**BLOG**



**Insurance Claim-Fraud Detection Project**

Submitted by: Amit Matreja



**INTRO:**

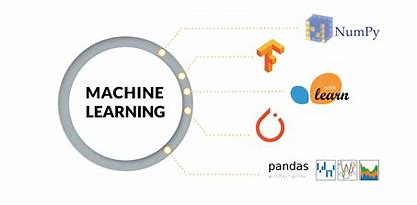
Insurance fraud is a major problem in the United States at the beginning of 21st century. Insurance fraud occurs when an insurance company, agent, adjuster or consumer commits a deliberate deception in order to obtain an illegitimate gain. So Insurance fraud has many categories among them Automobile insurance fraud is the major fraud type.

To overcome this problem we need a program which can help us predict whether someone is fraudulently taking insurance claim or is it a genuine Claim.For this, I am building a Machine learning model which can predict the claim is fraudulent or not.

I have procured the data regarding my Prediction model, The data contains very useful information regarding same, Data Contains details like **Insured information, insured persons personal details and the incident information totally we have 40 features in the dataset**.

So using all these previously known information, analyzing the data and implementing all the algorithms, I achieved a good model that has **90.37% accuracy**. So let’s understand what all the steps we did to reach this good accuracy. I used libraries mentioned below:

* Numpy
* Matplotlib
* Seaborn
* Datetime
* Pandas



Now let’s get into the problem and build a best possible model to predict whether insurance claim is fraudulent or not. **In this particular problem we will be dealing with 40 features and we have to be very particular while analyzing the problem.** Let’s start with understanding problem statement:

**1.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, we are provided a dataset which has the details of the insurance policy along with the customer details. It also has the details of the accident on the basis of which the claims have been made.

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

In this particular problem we need to observe and analyse the insured person details with incident details and try to analyse the samples to know whether this is fraud or genuine claim.

The steps which we are going to follow are **first we will import and analyze the dataset , then exploratory data analysis(EDA) , data visualization, data cleaning, pre-processing, model building, model saving and final prediction to check the performance of our model.**

**2.Analyzing the Dataset:**

Let’s first import the dataset. We have 40 features in the dataset. We have to observe and analyze the data. Although, as a Data Scientist we must procure lot of information regarding the subject which can help us build a model which can provide us best accuracy score.But, There can be some columns which may or may not affect our model prediction, we must observe those columns closely and if required, we must drop them as to . Now ,lets observe our target

column ‘fraud\_reported’ and we will also check the data type of target column to decide which kind of problem algorithms we have to implement.

After observing the dataset, we got to know that our target column “fraud\_reported” is a categorical dtype , so , This is a classification problem and we have to use all classification algorithms while building our model.



First I have imported the dataset which was in csv format. Below is how the dataset looks.



By looking into the features and observing the DataFrame I can say that we have both numerical and categorical columns with some unnecessary and missing information. So now we need to clean this data.

**Data Preparation and cleaning:**

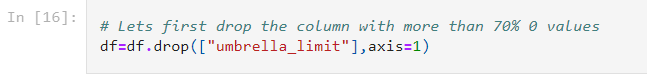
* First we need to do some analysis like checking shape, nunique, value counts, info etc…..
* After looking and observing the value counts if we find any unnecessary columns in the dataset we can drop those columns.
* Presently I found \_c39 column whose entries are all NaN Values.



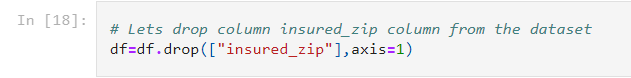
* After Analyzing the value counts of each column policy\_number and incident\_location has 1 element with 1000 value counts which means all the values are unique. These features will not help us in model building indeed will make it more complex and which will affect our model’s accuracy.



