**INSURANCE CLAIM FRAUD DETECTION**



**INTRODUCTION**

Insurance fraud refers to a variety of unethical behaviors that a person may engage in , in order to obtain a favourable outcome from an insurance company. Given the variety of fraud types and the low ratio of known frauds in typical samples, detecting insurance fraud is a difficult task. When developing detection models, the cost of false warnings must be balanced against the cost of loss avoidance. Not all claims are analysed thoroughly.

All this steps are involved along with the algorithm on the partial training datasets and then it is tested on the test datasets, finally it is then examined by some random splits .The data in the datasets are handled by certain rules which are as follows:.

**1. Problem Definition.**

**2. Data Analysis.**

**3. EDA Concluding Remark.**

**4. Pre-Processing Pipeline.**

**5. Building Machine Learning Models.**

**6. Concluding Remarks.**

**PROBLEM DEFINITION:**

Business case: 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.

**TASK:**

In this example, 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.

**Machine learning in Fraud detection :**

Artificial Intelligence includes Machine Learning (AI). Artificial Intelligence's goal is to construct a computerised system that can perform complicated analysis and not only replace but also improve on human input. Machine Learning uses artificial intelligence to “give” systems the ability to learn and grow from their experiences without the need for additional programming. Fraud detection is a knowledge-intensive process that determines if a transaction or claim is authentic or fraudulent. Deep anomaly detection is a common type of machine learning used in the insurance sector. Anomaly detection works by examining the customer's normal, authentic claims and creating a model of how a typical claim appears. The model is then used to analyse big datasets. This method of anomaly detection can also be used to build other components of artificial intelligence. Predictive analytics is one such , that can be used to design the program.This program not only design and analyse the features it also identify the fraudulent claims.

**Objective :**

The techniques in the machine learning is to improve the accuracy of detection on various imbalanced datasets. With the goal of generating higher predictive performance, the impact of feature engineering, feature selection, and parameter tweaking is also analyzed.

Predictive analysis involves certain steps to obtain accuracy detection

* **Training**
* **Testing**
* **Validation**

All this steps are involved along with the algorithm on the partial training datasets and then it is tested on the test datasets, finally it is then examined by some random splits .The data in the datasets are handled by certain rules.

**Summary :**

The difficulty with machine learning fraud detection is that frauds are significantly less frequently than legitimate insurance claims. This type of data are called imbalanced data. Model is proposed to find accuracy of fraud detection on different algorithm to handle those imbalanced data. The best out of that are fitted in to the model to predict the accuracy.

The final model fitted DecisionTree Classifer which has obtained **86.67 %** score and its f1score is 86% which better than the other Classification Models .

**About Dataset :**

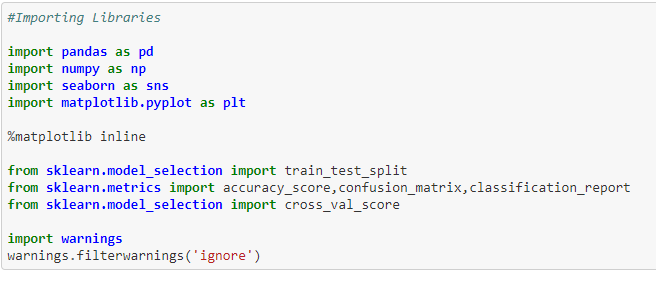
The dataset is about the auto claim insurance dataset along with the customer details for which they have claimed their insurance. Model is predicted whether the claim is authentic or fraud

**In this blog, I’m going to create a few ML models using Scikit-learn library and we’ll compare the accuracy for each of them.**

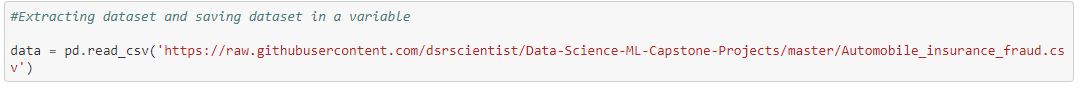
**In this project we have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and also visualized it using heatmap.**

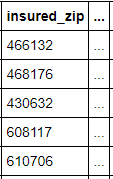
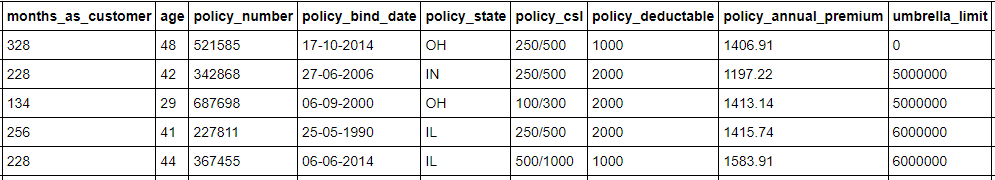
**DATA ANALYSIS**

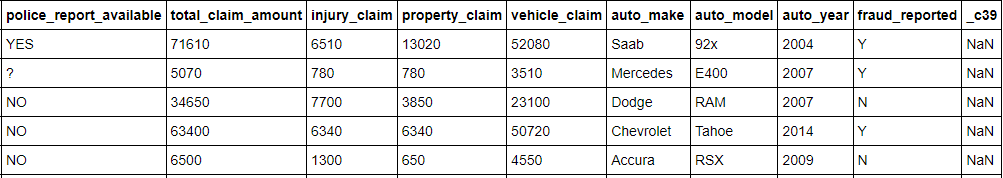
**Importing Libraries:**

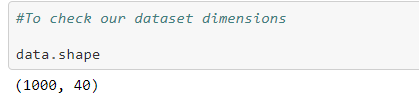


**Extracting dataset**



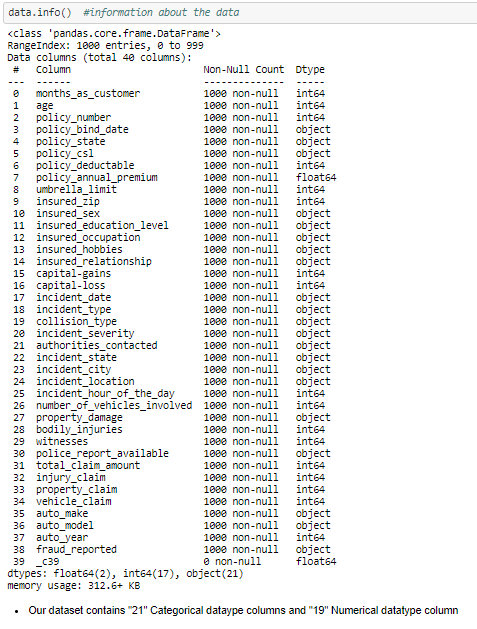






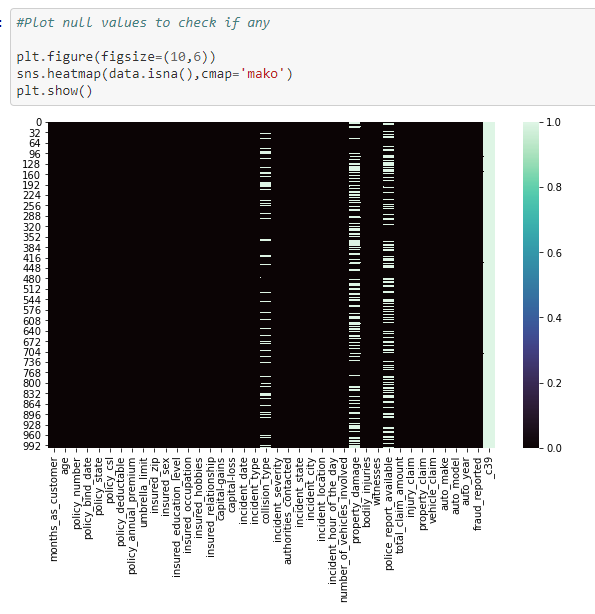
The given dataset contains 1000 rows and 40 columns. Using this dataset we will be training the Machine Learning models on 80% of the data and the models will be tested on 20% data.

