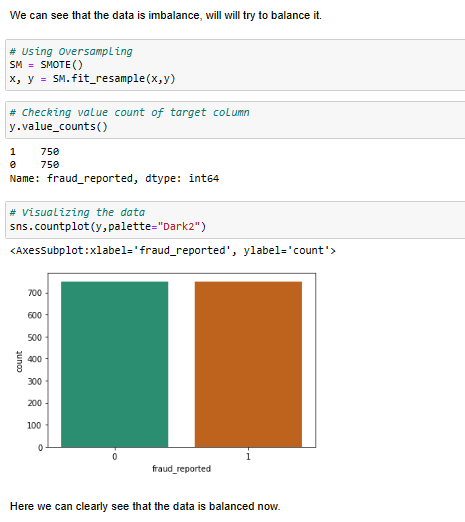
****

**Balancing the Data:-**

As we know that our dataset is imbalanced so we will try balancing it by using SMOTE method and we will also visualize it on graph.

**SMOTE:-**

SMOTE (synthetic minority oversampling technique) is **one of the most commonly used oversampling methods to solve the imbalance problem**. It aims to balance class distribution by randomly increasing minority class examples by replicating them. SMOTE synthesises new minority instances between existing minority instances.

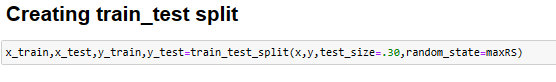


# ****Building machine learning models:-****

Building an ML Model **requires splitting of data into two sets, such as 'training set' and 'testing set' in the ratio of 80:20 or 70:30**

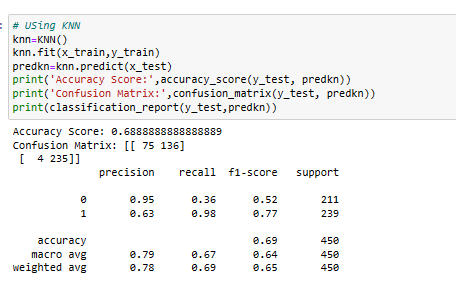
**Machine Learning techniques are divided mainly into the following 4 categories:**

* Supervised Learning. Supervised learning is applicable when a machine has sample data, i.e., input as well as output data with correct labels.
* Unsupervised Learning
* Reinforcement Learning
* Semi-supervised Learning

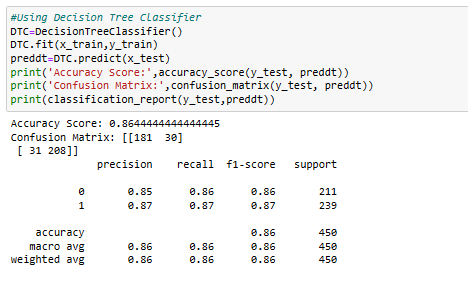


I have selected 4 models:-

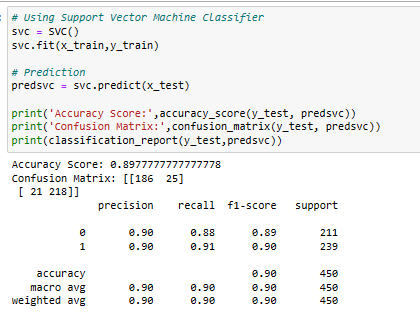
1. **K-Nearest Neighbor:-** The k-nearest neighbors algorithm, also known as KNN or k-NN, is **a non-parametric, supervised learning classifier.** **It's easy to understand and simple to implement**. It can be used for both classification and regression problems. It's ideal for non-linear data since there's no assumption about underlying data. It can naturally handle multi-class cases.

****

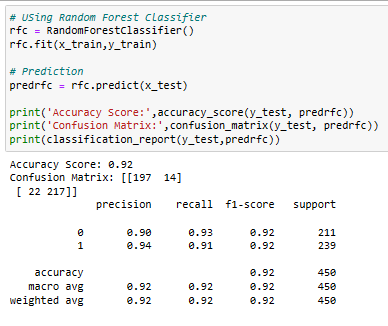
1. **Decision Tree Classifier:-** A decision tree is a support tool with a tree-like structure that models probable outcomes. Decision Tree is a very popular machine learning algorithm. Decision Tree solves the problem of machine learning by transforming the data into a tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label.

****

1. **Support Vector Machine Classifier:-** The main advantage of SVCs over simple mechanically switched compensation schemes is their **near-instantaneous response to changes in the system voltage**. For this reason they are often operated at close to their zero-point in order to maximize the reactive power correction they can rapidly provide when required.