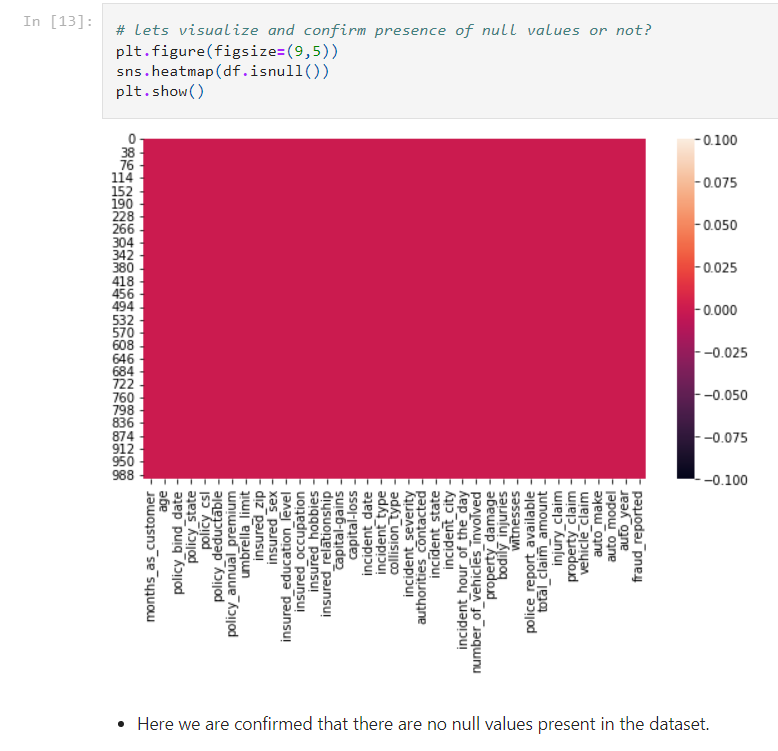
* By looking into the value counts of umbrella\_limit column I noticed there was 80% zero’s in this column so this column will create some skewness in data so it is better we drop this column.

.

* And I also noticed that insured zip is the zip-ID given to insurance person and this also will not help us in model building so I’m going to drop this column as well.



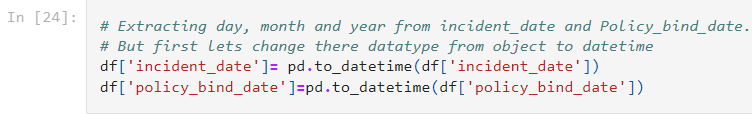
**Checking for Null values:**

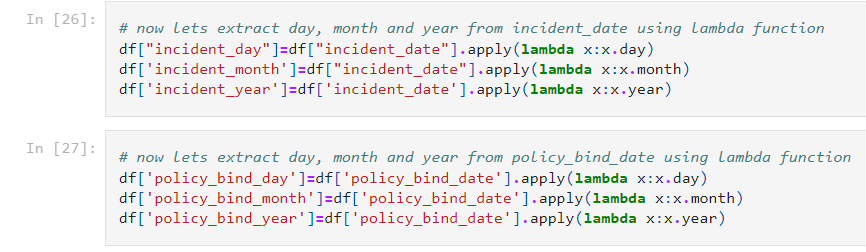


* Here we can see only one color red which denotes to 0.000 and no null values present in the dataset. So we can proceed further.

**Feature Extraction:**

* Let’s change the datatype of these two columns from object to datetime and then extract day, month and year from policy\_bind\_year and incident\_date.

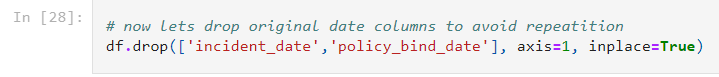




* I Studied value counts of extracted columns I found single unique entries in incident\_year column which means all the entries in this column are same which will not help us in prediction so I am Dropping it.



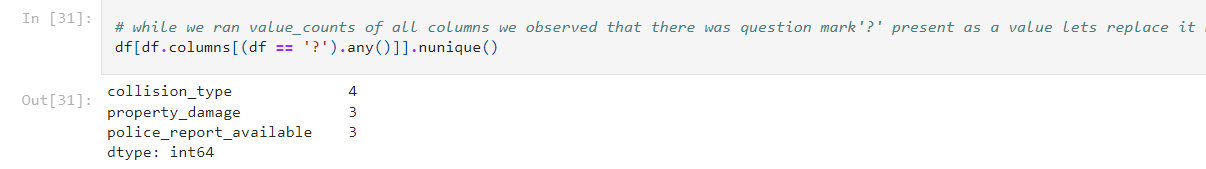
* After extracting all the necessary columns from the old columns we have to drop old columns. If we don’t drop those columns they will behave as duplicate columns and create multicolinearity issue.