In this paper, the response variable, Fraud-Detected, is a binary variable (whether the Insurance claim was fraud or not). Therefore, the Classification model is a suitable technique to use because it is developed to predict a binary dependent variable as a function of the predictor variables. The logistic regression model is widely used in credit default studies where the dependent variable is binary Although the given dataset doesn’t have any null value, we can expect outliers and un- realistic values for certain variables.



***To check the total null values in all the columns individually***

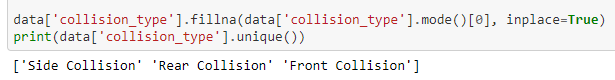
Before checking null values I have replaced the unique value ‘?’ with NAN, so that the model can detect if there any null/unique unknown vales present in our dataset.



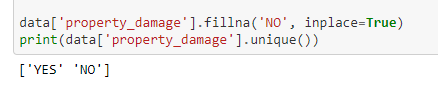
* Coumn: "\_c39" contains only null values for the entire column, so, let's remove this column.
* *We have some null values in the below column:*

‘collision\_type’, ‘property\_damage’, ‘police\_report\_available’

# let's check the column collision\_type as it contains 178 null values. # collison type column contain 3 unique values namely "Rear collision", "side collision", "front collision". we will replace the null value with most common collision type.



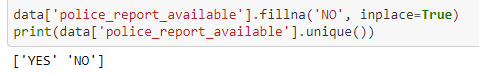
# lets check for property\_damage column as it contains 360 null values.# property damage column contains only 2 values "YES" or "NO".# we can replace the null values with "NO", assuming no property damage.



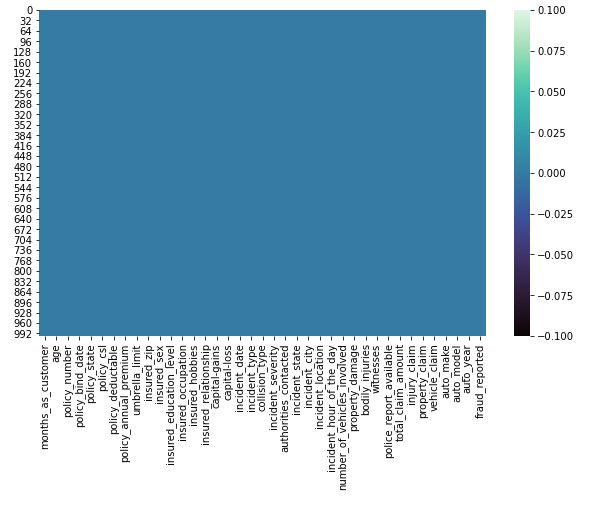
# lets check for the column police report available as it contaains 343 values.

# police\_report\_available contains only 2 values "YES" or "NO".# we can replace

the null values with "NO", assuming there is NO police\_report\_available.



**Let’s recheck the null values by plotting the heatmap**



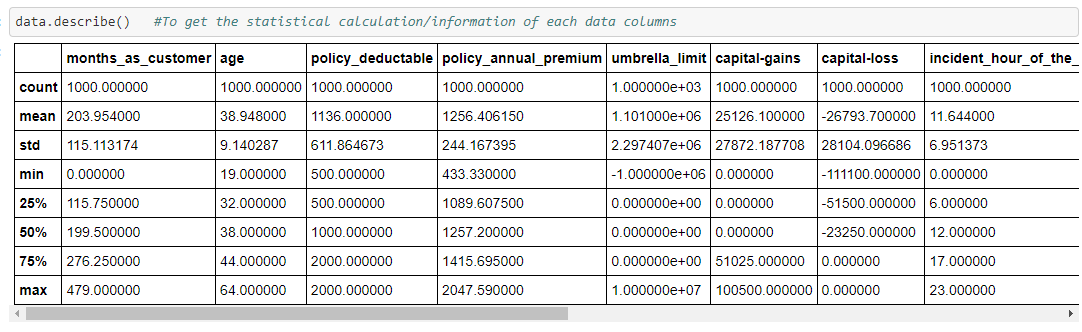
#### **Our dataset contains no null values. Hence, let's proceed further**

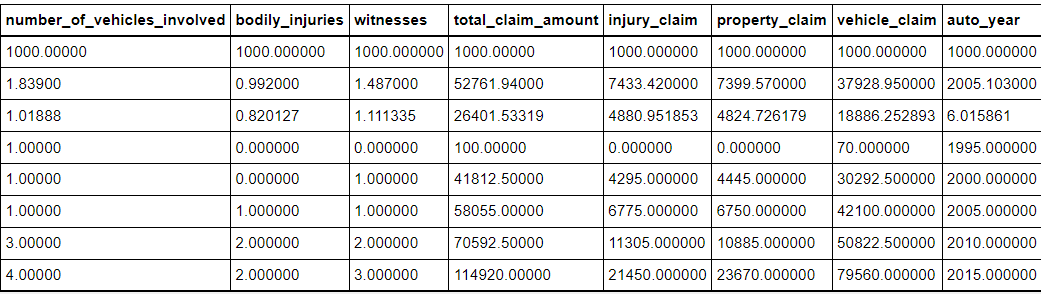
* The columns: "policy\_number" and "incident\_location" has unique 1000 rows and our total rows are 1000.
* Hence it means in every single row, the "policy\_number" and "incident\_location" are unique independently

Conclusion: We can drop these columns from our dataset

* Column: "insured\_zip" contains 995 unique values out of total 1000 rows, let's remove this column, because it would not affect in our data modeling (another reason it is the postal zip code, with almost all unique features which doesn't relate in # confirming the fraud opportunites)

**Statistical information of each column:**



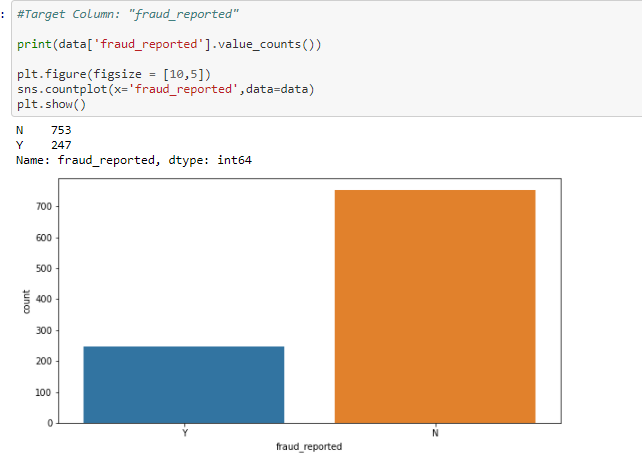


* 50 % Of insurance claims has "0 Capital gains"
* 'total\_claim\_amount','injury\_claim', 'property\_claim' has a large difference between their Mean and 50% median
* We have outliers and skewness which needs to be removed

# **Exploratory Data Analysis (EDA)**

Exploratory data analysis can help detect obvious errors, identify outliers in datasets, understand relationships, unearth important factors, find patterns within data, and provide new insights. Let us explore our dataset features and visualize it

### **Univariate Analysis**



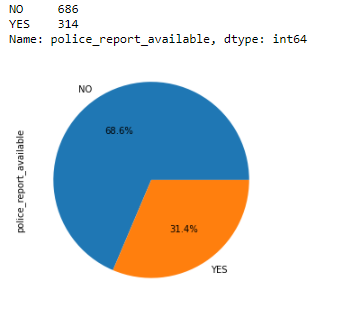
#### **Observation:**

- Here, "753" frauds were not reported that has larger count when compared to "247" frauds reported

- Target column: "fraud\_reported" has imabalance data

- We need to use Oversampling / Undersampling method to make our target data balanced

