Insurance Fraudulent Claim Detection

## Problem Statement

Global Insure, a leading insurance company, processes thousands of claims annually. However, a significant percentage of these claims turn out to be fraudulent, resulting in considerable financial losses. The company’s current process to identify fraudulent claims involves manual inspections, which is time-consuming and inefficient. Fraudulent claims are often detected too late in the process, after the company has already paid out significant amounts. Global Insure wants to improve its fraud detection process using data-driven insights to classify claims as fraudulent or legitimate early in the approval process. This would minimize financial losses and optimize the overall claims handling process.

## Business Objective

Global Insure wants to build a model to classify insurance claims as either fraudulent or legitimate based on historical claim details and customer profiles. By using features like claim amounts, customer profiles and claim types, the company aims to predict which claims are likely to be fraudulent before they are approved.

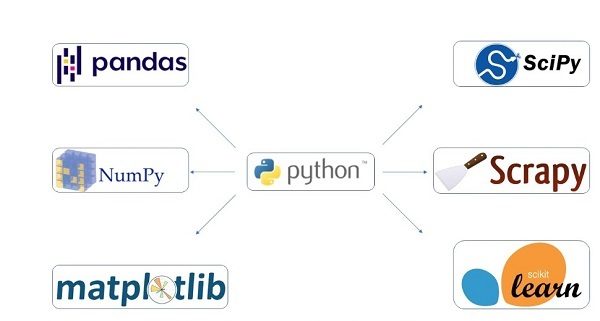
This project utilizes Machine Learning models to assist the Auto Insurance sector in addressing this issue.

## Hardware & Software Requirements

* OS: Windows 11 Enterprise, 64 bit
* Processor: 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz 3.00 GHz
* RAM: 16 GB
* Anaconda Navigator - Jupyter Notebook

**Libraries** :

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Scipy
* Date Time
* Scikit Learn



# Exploratory Data Analysis (EDA)

**Import the necessary Libraries:**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from scipy.stats import zscore

from sklearn.preprocessing import PowerTransformer

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import StandardScaler

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

from sklearn import datasets

from sklearn import metrics

from sklearn import model\_selection

#from sklearn.metrics import plot\_roc\_curve

from sklearn.metrics import roc\_curve

import warnings

warnings.filterwarnings('ignore')

## Data Overview

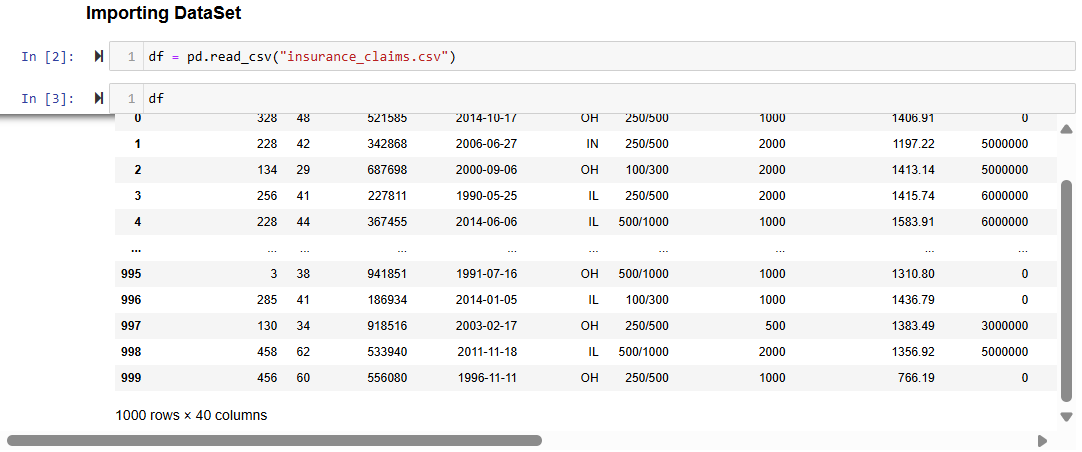
The data used in this case study comprises insurance claims with various attributes such as claim amount, claimant details, type of insurance, and claim status. The dataset includes both genuine and fraudulent claims, labeled accordingly.

### Preprocessing the Data

The first step in detecting fraudulent claims is to preprocess the data. This involves:

* Handling missing values
* Encoding categorical variables
* Normalizing numerical features
* Splitting the data into training and testing sets

#### **Import the Dataset**



By examining the dataset, we can determine that it includes both categorical and numerical columns. The "fraud\_reported" column serves as our target variable. Since it contains two categories, this problem is classified as a "Classification Problem," where our goal is to predict whether an insurance claim is fraudulent. Given that it is a classification problem, we will employ various classification algorithms during the model-building process, which will be discussed as we proceed with the data analysis.

#### **Data Preparation**

In this part we will firstly be exploring the data with some basic steps and then further proceed with some crucial analysis, like feature extraction, imputing and encoding.

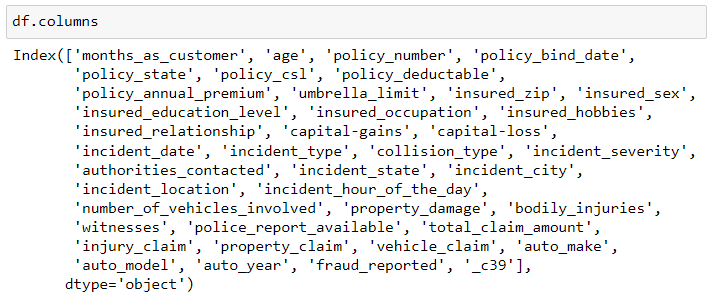
Let’s start with checking shape, unique values, value counts, info etc.

After doing the analysis if we find any unnecessary columns in the dataset, we can drop those columns.

By using ‘df.shape’, we determined the number of rows and columns in the dataset, which revealed 1000 rows and 40 columns. Although PCA (Principal Component Analysis) could be applied, we chose not to reduce the data at this stage. Given the relatively small size of the dataset and the fundamental principle for data scientists to retain as much data as possible, we decided to proceed with the entire dataset.



Out of 40 columns 39 are independent columns and remaining one is our target variable “fraud\_reported".



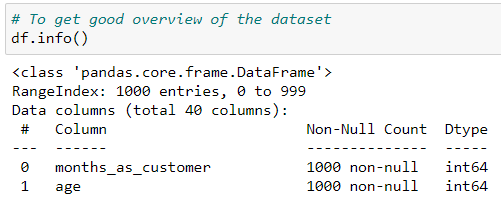
**Company's data for insurance claim policy :**months\_as\_customer, age, policy\_number(dropped), policy\_bind\_date, policy\_csl, policy\_deductable, policy\_annual\_premium, umbrella\_limit

**Customer Personal details :**insured\_zip, insured\_sex, insured\_education\_level, insured\_occupation, insured\_hobbies, insured\_relationship, capital- gains, capital-loss.

**Details of the incident :**incident\_date, incident\_type, collision\_type, incident\_severity, authorities\_contacted, incident\_state, incident\_city, incident\_location(dropped), incident\_hour\_of\_the\_day, number\_of\_vehicles\_involved, property\_damage, bodily\_injuries, witnesses, police\_report\_available, total\_amount\_claimed, injury\_claim, property\_claim, vehicle\_claim, auto\_make, auto\_model, auto\_year

**Target Variable:**

fraud\_reported : Y-YES / N-NO

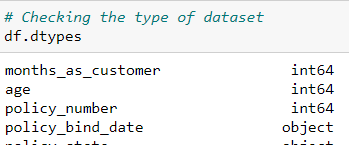


This gives the information about the dataset which includes indexing type, column Name, non-null values, dtypes and memory usage of the dataset.

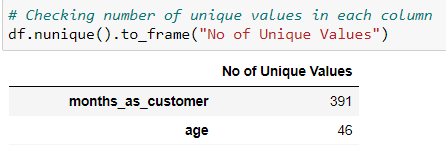
Here the column \_c39 has 0 non null values which means it has NaN throughout the data so we can drop this column.