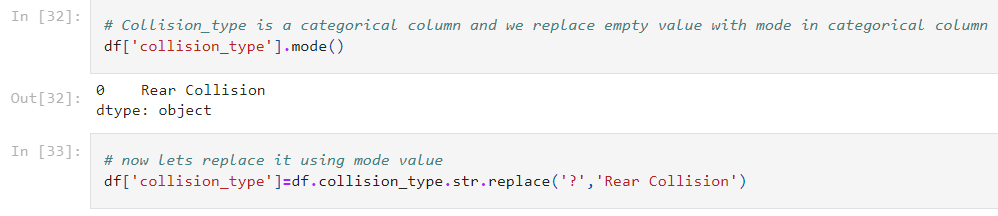
**Now it’s time to replace ‘ ? ’ :**

I have noticed some unusual entries in the dataset like ‘ ? ’. It may be because of Missing data which is replaced by “ ? ” or some techinal error we got some entries as ‘ ? ’. So now it’s time to replace those unusual entries.

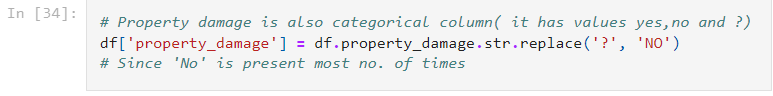
* Checking for ‘ ? ’ entries in all the columns. I found these entries in 3 columns.



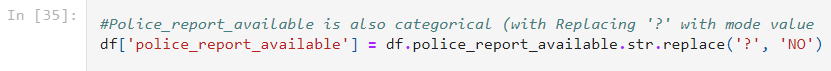
* Since collision\_type is a categorical type column so I have replaced the ‘ ? ’ values with it’s mode.



* And in property\_damage column NO has maximum count so I have replaced ‘?’ with NO.



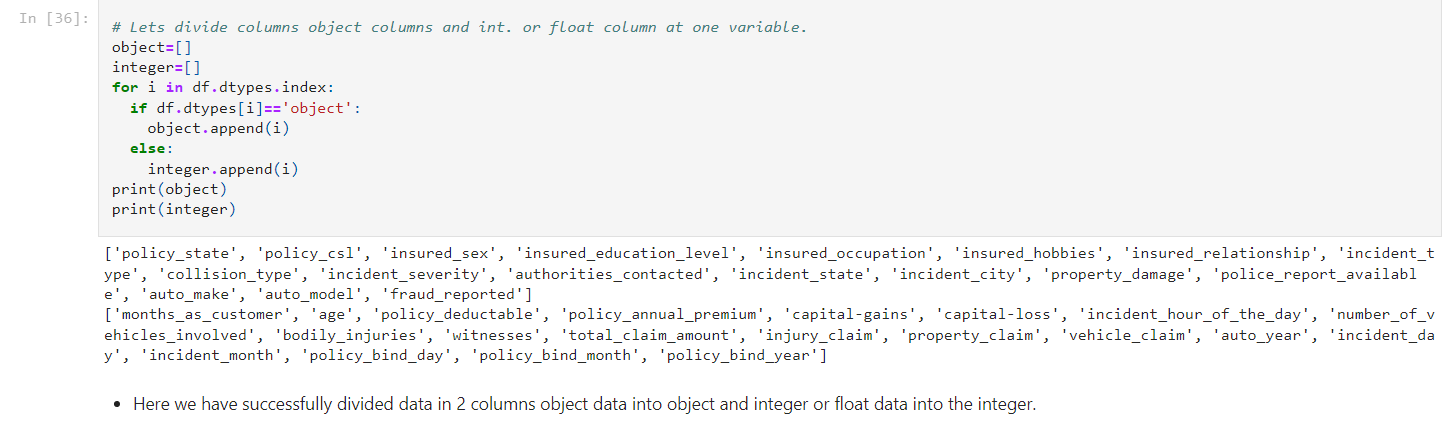
* In police\_report\_available column NO has maximum count so I have to replace ‘?’ with NO.



* Now all the feature extraction is complete and the data is set for analysis.