**“police\_report\_available"**

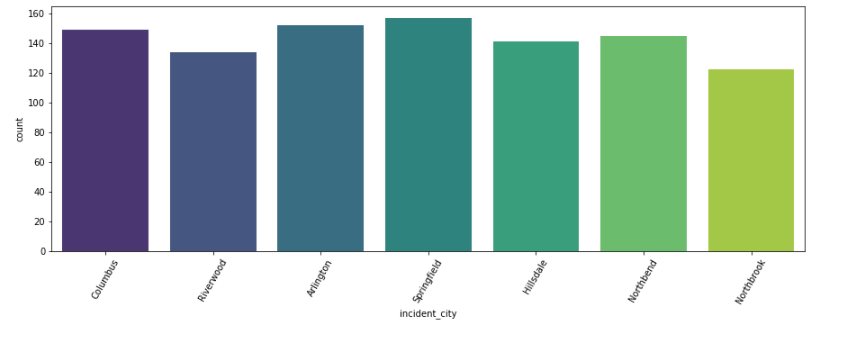


#### **Observation:**

- Only 31.4% police\_report was filed and was available to check

- Rest 68.6% police\_report was not available

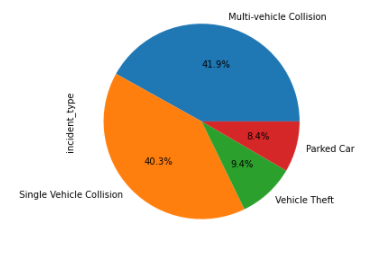
**“incident\_city"**



### **Observation:**

* Most of the Insurance are claimed from almost all the cities
* City "Riverwood" and "Northbrook" are the cities where less Insurance claim counts

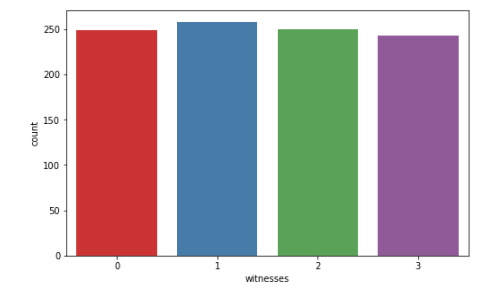
**“incident\_type"**



### **Observation:**

* Insurace claims with Incidents are maximum due to "Single and Multi Vehicle Collision"
* Less incident cases are come up for "Parked Cars" amd "Vehicle Thefts"

**“witnesses"**



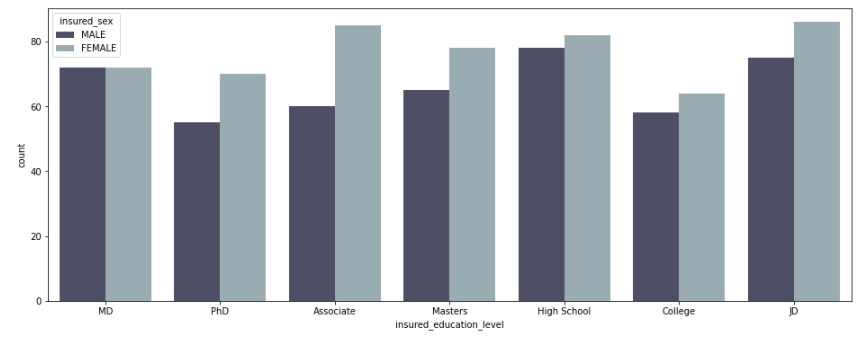
#### **Observation:**

* Number of Insurance claims with Zero, one, two and three Witnesses are almost equal

### **Bivariate Analysis:**

### Let us check the relation betweentwo variables and explore them using various plotting methods.

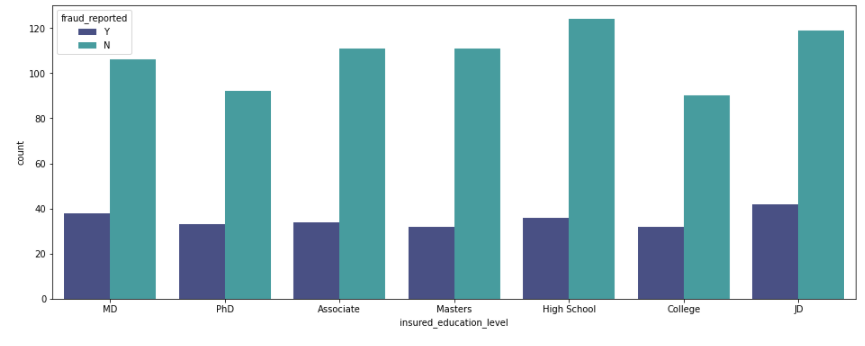
***Relation between "insured\_education\_level" and "insured\_sex"***



#### **Observation:**

* In all the Education-level, Female candidates have applied for Insurance claims
* Except the Education-level, "MD", No. of Males and Females are equal who applied for insurance claim

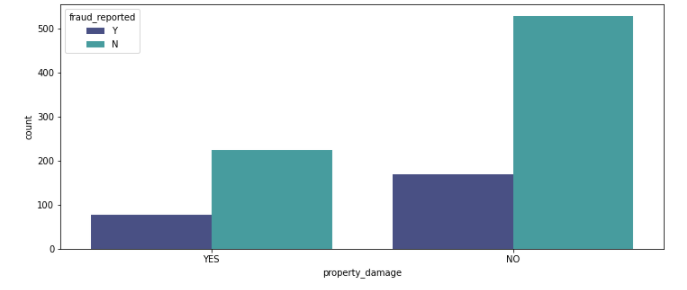
***Relation between "insured\_education\_level" and "fraud\_reported"***



#### **Observation:**

* In all the "insured\_education\_level", most the highest number has "No Fraud Cases". It is a good sign
* But, we can't ignore that in all the "insured\_education\_level" there is equal number of "Frauds Reported"

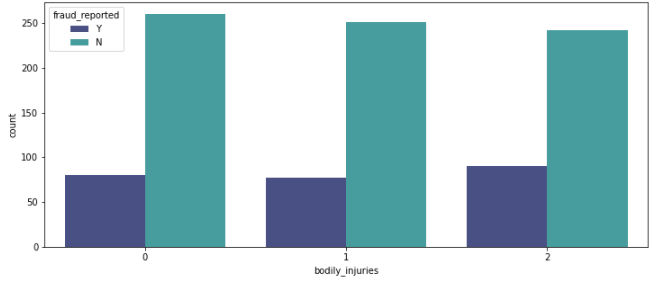
***Relation between "property\_damage" and "fraud\_reported"***



#### **Observation:**

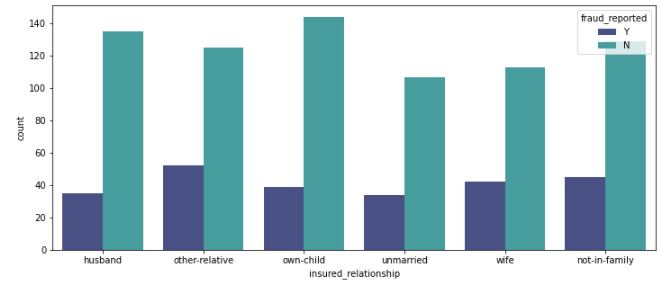
* Most "Property\_damage" cases are not reported as fraud
* And the cases without "property\_damage" also has large number of frauds not reeported

***Relation between "bodily\_injuries" and "fraud\_reported"***

**Observation:**

* In all the "bodily\_injuries" irrespective zero, one and two, most of them have "No Fraud Cases". It is a good sign
* But, we can't ignore that in all the "bodily\_injuries" there is equal number of "Frauds Reported"

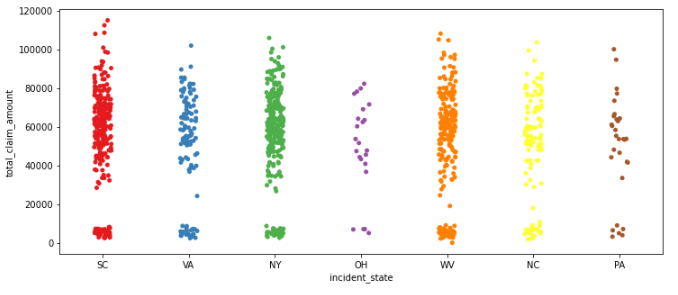
***Relation between "insured\_relationship" and "fraud\_reported"***



#### **Observation:**

* In all the "insured\_relationship" category, the highest number hasmost of them "No Fraud Cases". It is a good sign
* But, we can't ignore that in all the - In all the "insured\_relationship" category, there is equal number of "Frauds Reported"