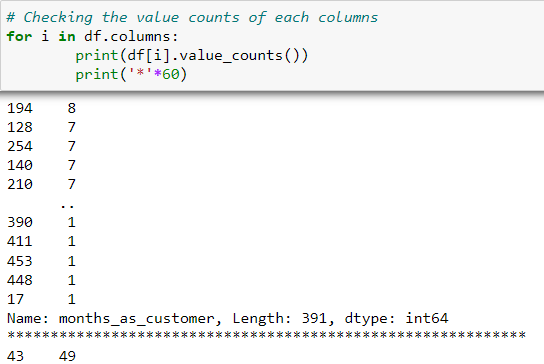
Checking the data types using the “df.dtypes” below.



Checking the number of unique values of each column in the data set using the code below



Checking the value counts of each columns using the code below.



Check Null values

A screenshot of a computer

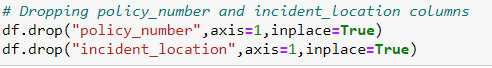
AI-generated content may be incorrect.

#### **Observations**

* We can see that there are null values in the dataset.
* The dataset contains 3 different types of data namely integer data type, float data type and object data type.
* After analyzing it is seen that c39 column has only entries those are all NaN. Keeping all entries NaN is useless hence dropping that column.



* We can observe the columns “policy\_number” and “incident\_location” have 1000 unique values. So, it not required for the prediction so we can drop it.



* These are the list of value counts in each columns. By looking at the value counts of each column we can realize that the columns umbrella\_limit, capital-gains and capital-loss contains more zero values around 79.8%, 50.8% and 47.5%. I am keeping the zero values in capital\_gains and capital\_loss columns as it is. Since the umbrella\_limit columns has more that 70% of zero values, let's drop that column. Also the collision\_type, police\_report\_available and property\_damage have ? in them



* The column insured\_zip is the zip code given to each person. If we take a look at the value count and unique values of the column insured\_zip, it contains 995 unique values that means the 5 entries are repeating.

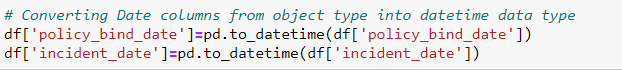


# Feature Engineering

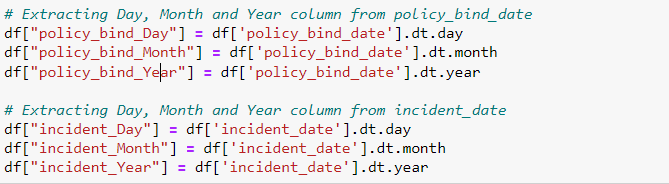
Feature engineering is a crucial step in improving the accuracy of the model. In this case study, we derived new features based on the existing data, such as:

* Claim frequency by the same claimant
* Time taken to file the claim after the incident
* Comparison of claim amount with average claims for similar incidents

#### **Feature Extraction**

The policy\_bind\_date and incident\_date have object data type which should be in datetime data type that means the python is not able to understand the type of these columns and giving default data type. We will convert this object data type to datetime data type and we will extract the values from these columns.

After converting object data type into datetime data type, we will extract Day, Month and Year from both the columns.



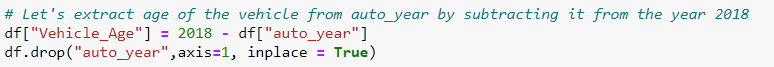
After we have extracted Day, Month and Year columns, from both policy\_bind\_date and incident\_date columns. So, we can drop these columns.



Also, we have observed that the feature ‘incident-year’ has one unique value throughout the column also it is not important for our prediction so we can drop this column.

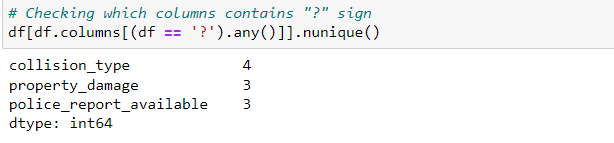


Here we have extracted age of the vehicle based on auto year by assuming the data is collected in the year 2018 as below and dropped “auto year” column after extraction.

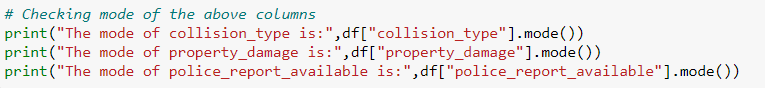


**Imputation:**

Imputation is a technique to fill null values in the dataset using mean, median or mode. We did not get any null values while checking for the null values, however from the value counts of the columns we have observed that some columns have "?" values, they are not NAN values but we need to fill them.



These are the columns which contains "?" sign. Since these columns seems to be categorical so we will replace "?" values with most frequently occurring values of the respective columns that is their mode values.



The mode of property damage and police\_report\_available is "?", which means the data is almost covered by "?" sign. So, we will fill them by the second highest count of the respective column.



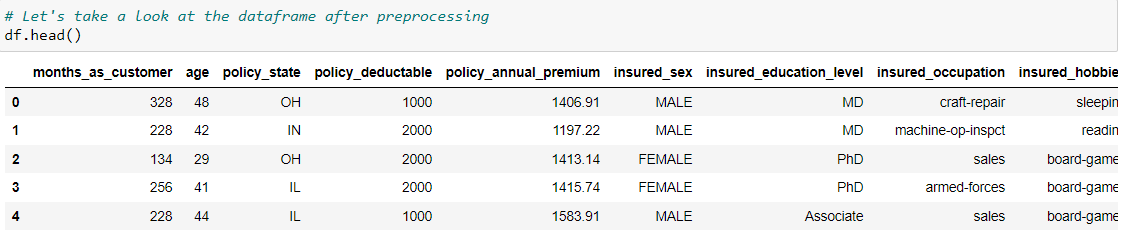
A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

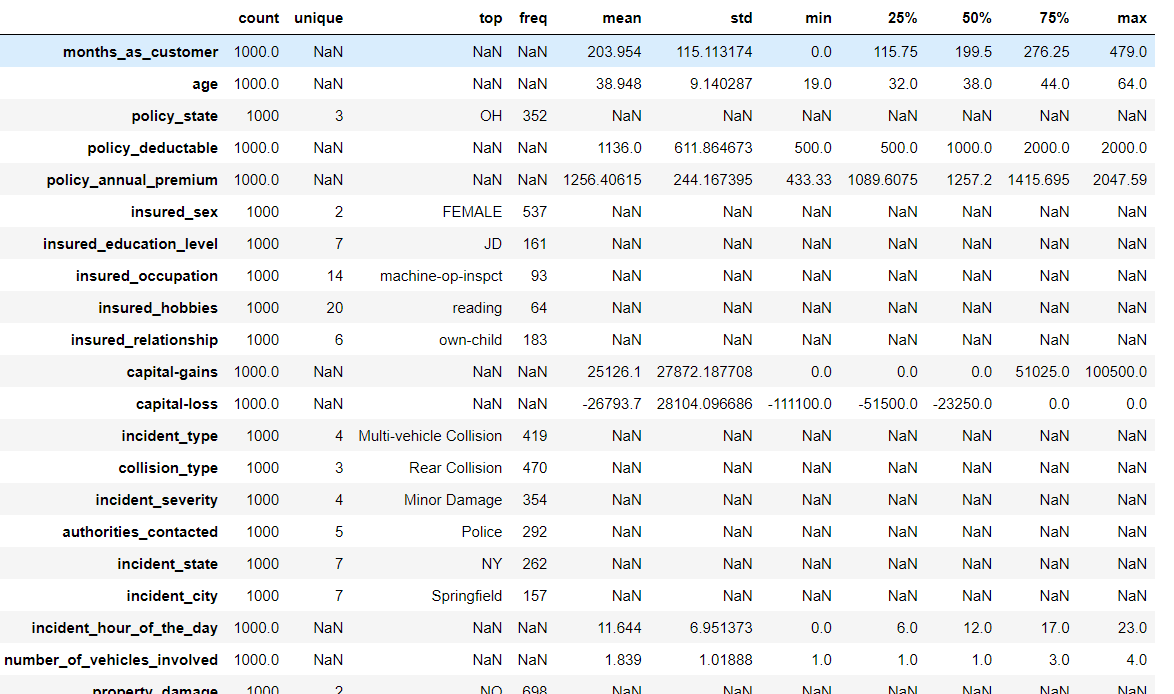
Now after all the data cleaning until now, the dataset looks like below



#### **Statistical Summary of Dataset**



Here we use "describe" method along with it's parameter "all" to include each column present in our dataset irrespective of them being numeric or text data. We have used transpose method to see the column information properly without scrolling multiple times.