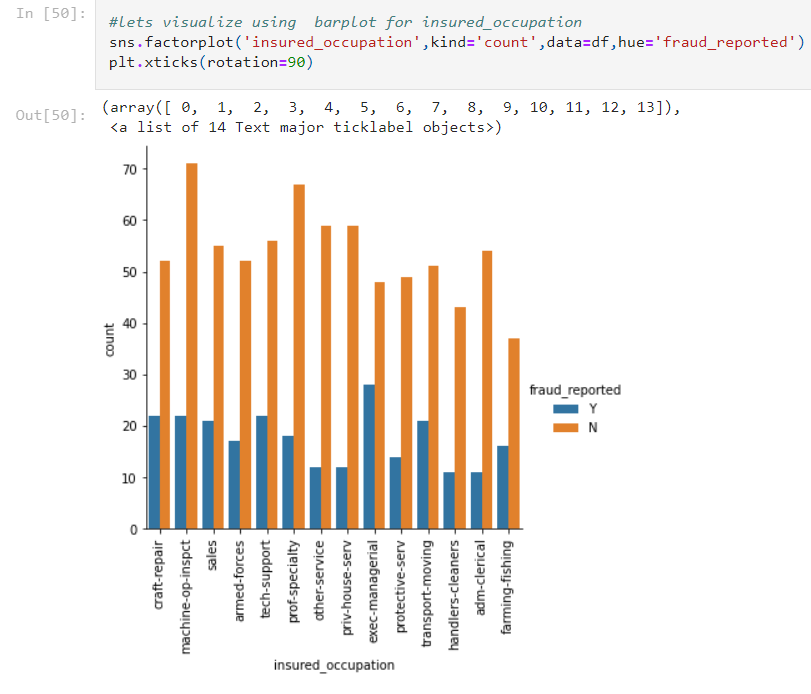
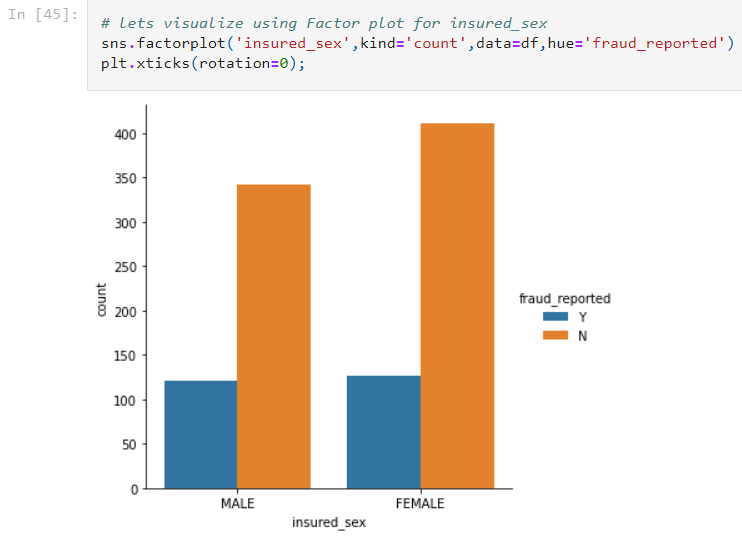
**Visualization:**

* As a first step we have to separate all numerical and categorical columns.