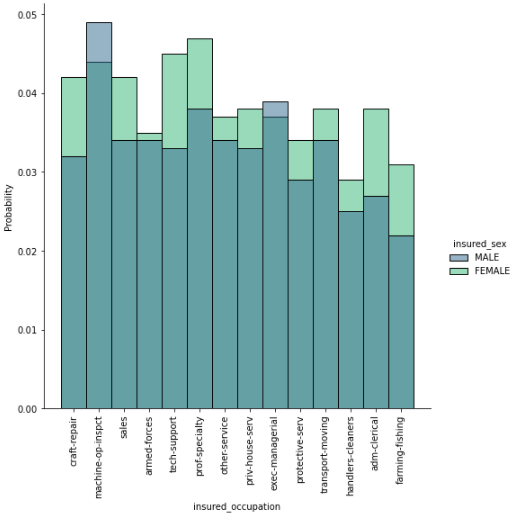
***Relation between "incident\_state" and "total\_claim\_amount"***



#### **Observation:**

* Most of the incident\_state has applied for Insurance total\_claim\_amount.
* Cities: "OH" and "PA" can be seen with less number of claims

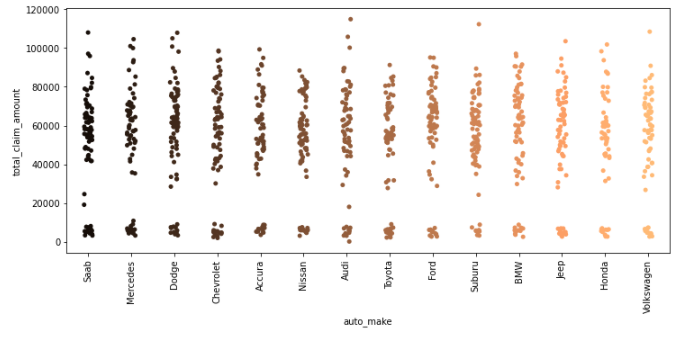
***Relation between "insured\_occupation" and "insured\_sex"***



#### **Observation:**

* Insurance claims are high in number for "Insured\_occupation: Machine-op-inspect" where there are only the male candidates who has applied

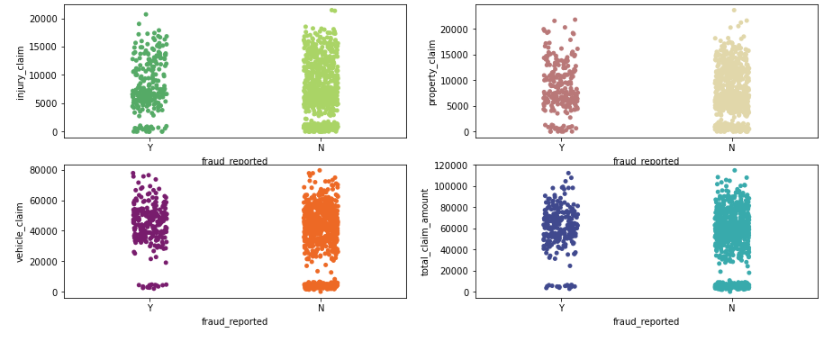
***Relation between "auto\_make" and "total\_claim\_amount"***



#### **Observation:**

* Every "auto\_make" data has been involved in total\_claim\_amount for insurance
* “Audi” auto\_make has the highest claim amount
* “Nissan” is the lowest

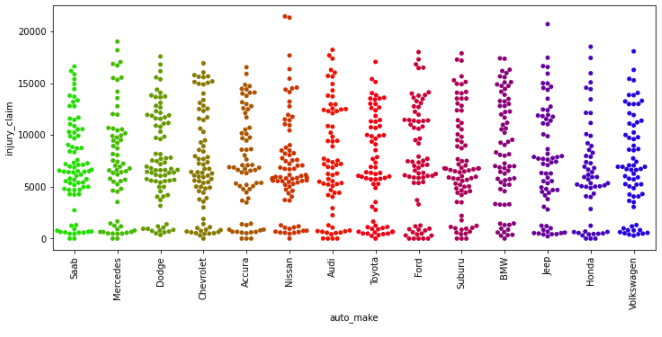
***Relation between "injury\_claim",”property\_claim”, “vehicle\_claim”, “totla\_claim\_amount” with "fraud\_reported"***



#### **Observation:**

* All the four columns: "injury\_claim", "property\_claim", "vehicle\_claim" and total\_claim\_amount" are affecting our target column "Fraud\_Reported". These 4 columns are important in predicting the model

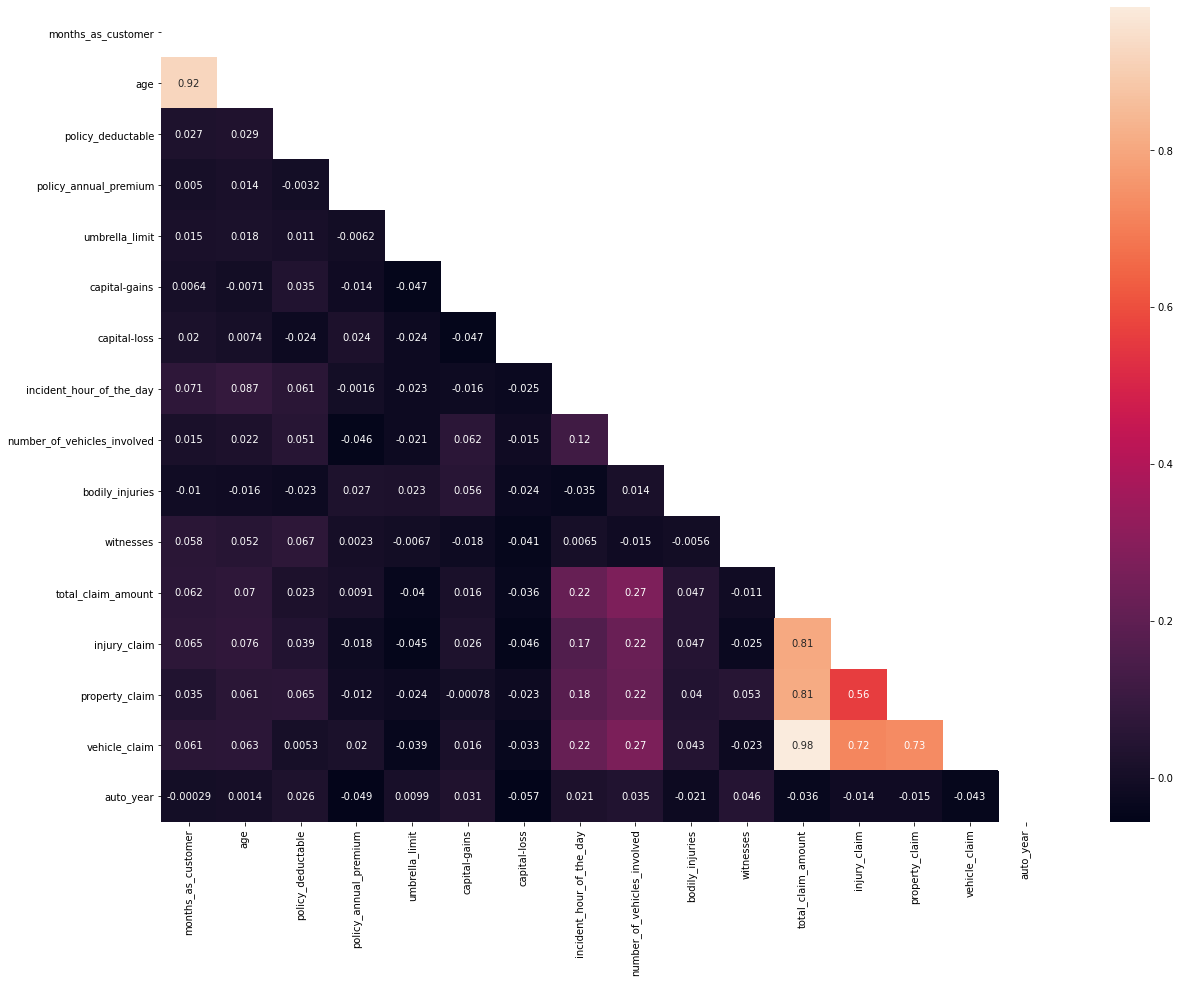
***Relation between "auto\_make" and "injury\_claim"***