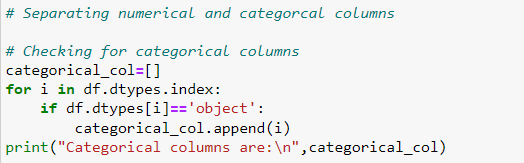
**From the above dataset, we can infer below details:**

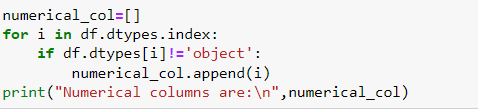
* The counts of all the columns are equal => no missing values in the dataset.
* In some columns like policy\_deductable, capital-gains, injury\_claim etc, mean value is greater than the median(50%), => data in these columns are skewed to right.
* Columns like total\_claim\_amount, vehicle\_claim etc, the mean is less than the median => data in these columns are skewed to left.
* Columns like policy\_annual\_premium have equal mean and median that means the data is symmetric and is normally distributed and no skewness present.

## Data Visualization

**Preparing for Visualization**

We will look into the categorical and numerical columns so that we can create two different lists and visualize the features accordingly.





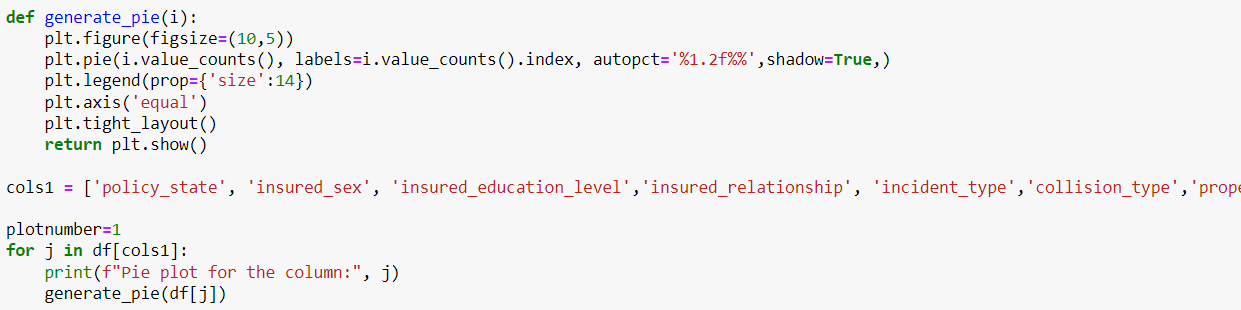
### Univariate Analysis (Categorical Columns)

A screenshot of a computer screen

AI-generated content may be incorrect.

From the plot we can observe that the count of "N" is high compared to "Y".

We will balance the data using the oversampling method in later part.



A pie chart with numbers and a percentage

AI-generated content may be incorrect.

A pie chart with numbers and a few percentages

AI-generated content may be incorrect.

A pie chart with numbers and a percentage

AI-generated content may be incorrect.

Above are the pie plots for some of the categorical columns and below are few observations

* The types of the policies claimed by the customers are almost same but still the policy state type OH has bit high counts and the type IL has bit less count.
* Both male and female have insurance but the count for Female is higher than Male counts.
* The count is pretty much same for all the education level but still the people who have completed their college and PhD have less count compared to others.
* Similar to insured education, insured relationship is also almost equally distributed.
* Under the incident type, Multi-vehicle collision and Single Vehicle Collision have almost similar counts of around 41.9% and 40.3%. But the count is very less in Parked car and Vehicle Theft.
* The collision type has 3 different types . In this the count is high in Rear collision and the other two types have almost equal counts. As we observe the propert damage plot, around 69.8% of the people did not face any property damage while 30.2% of the people faced the property damage. About 68.6% of the people produced the police reports while 31.4% of the people didn't submit any police reports.

A screen shot of a computer

AI-generated content may be incorrect.

A group of colorful bars

AI-generated content may be incorrect.

* Above are the count plots for the remaining categorical columns and below are the observations
* In the insured occupation category, the majority of the data is represented by machine operation inspectors, followed by professional specialties. The other insured occupations have nearly equal counts.
* Regarding insured hobbies, reading has the highest representation, followed by exercise. The other categories have average counts. Incident severity shows a high count for minor damages, while trivial damage has a significantly lower count compared to others.
* When accidents occur, authorities most frequently contact the police, with this category having the highest count. Fire has the second highest count, while Ambulance and Others have nearly equal counts. The None category has a much lower count compared to all others.
* For the incident state, New York, South Carolina, and West Virginia have the highest counts. In terms of incident city, almost all columns have equal counts

**Checking the Distribution of the dataset (numerical columns)**

A screenshot of a computer screen

AI-generated content may be incorrect.

* Above are the distribution plots for all the numerical columns. From these plots, we can observe the following:
* Most of the columns exhibit a normal distribution. However, some columns, such as capital gains and incident months, have a mean value greater than the median, indicating a left skew.
* Conversely, the capital loss column is right-skewed, as the median is greater than the mean. We will address and correct these skews using appropriate methods later on

### Bivariate Analysis

A screenshot of a computer code

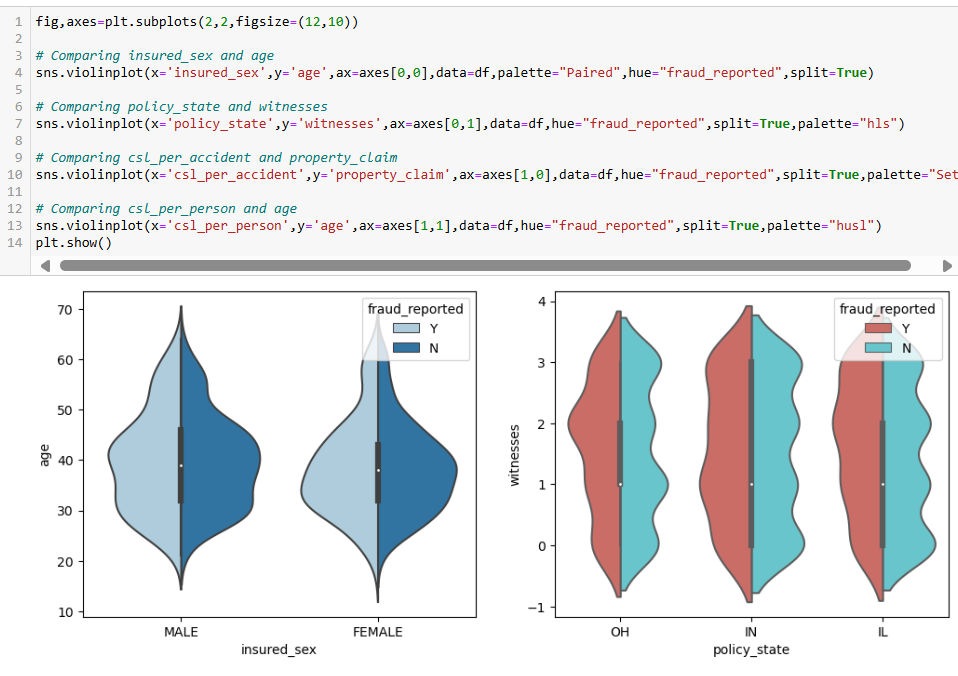
AI-generated content may be incorrect.

Output:

A group of graphs showing different colored dots

AI-generated content may be incorrect.