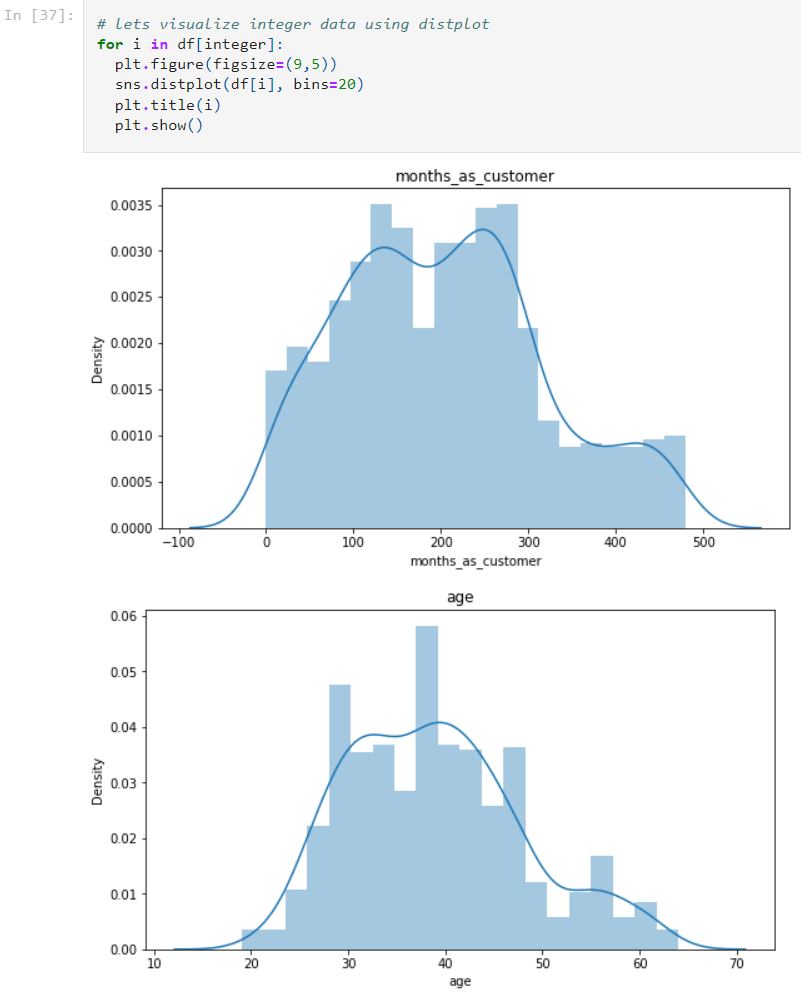


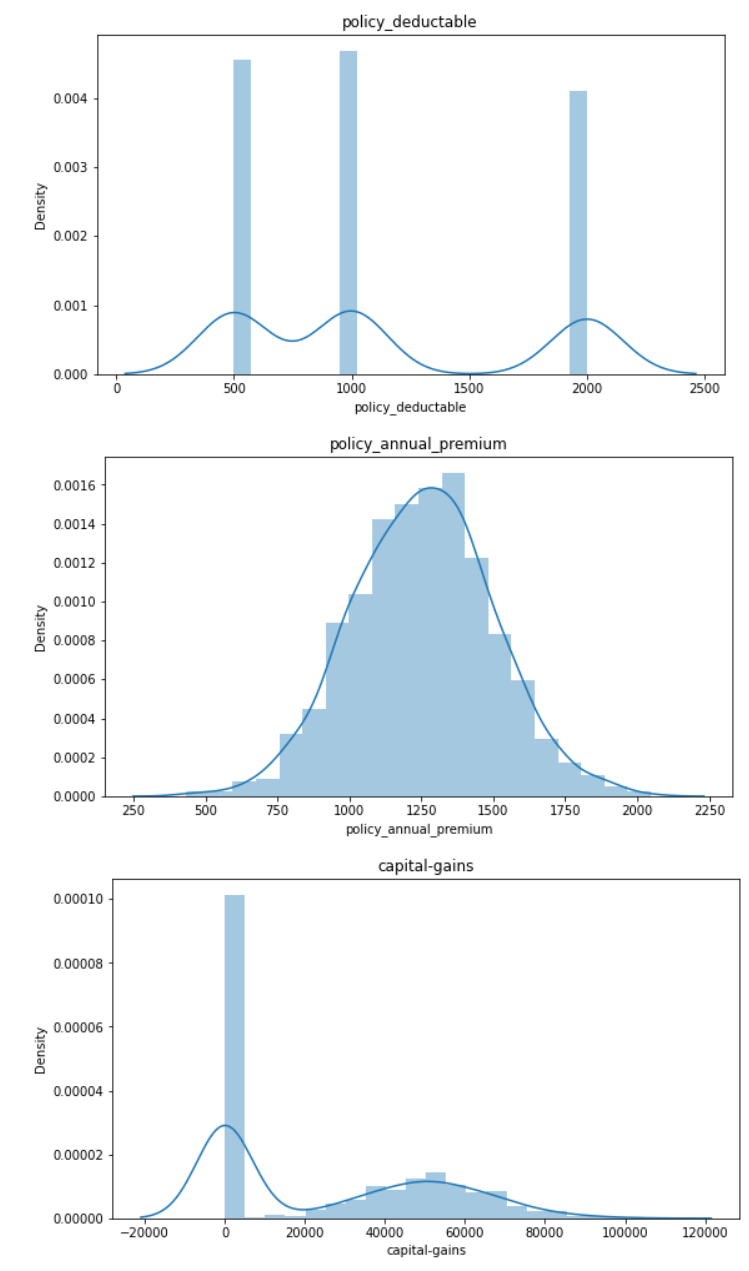


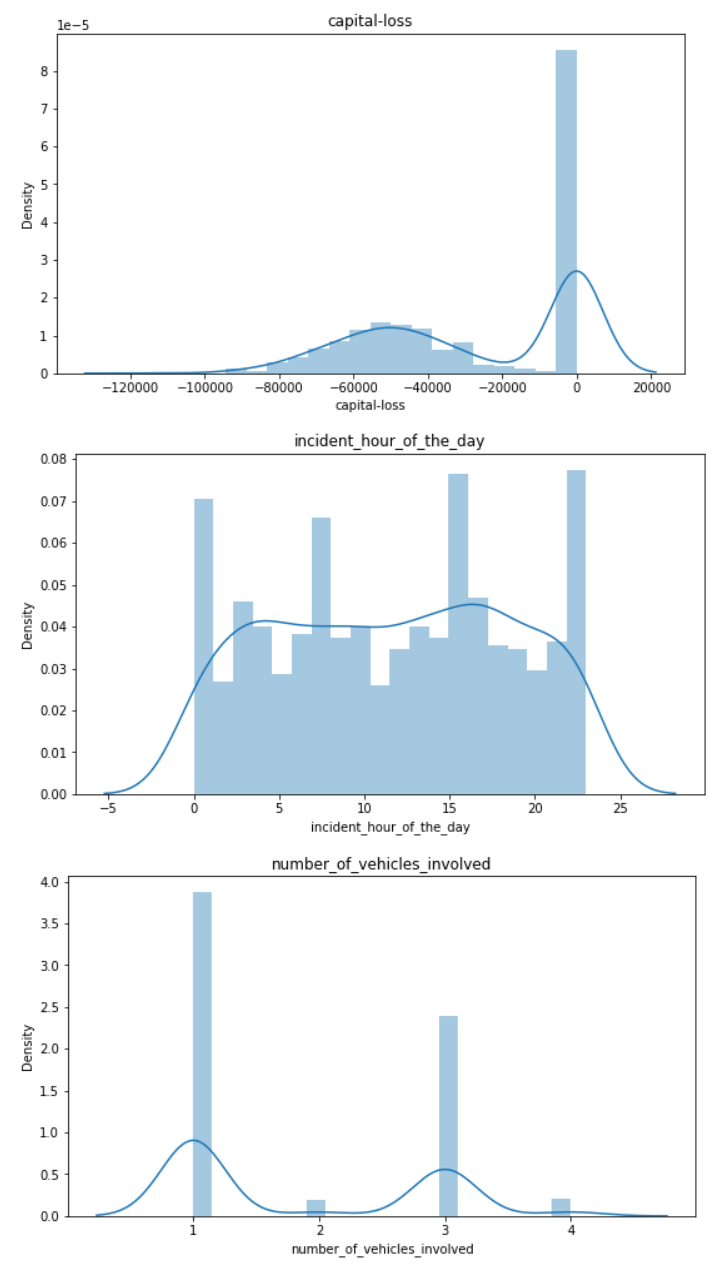
* Almost 1/3rd of the Insurance claims in state OH and IN were fraud\_reported, while in IL the fraud\_report percentage stands near 25%
* Below is the factor plot for Insured sex I noticed that in both the genders the count of fraud reported is same. But the fraud not reported is high with females. Which means females are more trust worthy than men.