#### **Observation:**

* "Nissan" of the data column: "auto\_make" has made the highest injury\_claim
* Then, "jeep"

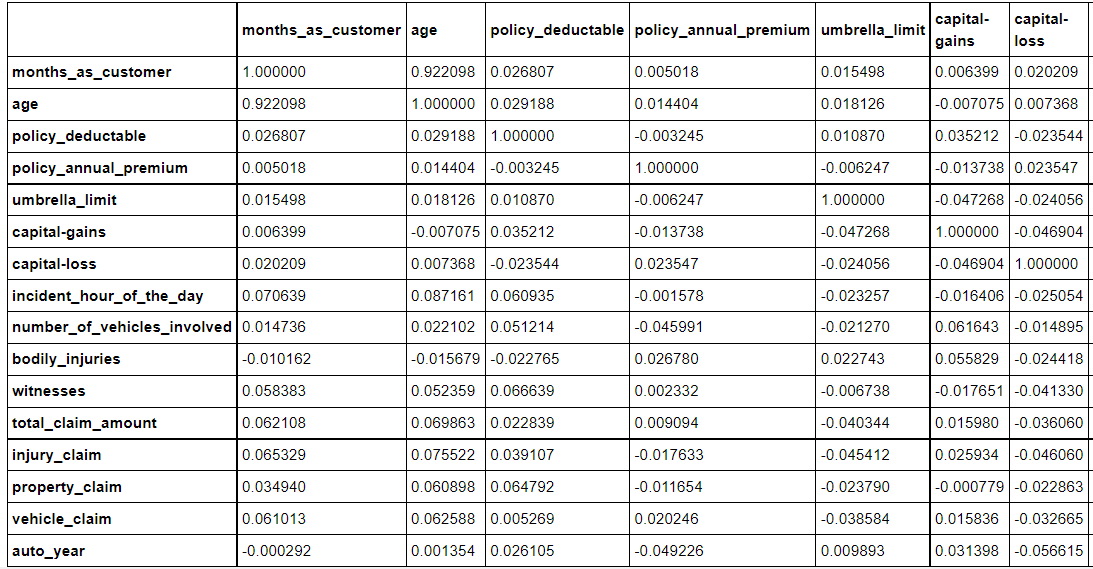
**Correlation matrix using heatmap**

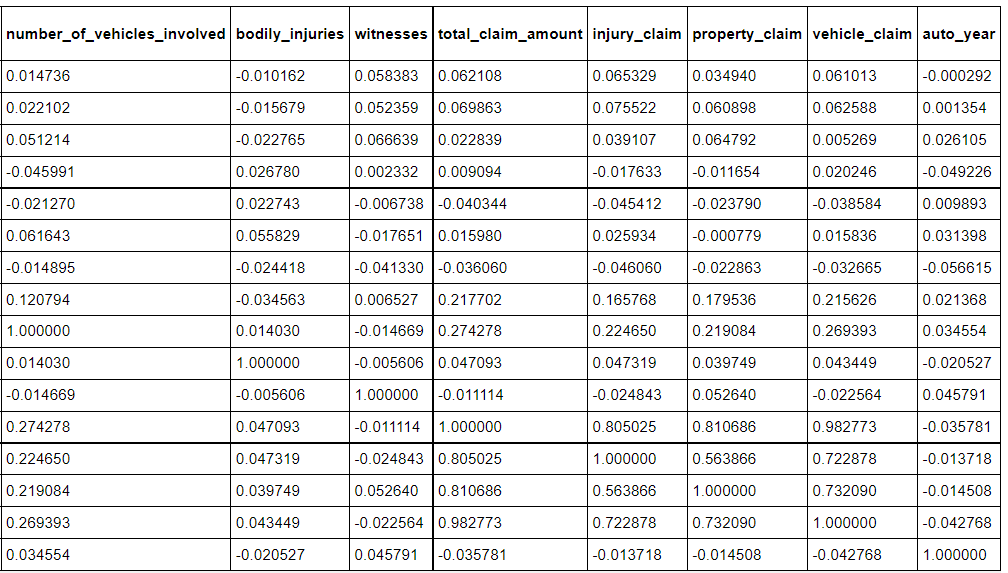
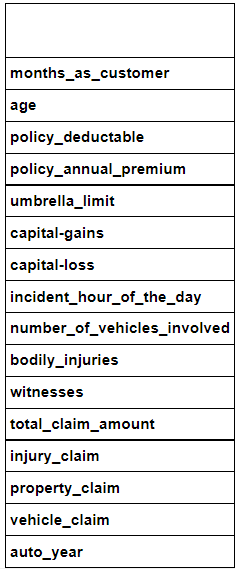


- “Vehicle\_claim”, “age” is highly correlated then, “property\_claim” and “injury\_claim” are largely correlated

PRE-PROCESSING PIPELINE:

**CORRELATION TABLE**

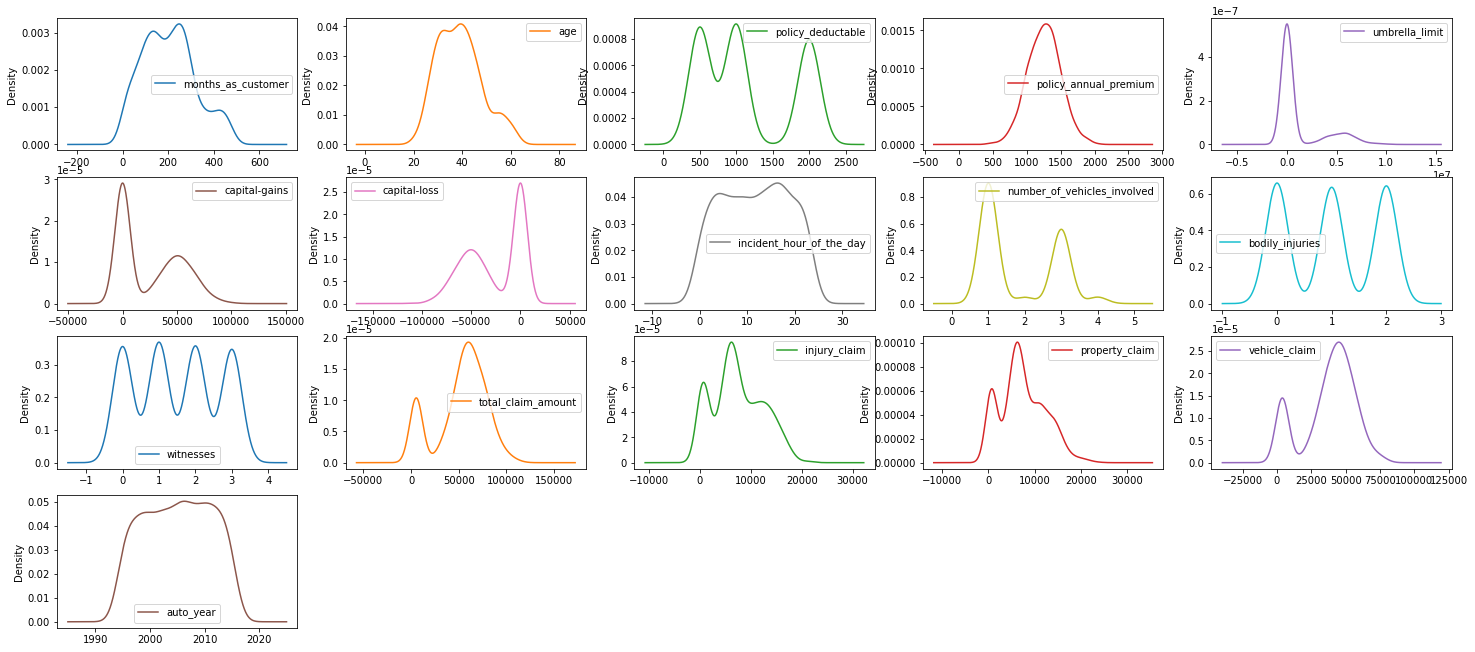




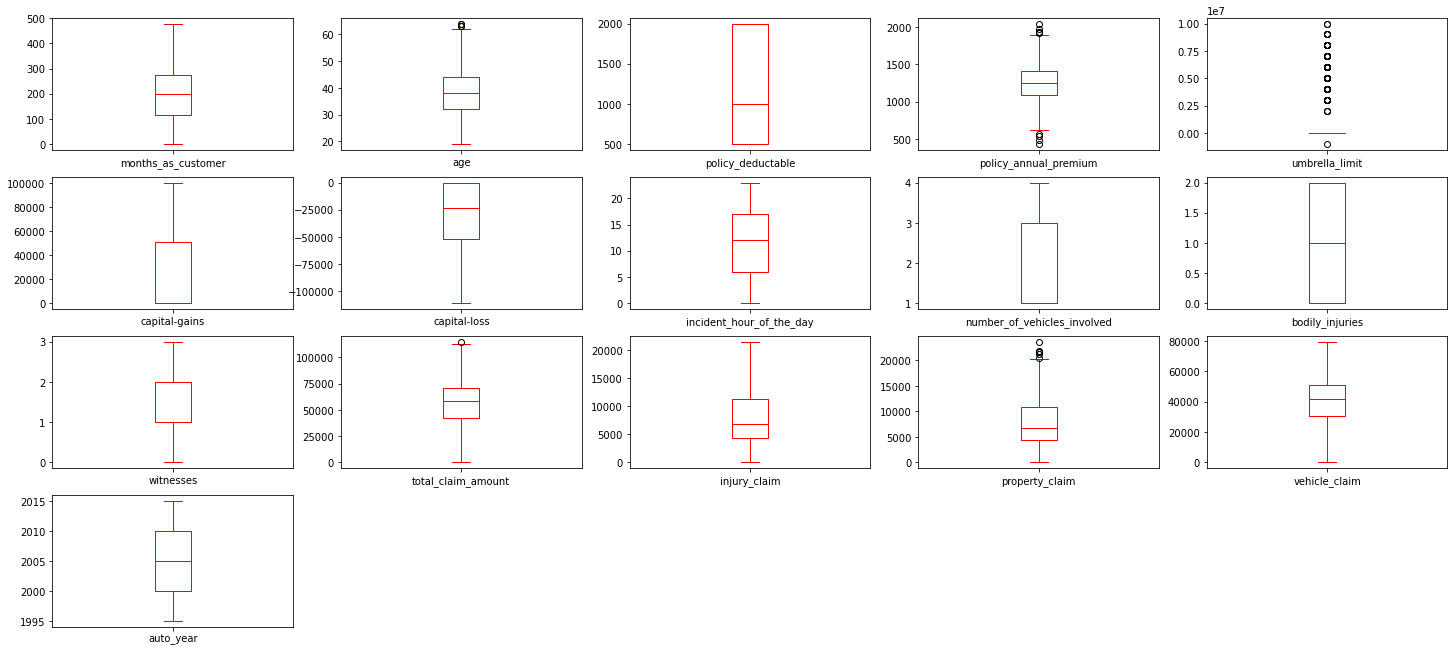
#### **Observation:**

* Vehicle\_claim and total\_claim\_amount has the high correlation with value: "0.98

**Poltting Normal distribution to check the skewness of our data columns**



**Poltting Box plot to check the outliers of our data columns**



*#We found the below Columns with Outliers by visualizing the above diagrams*

**outliers = ['age', 'policy\_annual\_premium', 'total\_claim\_amount',**

**'property\_claim']**

#### A**pplying IQR Method**

Shape - Before and After:

Shape Before : (1000, 36)

Shape After : (996, 36)

Percentage Loss : 0.4

#### **Applying z-score Method**

Shape - Before and After:

Shape Before : (1000, 36)

Shape After : (996, 36)

Percentage Loss : 0.4

The percentage loss by applying both the techniques are equal, so, let's proceed by using any one technique. Let’s choose z score method.

**SKEWNESS**

months\_as\_customer 0.359605

Age 0.474526

policy\_deductable 0.473229

policy\_annual\_premium 0.032042

umbrella\_limit 1.800271

capital-gains 0.478850

capital-loss - 0.393015

incident\_hour\_of\_the\_day - 0.039123

number\_of\_vehicles\_involved 0.500364

bodily\_injuries 0.011117

witnesses 0.025758

total\_claim\_amount - 0.593473

injury\_claim 0.267970

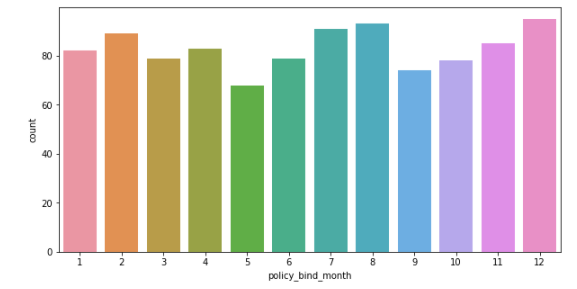
property\_claim 0.357130

vehicle\_claim - 0.619755

auto\_year - 0.049276

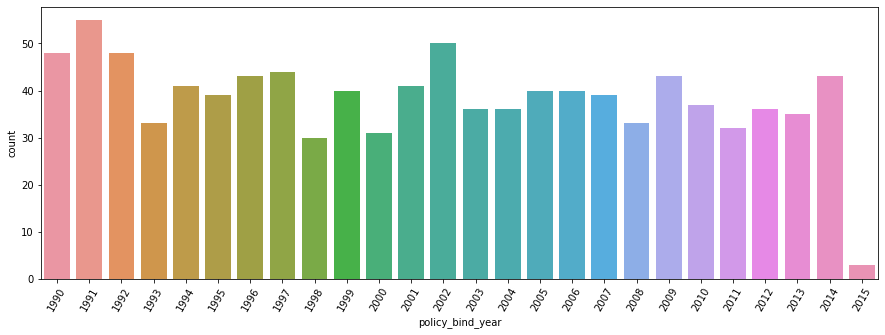
* We could see most of the skewness is present in "umbrella\_limit"
* There are skewness in "total\_claim\_amount" and "vehicle\_claim" but if we try reducing using any technique then, the skewness gets increased, so, let's leave these 2 columns as it is
* "umbrella\_limit" is with the ordinal data , so we will ignore the skewness

**Extracting month from “policy\_bind\_date” column**



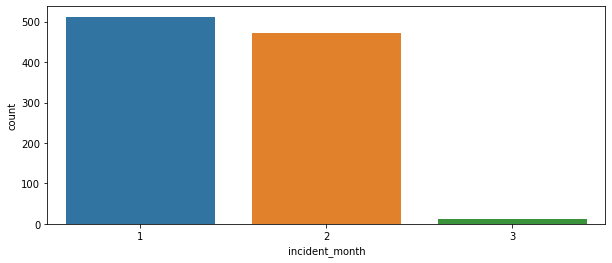
- “May” has the lowest number of “policy bind month”

**Extracting “year” from “policy\_bind\_date” column**

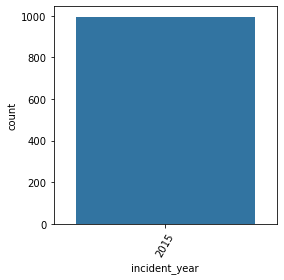


- “2015” year has the lowest number of policy binded

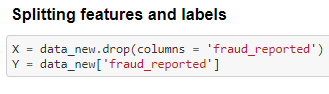
**Extracting “month” from “incident\_month” column**



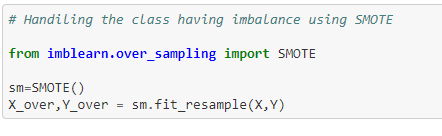
**Extracting “year” from “incident\_month” column**

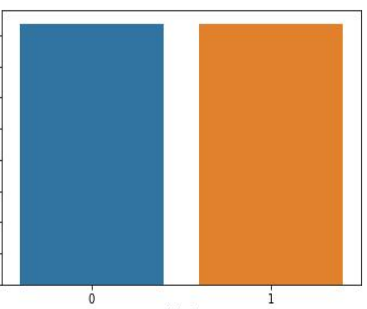


***The above extracted “year” column is removed, because the whole column contains the same year: "2015"***



### **Balance the Imbalanced class**





Our dataset label is now balanced by using over sampling method

Further, before build the model we will have to split the data to test and train.