* From the above scatter plot, we can observe the following:
* There is a positive linear relationship between the age and month\_as\_customer columns.
* As age increases, the month\_as\_customer value also increases, with very few fraud cases reported in this scenario.
* In the second graph, we see a positive linear relationship where, as the total claim amount increases, the injury claim amount also increases.
* The third plot shows a similar trend: as the property claim amount increases, the vehicle claim amount also increases.
* In the fourth plot, the data is scattered, indicating no significant relationship between the features



Fraud reports are high for both males and females aged between 30-45. Individuals who own policies in the state of "IN" have a high incidence of fraud reports. Those with CSL per accident insurance and property claims ranging from 5,000 to 15,000 also report high fraud. Additionally, individuals aged 30-45 with CSL per person insurance are facing fraudulent reports

A screenshot of a computer

AI-generated content may be incorrect.

Most fraud reports are found when the total claimed amount is between 50,000 and 70,000. Fraud reports are also high when the claimed vehicle value is between 37,000 and 57,000. Additionally, fraud reports are frequent when the property claimed is between 5,200 and 8,500. Most fraud is reported when injury claims are between 5,000 and 8,000

A screenshot of a graph

AI-generated content may be incorrect.

A graph of blue bars

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a graph

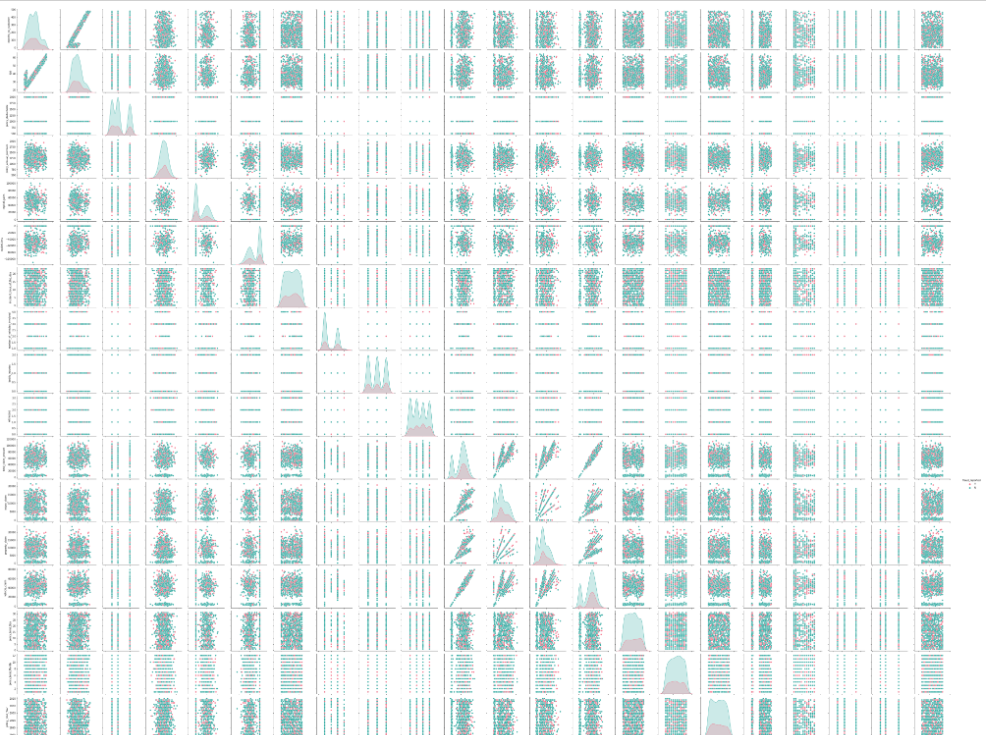
AI-generated content may be incorrect. A screenshot of a graph

AI-generated content may be incorrect. A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a computer screen

AI-generated content may be incorrect.





In the pair plot we can see the relation between each variable with respect to other variables.

#### **Observations**

* Fraud report is relatively high in the "OH" policy state.
* Individuals in executive-managerial positions have higher fraud reports compared to others.
* Fraud reports are high for customers with other relatives and very low for unmarried individuals
* Fraud reports are high among people whose hobbies include playing chess and cross fit.
* The fraudulent level is very low for individuals with high school education, whereas those with "JD" education have a high fraud report. People with higher insured education levels face more insurance fraud compared to those with lower education levels.
* Fraud reports are very high in multi-vehicle and single-vehicle collisions compared to other types.
* The fraud level is high in rear collisions, while the other two collision types have average reports.
* Fraud reports are high in incidents with major damage severity, whereas trivial damage incidents have lower fraud reports.
* If no police report is available, the fraud report is very high.
* Fraud reports are almost the same across all auto makes.
* There is no significant difference between the features in Vehicle\_Age vs fraud\_reported.
* Fraud reports are high for customers with other relatives and very low for unmarried individuals

Now we will Data Pre-processing by identifying the outliers and remove them. We will check the skewness of the dataset and remove the skewness.

### Pre-Processing Pipeline

#### **Identifying the outliers**

A screenshot of a computer screen

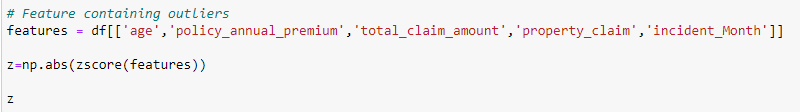
AI-generated content may be incorrect.

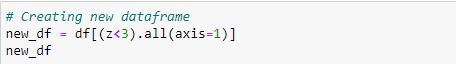
A group of green and white boxes

AI-generated content may be incorrect.

We can find the outliers in the following columns:

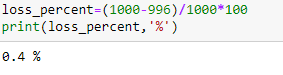
age policy\_annual\_premium , total\_claim\_amount , property\_claim and incident\_month.

These are the numerical columns which contain outliers. Remove the outliers in these columns using Zscore method.



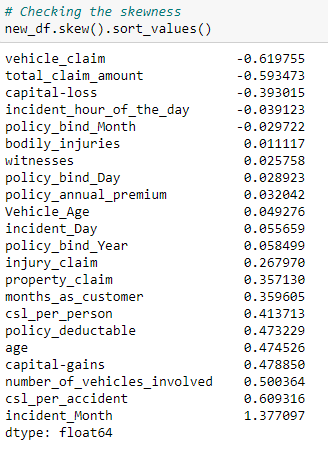
This is the new dataframe after removing the outliers. Here we have removed the outliers whose Zscore is less than 3.

**Percentage data loss:**

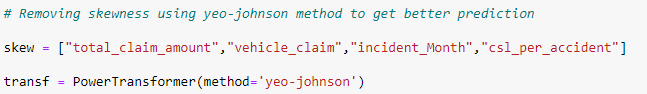


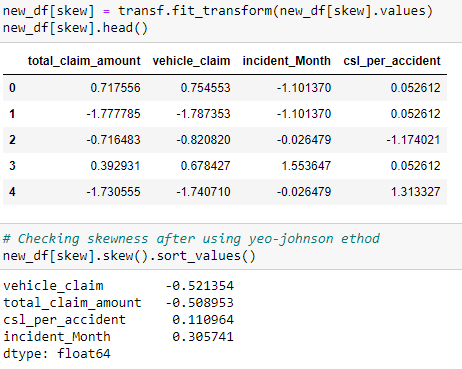
After removing the outliers, we are checking the data loss percentage by comparing the rows in our original data set and the new data set and 0.4% data loss is in the acceptable range.