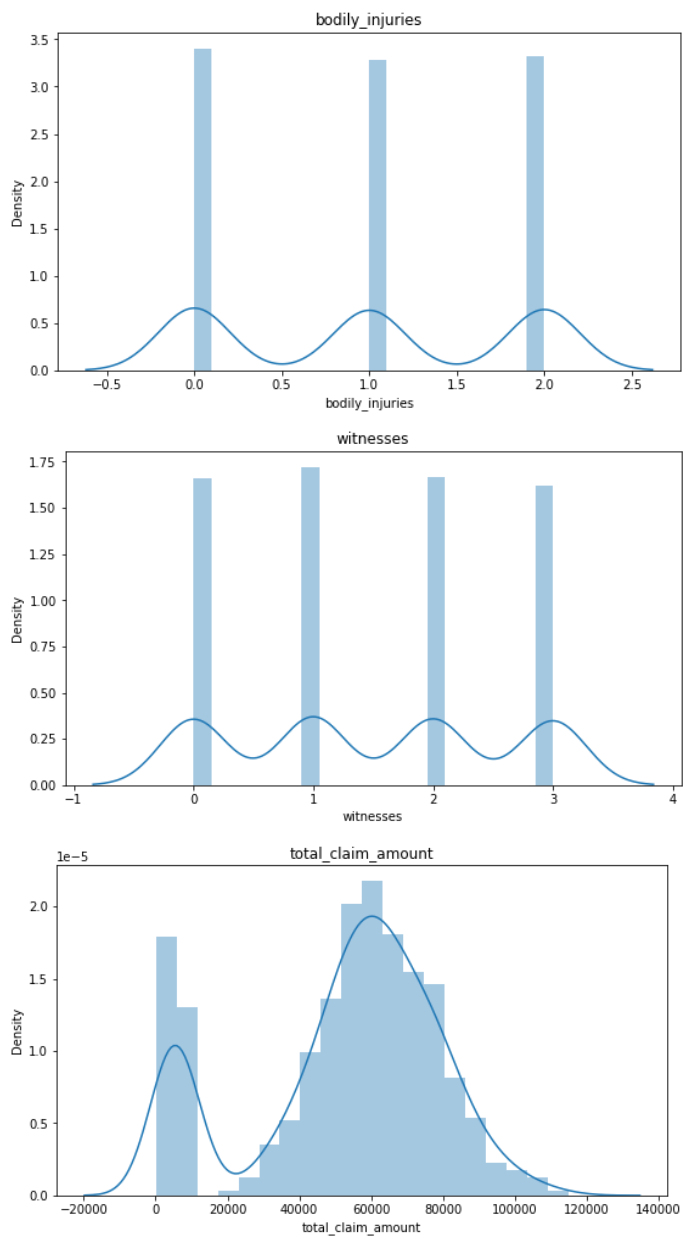


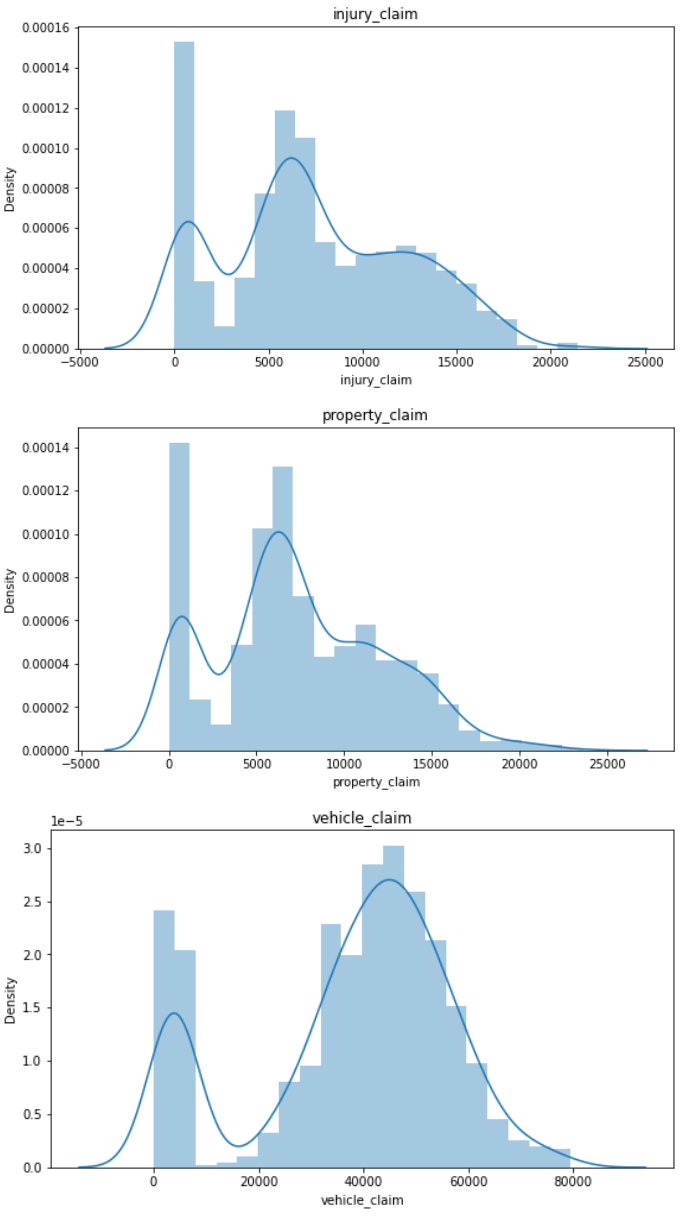
* Next plot is for insured occupation column. And the fraud count is more for exec-manegerial persons and fraud not reported count is high for machine-op-inspct. Which means good occupated persons looks most fraudulent.