The best possible way to split the data is by finding the best random state to split

and the benefit is that we can control over fitting up to certain extent before even

building the model.

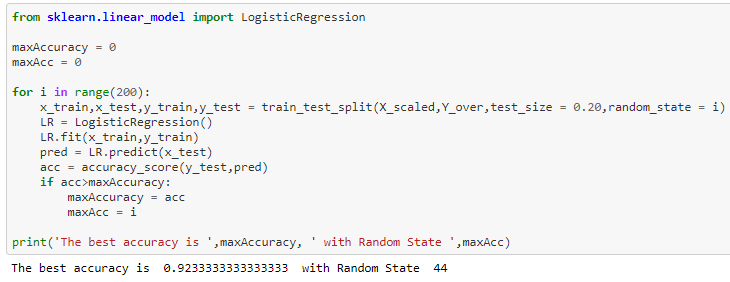
We are trying to match the accuracy score of the training data set and the test

dataset, which ever split (**random state**) satisfies the condition

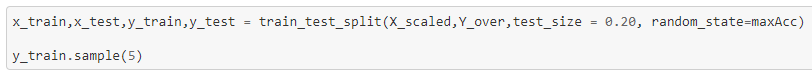
(**accuracy score of training dataset = accuracy score of testing dataset**).

We’ll take the same random state to split the dataset and build the model.

### **Finding the Best Random State**



**Split train test data**



MACHINE LEARNING MODELS

In the machine learning world, model training refers to the process of allowing a machine learning algorithm to automatically learn patterns based on data.

**Importing Libraries**

**from** **sklearn.linear\_model** **import** LogisticRegression

**from** **sklearn.ensemble** **import** RandomForestClassifier

**from** **sklearn.tree** **import** DecisionTreeClassifier

**from** **sklearn.neighbors** **import** KNeighborsClassifier

**from** **sklearn.ensemble** **import** AdaBoostClassifier

**from** **sklearn.ensemble** **import** GradientBoostingClassifier

**from** **sklearn.naive\_bayes** **import** GaussianNB

**from** **sklearn.svm** **import** SVC

**import** **scikitplot** **as** **skplt**

Since the dataset is large to my system configurations, ensemble techniques will be efficient although I’m testing the results with the below algorithms.

1. Logistic Regression

2.Decision Tree Classifier

3. Random Forest Classifier

4. KNeighbors Classifer

5. AdaBoost Classifier

6. GradientBoosting Classifier

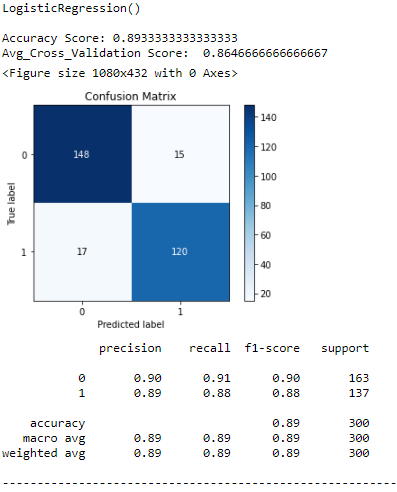
7.Gaussian NB

8.SVC

In order to test the model, I’m using accuracy score, AUC ROC score and F1 score, further in order to verify the model’s fit, I’m using cross validation score to identify the best model.

We will now use these libraries to build models

**Logistic Regression Model:**

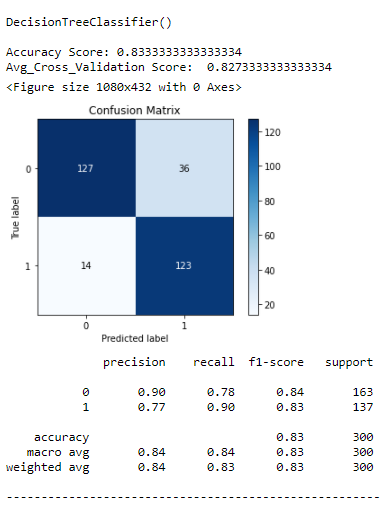


By  **Logistic Regression Model,** we were able to get the accuracy score of 0.89333.

Cross\_Validation\_Score: 0.86466

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting.

**DecisionTree Classifier Model:**

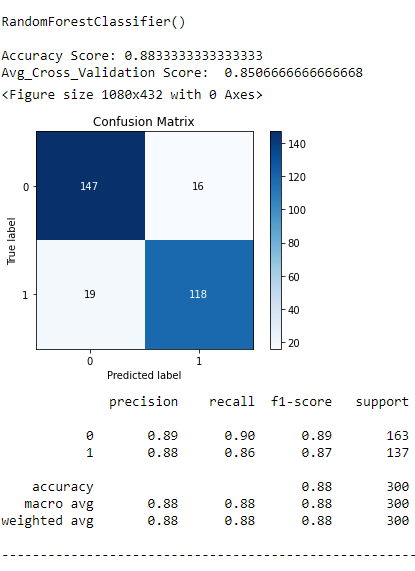


By **Decision Tree Classifier model,** we were able to get the accuracy score of 0.83333.

Cross\_Validation\_Score: 0.82733

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

**Random Forest Classifier Model:**

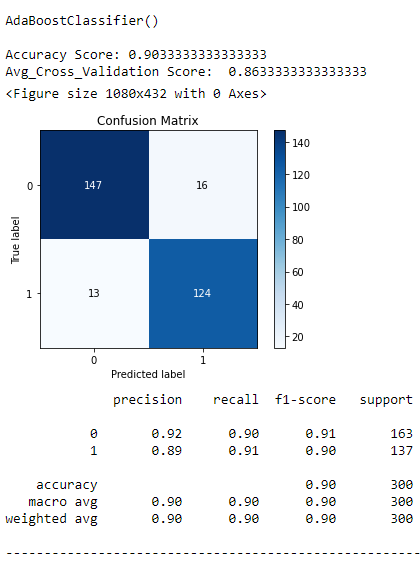


By **Random Forest Classifier model,** we were able to get the accuracy score of 0.883333.

Cross\_Validation\_Score: 0.850666

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting.

**AdaBoost Classifier Model:**

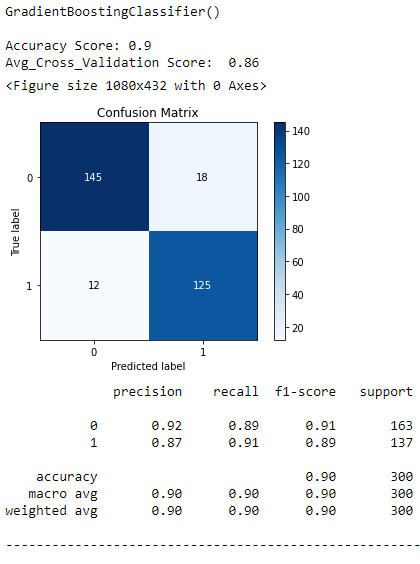


By **AdaBoost Classifier model,** we were able to get the accuracy score of 0.90333.

Cross\_Validation\_Score: 0.86333

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

**Gradient Boosting Classifier Model:**

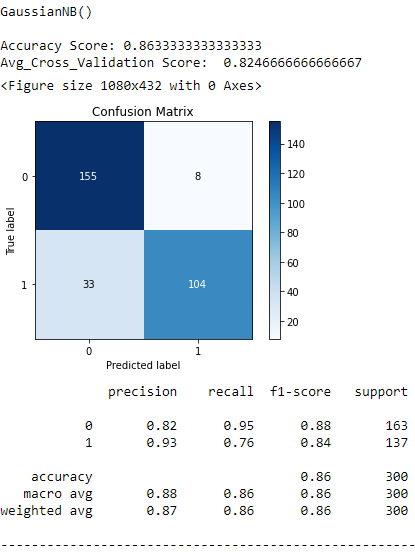


By **Gradient Boosting Classifier model,** we were able to get the accuracy score of 0.90.