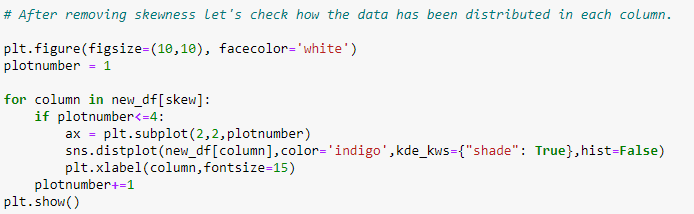
#### **Check skewness in the data:**

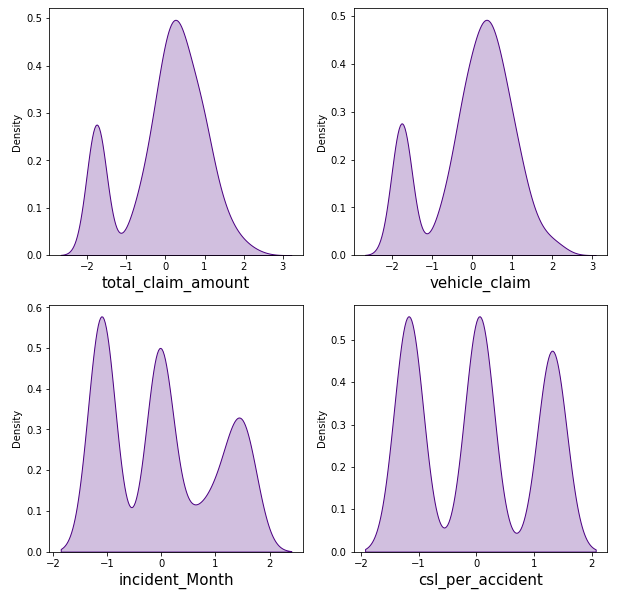


As we can see that skewness is present in the dataset, we will use yeo-johnson method to remove the skewness.

****

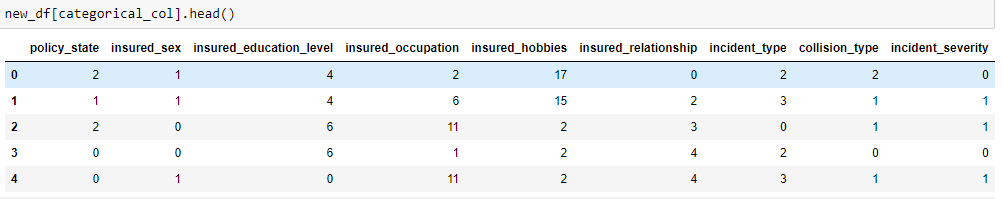
****





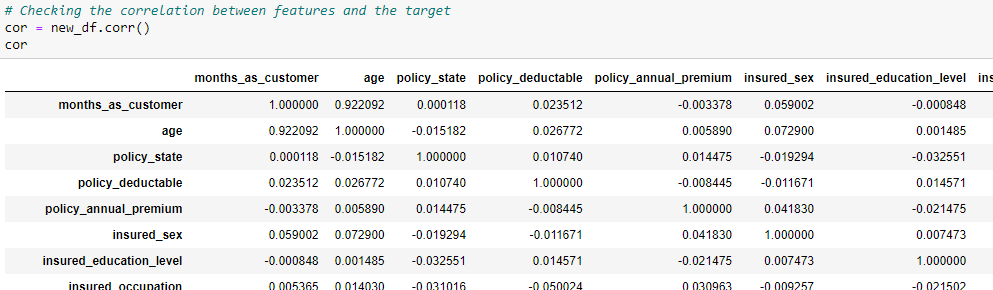
**Encoding the categorical columns using Label Encoding**

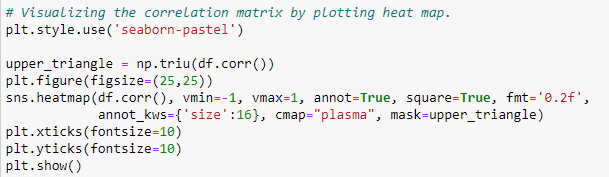
****

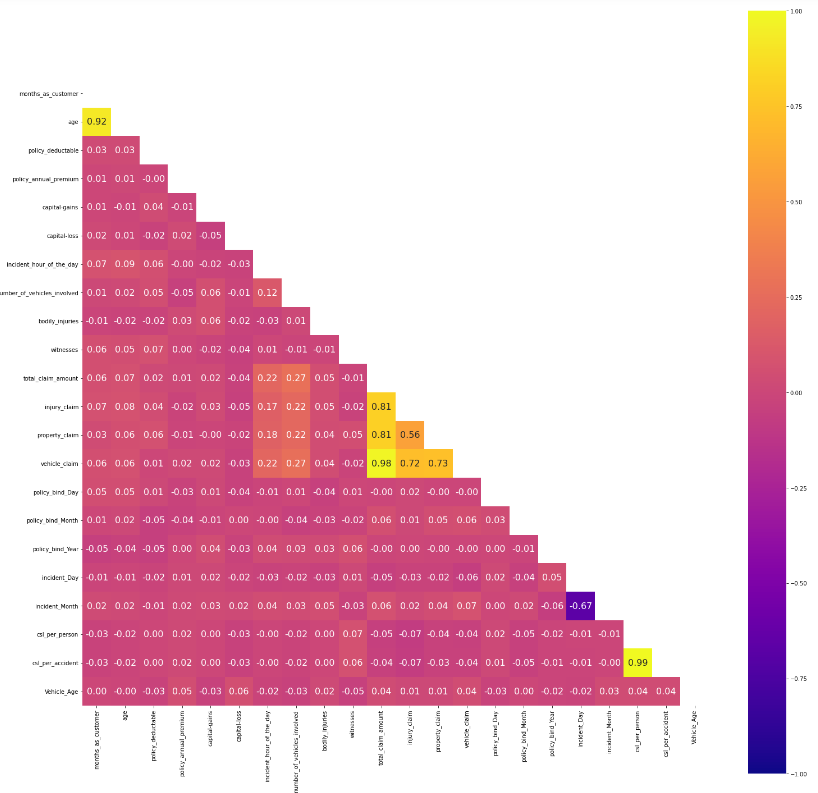


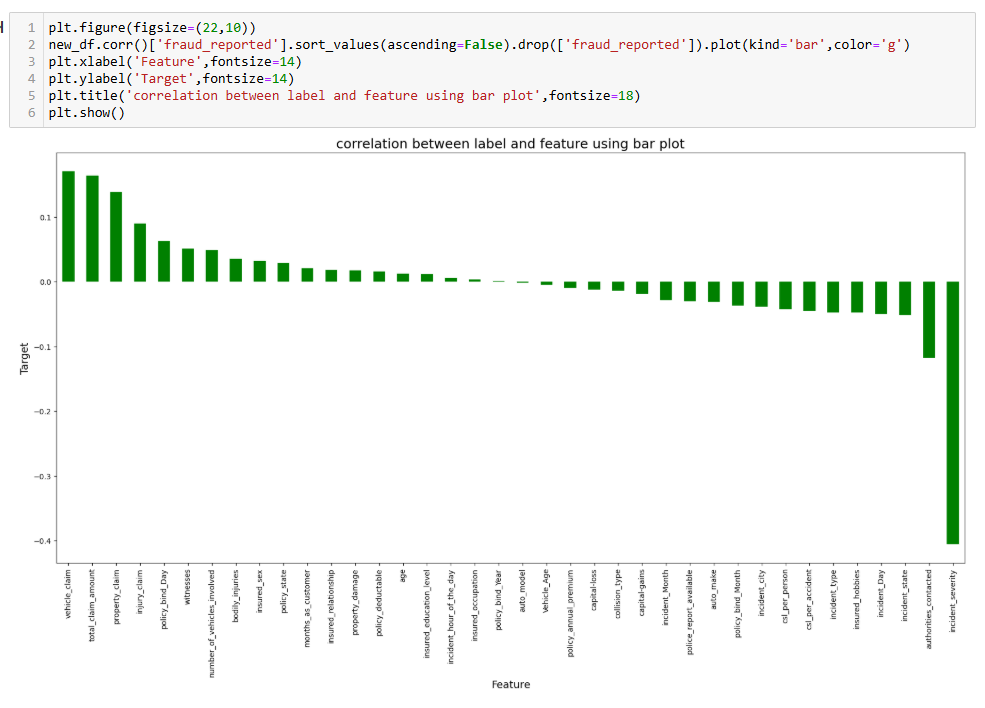
Now we have encoded the dataset using label encoder and the dataset looks like above.

Moving forward, to check the correlation between the feature and target and also the relation between the features using the heatmap.