****

1. **Random Forest Classifier:-** Among all the available classification methods, random forests **provide the highest accuracy**. The random forest technique can also handle big data with numerous variables running into thousands. It can automatically balance data sets when a class is more infrequent than other classes in the data.

****

**Conclusion from the models:-**

We can conclude that Random Forest Classifier is our best fitting model which is giving very less difference as compared to other models.

The accuracy score of 89.77%.

Here our model predicts 197 true positive cases out of 211 positive cases and 217 true negative cases out of 239 cases.

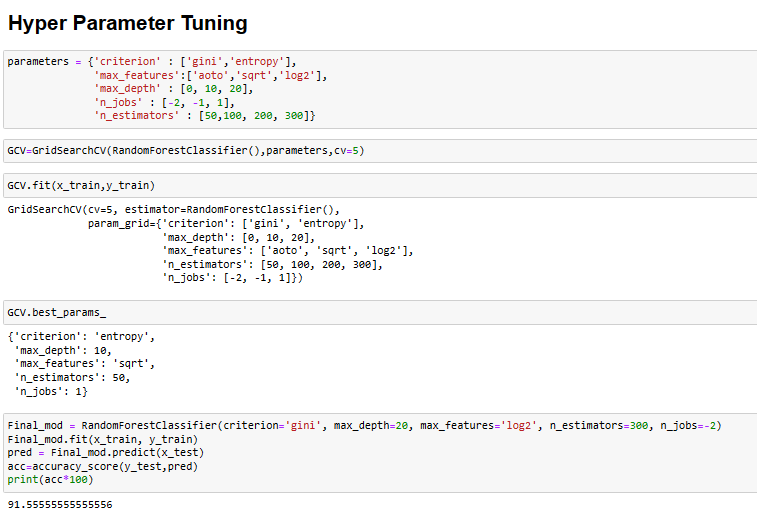
It predicts 14 false positive cases out of 211 positive cases and 22 false negative cases out of 239 cases.

# ****Hyper parameter tuning:-****

Hyperparameter tuning is **an essential part of controlling the behavior of a machine learning model**. If we don't correctly tune our hyperparameters, our estimated model parameters produce suboptimal results, as they don't minimize the loss function. This means our model makes more errors.

We will use **GridSearchCV** for the hyper parameter tuning.

In **GridSearchCV** the model is evaluated for a range of hyper parameter values. This approach is called **GridSearchCV**, because it searches for best set of hyper parameters from a grid of hyper parameters values.

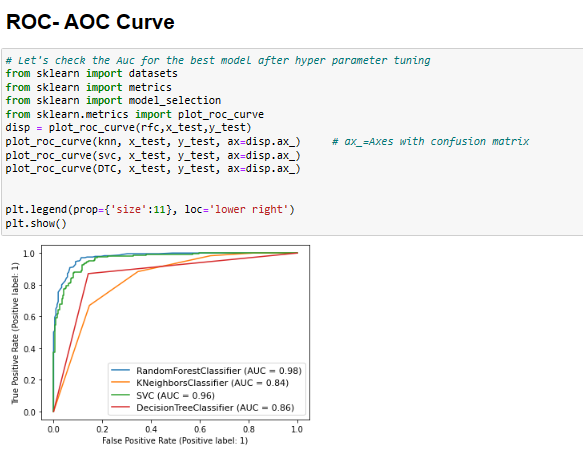


**ROC Curve:-**

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

* True Positive Rate
* False Positive Rate

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



After plotting the ROC Curve we have saved model using Pickle.

**Remarks:-**

* We have handled both the categorical and numerical data properly and also have checked by building different ML model on the same dataset.
* This project was built to detect insurance fraud.
* The model can help to reduce loss of insurance companies and can also help to differentiate between fraud and legit claims.
* Using this machine Learning Model we people can easily predict the insurance claim is fraudulent or not and we could reject those application which will be considered as fraud claims.
* Five different classifiers were used in this project: K-nearest neighbors, Random forest, Decision tree and SVC.
* We handled the imbalance data with classifiers like - oversampling with SMOTE, hyper parameter tuning, and plotting roc curve of the models.
* We have checked accuracy score of each model and chose the one which has the best of the same.
* We can see that in ROC score are good also the maximum of the area under the curve fall under true positive rate. Therefore we have saved the model as .obj file using Pickle so that it can be used to predict the result of the different data sets.