Cross\_Validation\_Score: 0.86

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

**GaussianNB Model:**

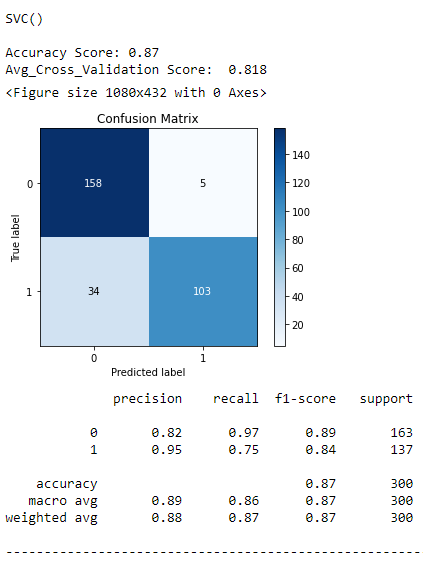


By **Gaussian NB model,** we were able to get the accuracy score of 0.86333.

Cross\_Validation\_Score: 0.82466

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting

**SVC Model:**

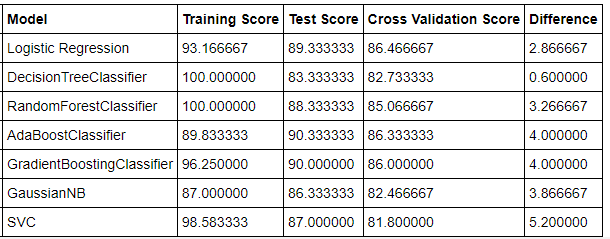


By **SVC model,** we were able to get the accuracy score of 0.87.

Cross\_Validation\_Score: 0.818

Further, I’m verifying the fit using cross\_val\_score with cross validation of 5 and see that the model is not overfitting.

**Finding the best model by subtracting the model’s accuracy with the cross validation scores and mentioning them under the “Difference” column.**



*#Let's check the row that has the least difference value in "Difference*

*Column"*



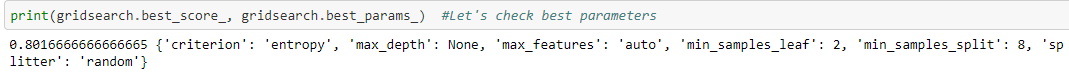
"DecisionTree Classifier" model comes up with "high accuracy score" and with least difference between the Accuracy Score and the Cross validation score

## **"DecisionTree Classifier" is our best model with 83.34 % Accuracy Score**

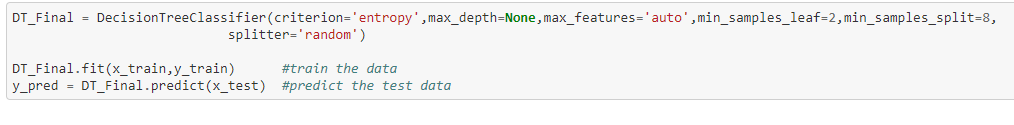
### **Hyper Tuning**

Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are: GridSearchCV. RandomizedSearchCV. GridSearchCV. In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values.

Let's Hyper tune our model to increase the accuracy score. And here are the best parameters that we have found

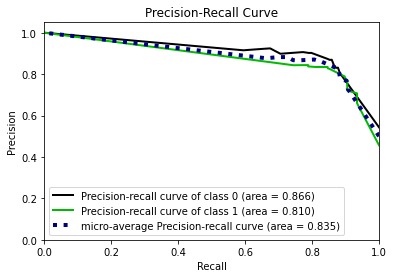


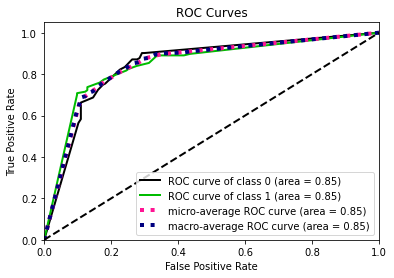
Fit these parameters and train the model

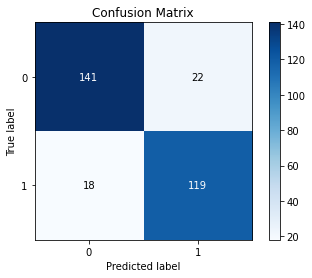


**Accuracy Score: 0.8666666666666667 (after hypertuning the model)**

Our model score increased by 3.3333 %







**Classification Report**

**precision recall f1-score support**

**0 0.89 0.87 0.88 163**

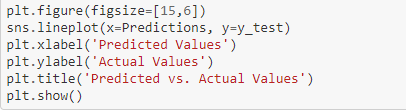
**1 0.84 0.87 0.86 137**

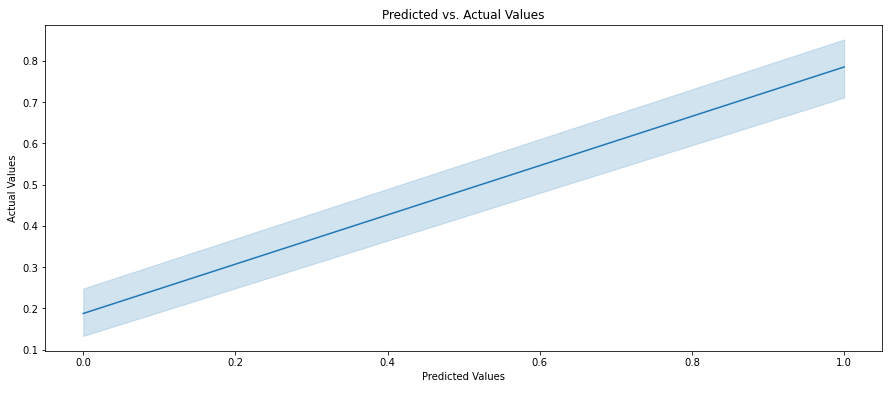
**accuracy 0.87 300**

**macro avg 0.87 0.87 0.87 300**

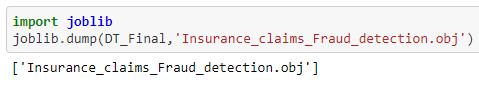
**weighted avg 0.87 0.87 0.87 300**

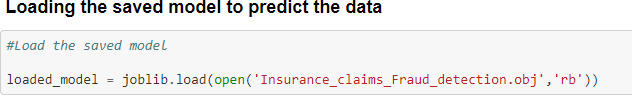
**Let’s plot the Actual and Predicted Values in a Lineplot**

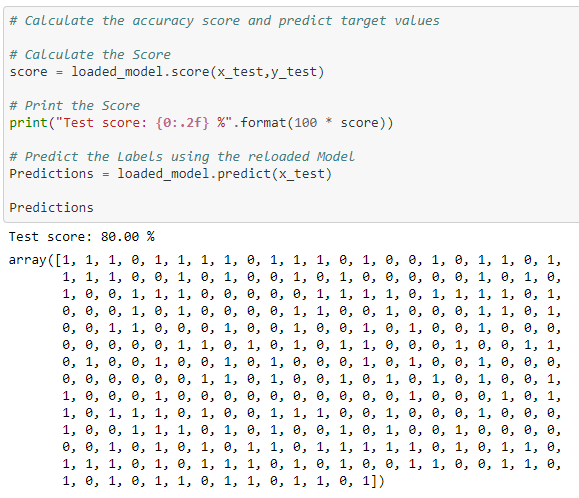




## **Saving Model for Future Predictions**







**Concluding Remarks**

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 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.

This model is built to detect the auto fraud claim in doing so number of frauds are less and it has reduced the loss for business and insurance companies since it showed the accuracy of authentic insurance claim.

**For the detailed coding please go through the below direct link which relates to the coding part of our model**

<https://github.com/PoonamRajput16/DataTrained_Evaluation_Projects/blob/main/Insurance_Claims_Fraud_Detections.ipynb>

## **Our model is now ready to predict the "Fraud Insurance Claim" with 86.67 % Accuracy**