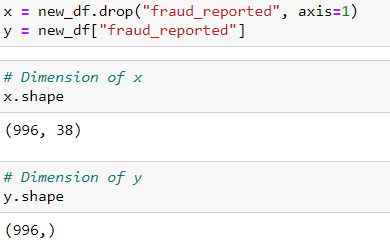


This heatmap shows the correlation matrix by visualizing the data. we can observe the relation between the features

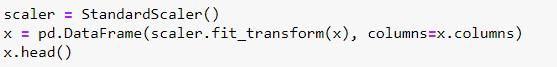
The heatmap contains both positive and negative correlations.

There is very little correlation between the target and the label. We can observe that most of the columns are highly correlated with each other that leads to multicollinearity problems. We will check the VIF value to overcome this multicollinearity problem.

#### **Splitting the dataset into Features and Target**



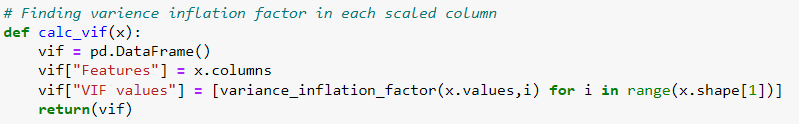
#### **Feature Scaling using Standard Scaler**

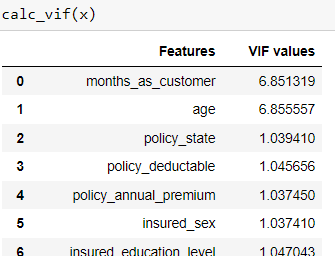


The data has now been scaled.

In the heat map we have found some features having high correlation between each other which means multicollinearity exists. So, let's check the VIF value to solve multicollinearity problem.

#### **Checking Multicollinearity**

****

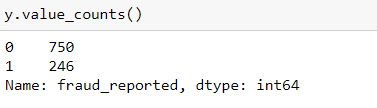
****

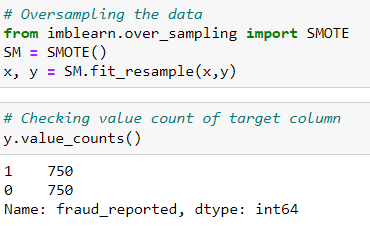
We can observe that some columns have VIF above 10 that mean they are causing multicollinearity problem. Let's drop the feature having high VIF value amongst all the columns.





As there is a huge difference between the count of 0 & 1, the data is not balanced. As this is a classification problem we will balance using oversampling method.





As we have treated the oversampling issue using SMOTE, now we can proceed with modelling.

A screen shot of a computer

AI-generated content may be incorrect.

# Model Selection

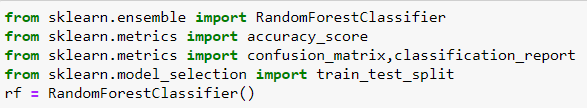
Several machine learning models were evaluated for detecting fraudulent claims. The models considered include:

* Logistic Regression
* Naïve Bayes
* Support Vector Machines (SVM)
* Decision Tree Classifier
* K Neighbors Classifier
* SGD Classifier
* Random Forest Classifier
* Extra Tree Classifier

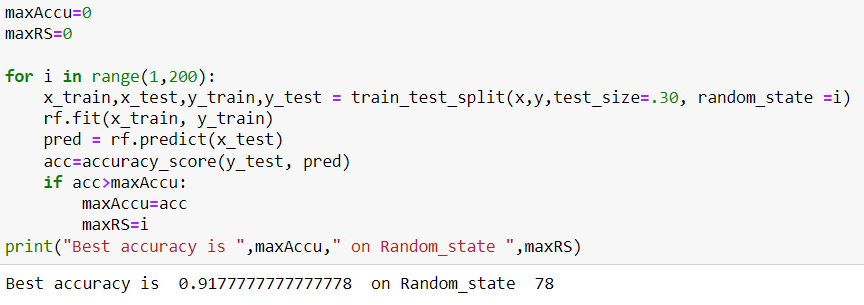
Each model was trained on the training dataset and evaluated using the testing dataset.

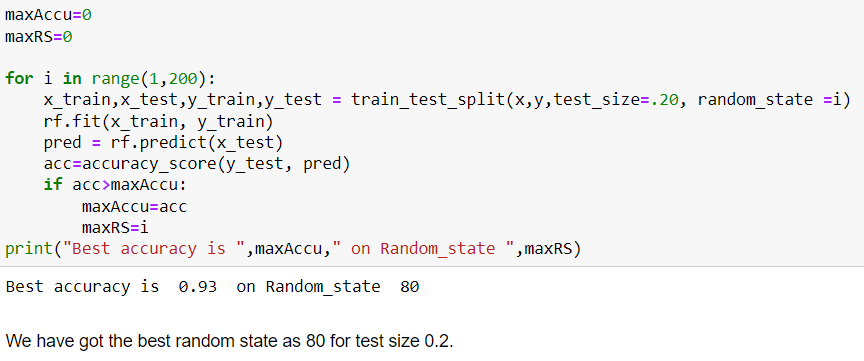
### Machine Learning

#### **Finding the best random state**



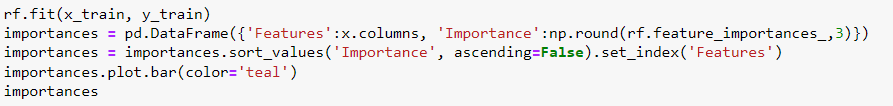
For Test size of 0.30



For Test size of 0.20

Here we have used the RandomForestClassifier to find the best random state and got an accuracy score of 93% at the random state of 80. Let’s use this random state to build our models.

#### **Feature importance bar graph**

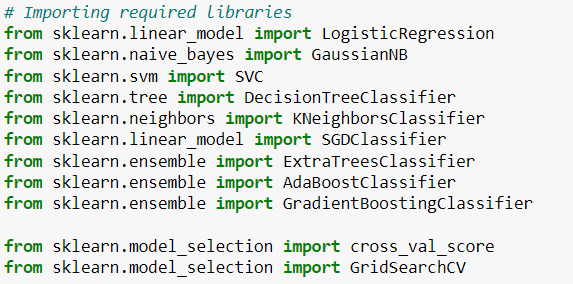
****

**A graph with text and numbers

AI-generated content may be incorrect.**

This bar plot shows us the importance of the features using random forest algorithm on predicting our Target variable.

We will import all the required libraries as shown below.



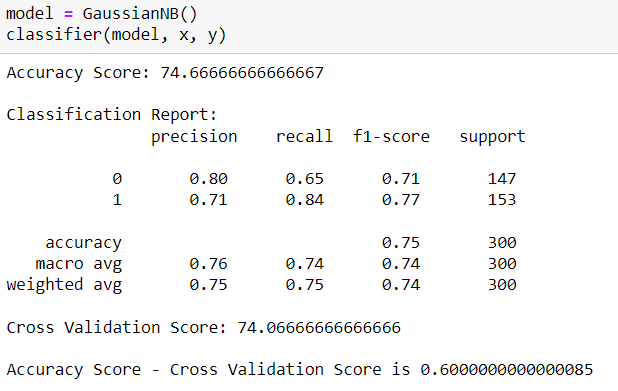
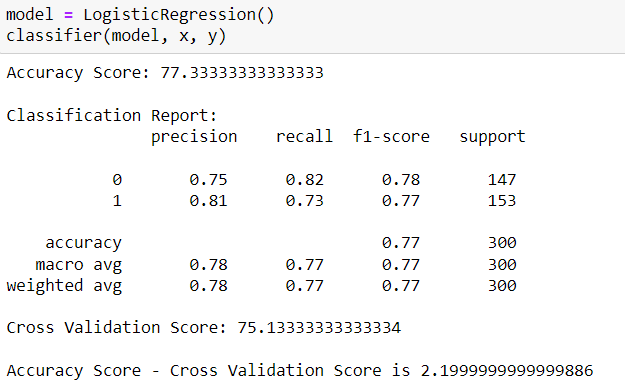
.

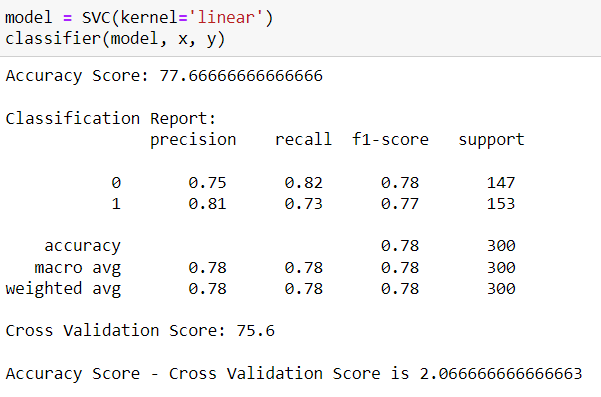
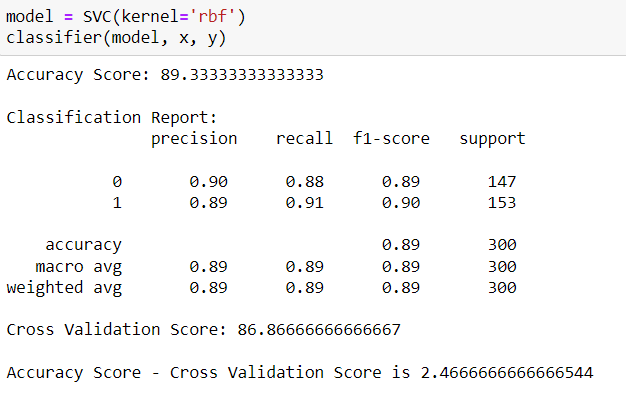
# Results

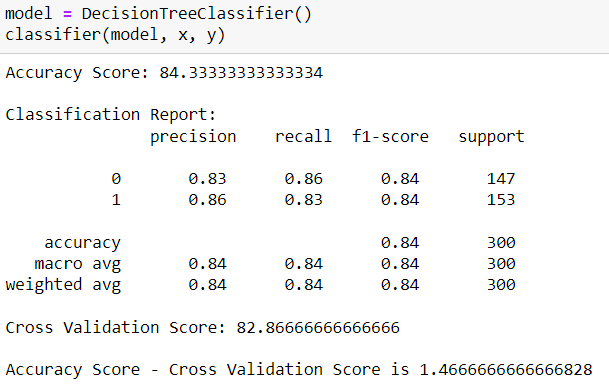
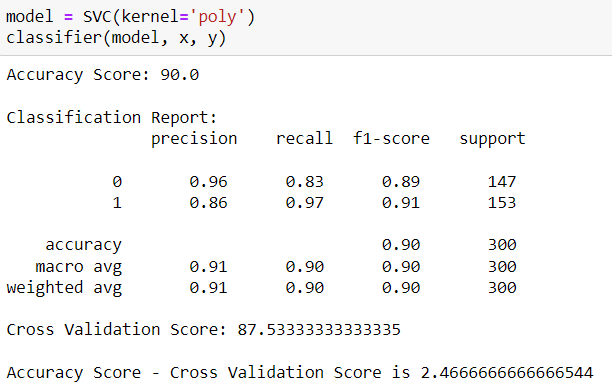
The results of the models were compared, and the best-performing model was selected based on the evaluation metrics. The extra tree classifier emerged as the top performer with high accuracy and a good balance between precision and recall.

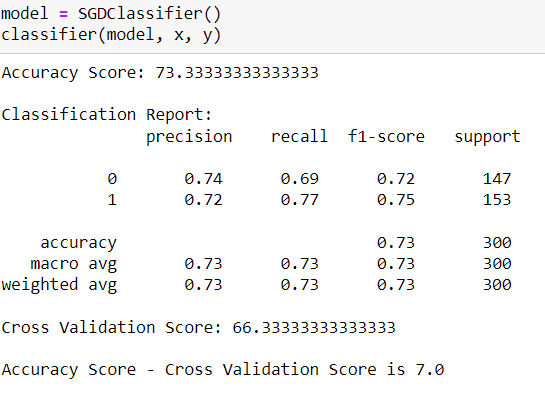
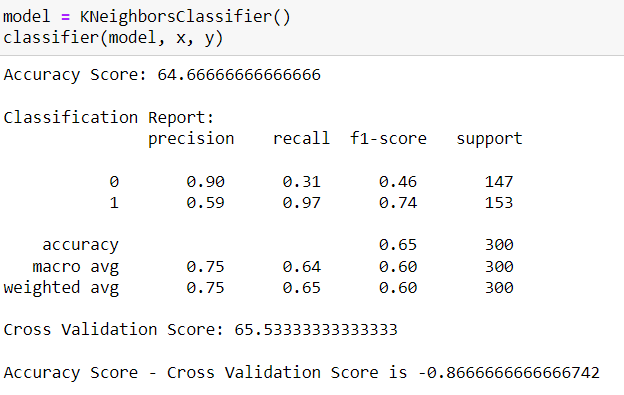
#### **Run Various Machine Learning Algorithms**

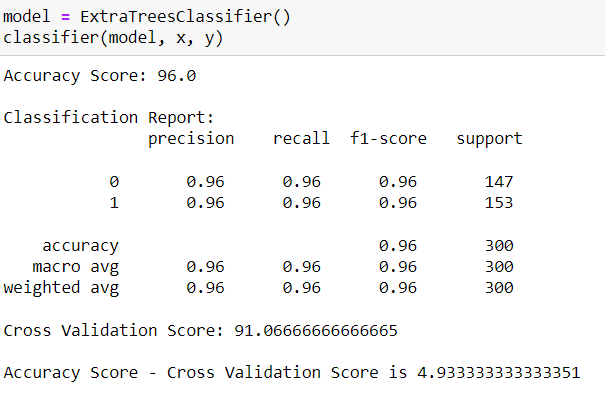
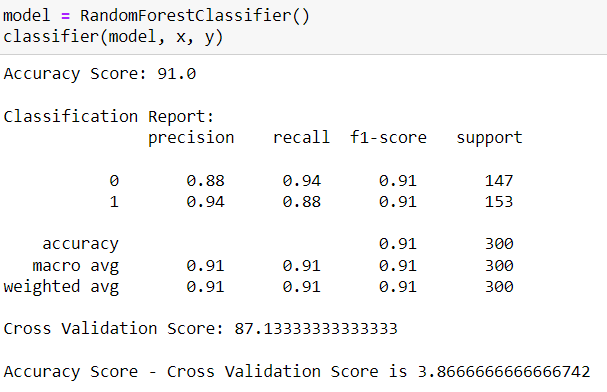
We will run various algorithms and check which has the highest Accuracy score, Cross Validation Score and (Accuracy Score - Cross Validation Score) for comparison between all the algorithms.

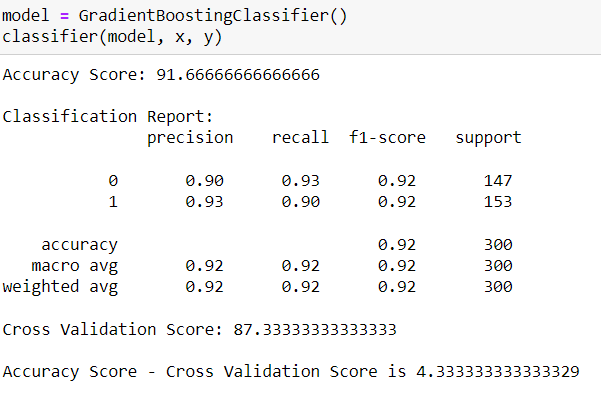
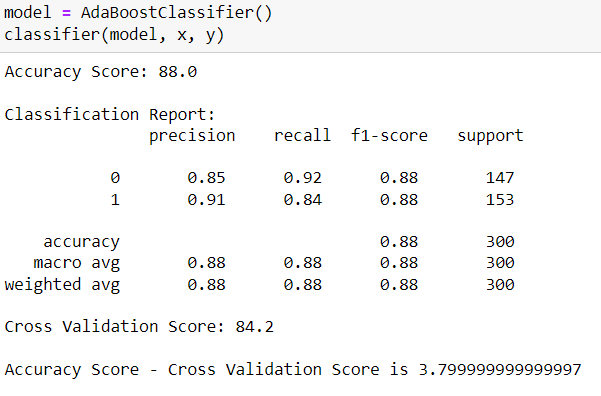












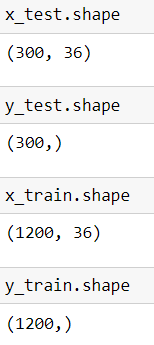
By referring all the above algorithms, we can see that ExtraTreesClassifier gives the best results since the (Accuracy Score - Cross Validation Score) is the least comparing others while having higher Cross Validation Score and the highest Accuracy Score comparing all the other models.

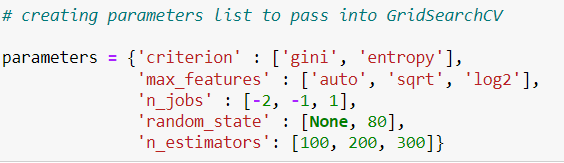
As we have found the best fit model, lets perform Hyper Parameter Tuning to improve the performance of the model.

#### **Hyper parameter tuning**

Creating train\_test\_split and checking the shape of the subsets.

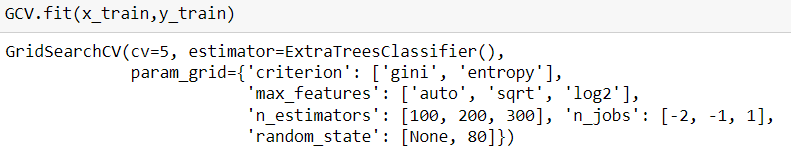




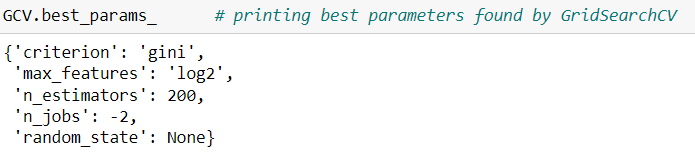
Creating a list of parameters to pass into the Grid Search CV.

Running Grid Search CV for ExtraTreesClassifier at cv = 5

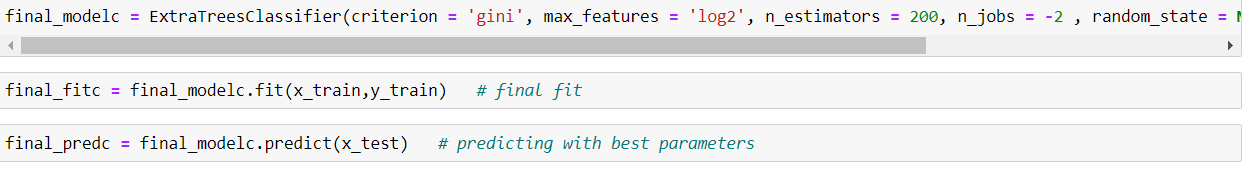




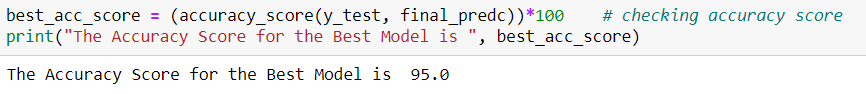
Get the list of the best parameters from Grid Search CV.



Here we got the best parameters, we will build our final model using these parameters.

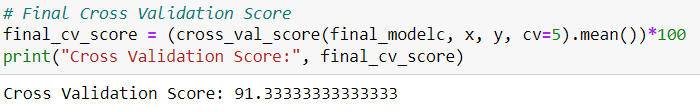


The Final Accuracy Score of the final Model.



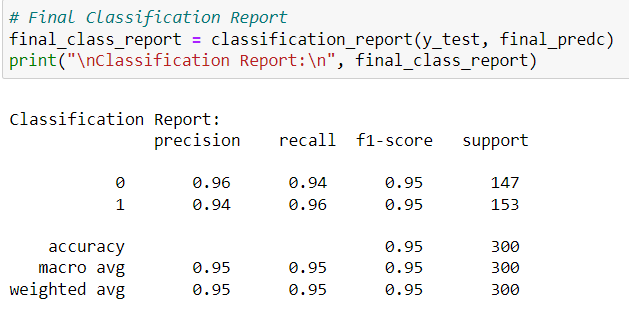
We successfully performed the Hyper Parameter Tuning on the Final Model.

The Final Cross Validation Score of the final Model.



We got final accuracy score of 95% and Cross Validation Score of 91.3333% which is good.

The Final Classification Report of the final Model.



# Evaluation Metrics

The performance of the models was evaluated using the following metrics:

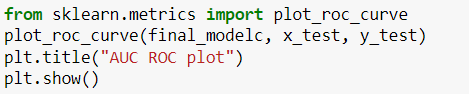
* Accuracy
* Precision
* Recall
* F1-Score
* Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

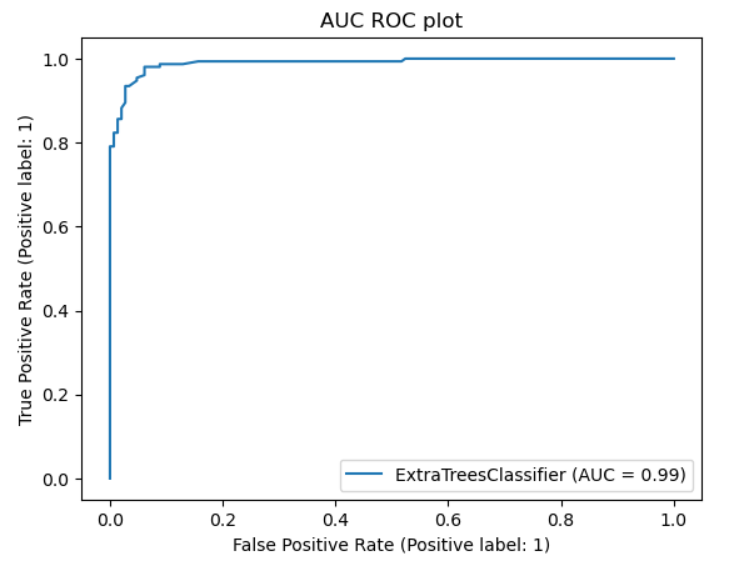
#### **Print Confusion matrix**

A screenshot of a computer

AI-generated content may be incorrect.

#### **AUC ROC curve**





We got a final accuracy score of 95% , Cross Validation Score of 91.3% and AUC score is 0.99 which is good.

#### **Save the model in pickle Format**

**A white background with black text

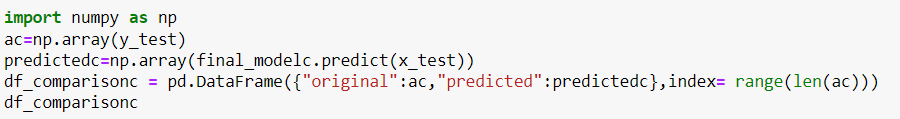
AI-generated content may be incorrect.**

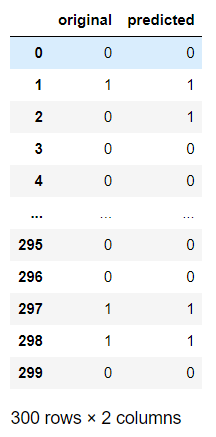
# Conclusion

Detecting fraudulent insurance claims is a complex yet essential task for the insurance industry. Through this case study, we demonstrated the use of various machine learning models and techniques to identify fraudulent claims effectively. The ExtraTreeClassifier model, with its robustness and high performance, proved to be an excellent choice for this purpose.

#### **Prediction Conclusion**

We will predict the "fraud\_reported" target column using the final model sending the “x\_test” set for predicting the “y\_test” and then compare the original “y\_test” and the predicted “y\_test” in a dataframe.



****

Hence predicted the "fraud\_reported" using the final Model and presented it as a data frame to compare the predictions.

Save the comparison file as a csv file.