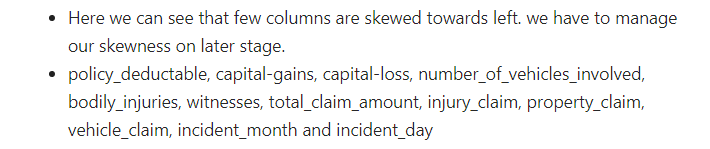


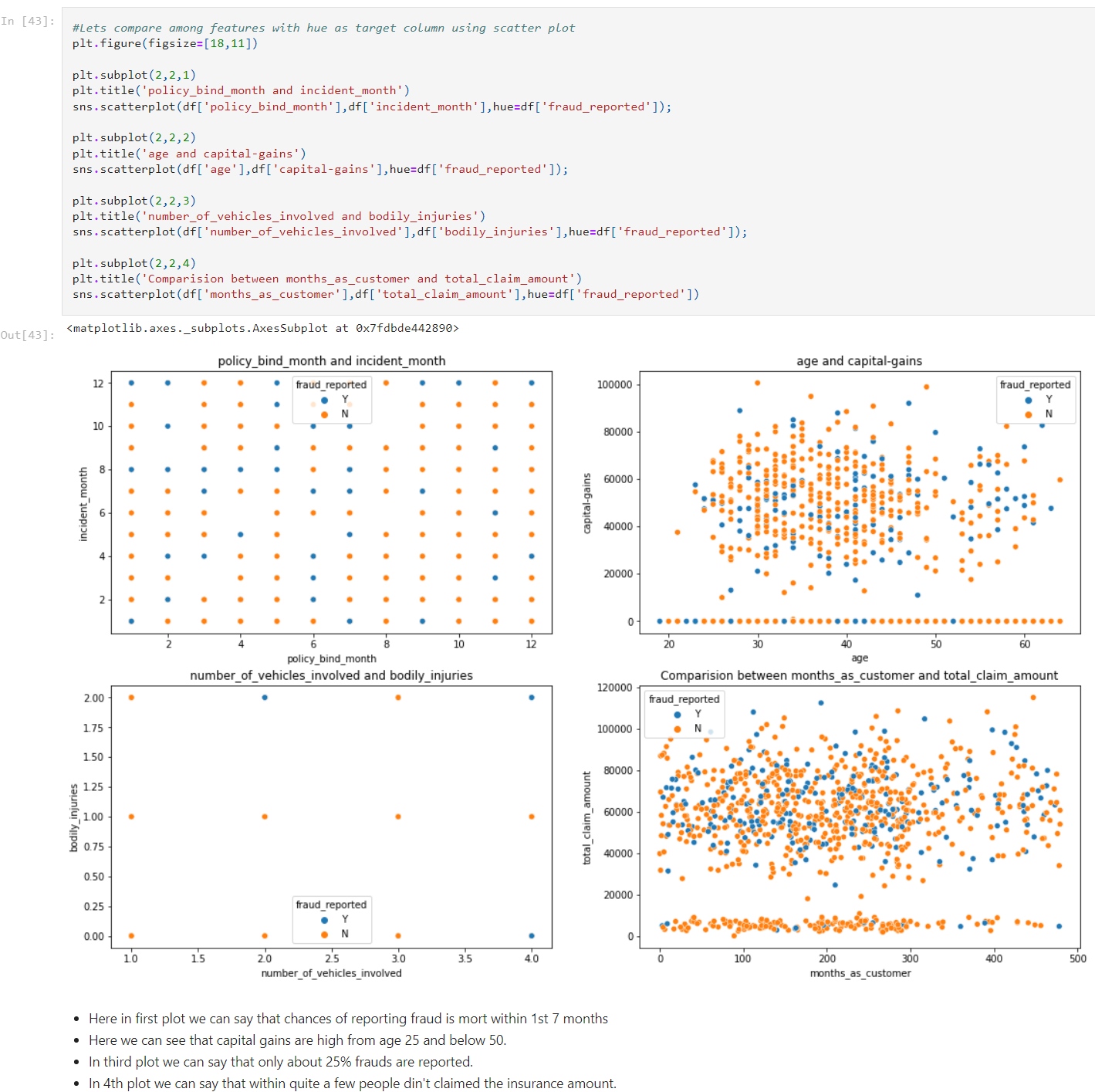






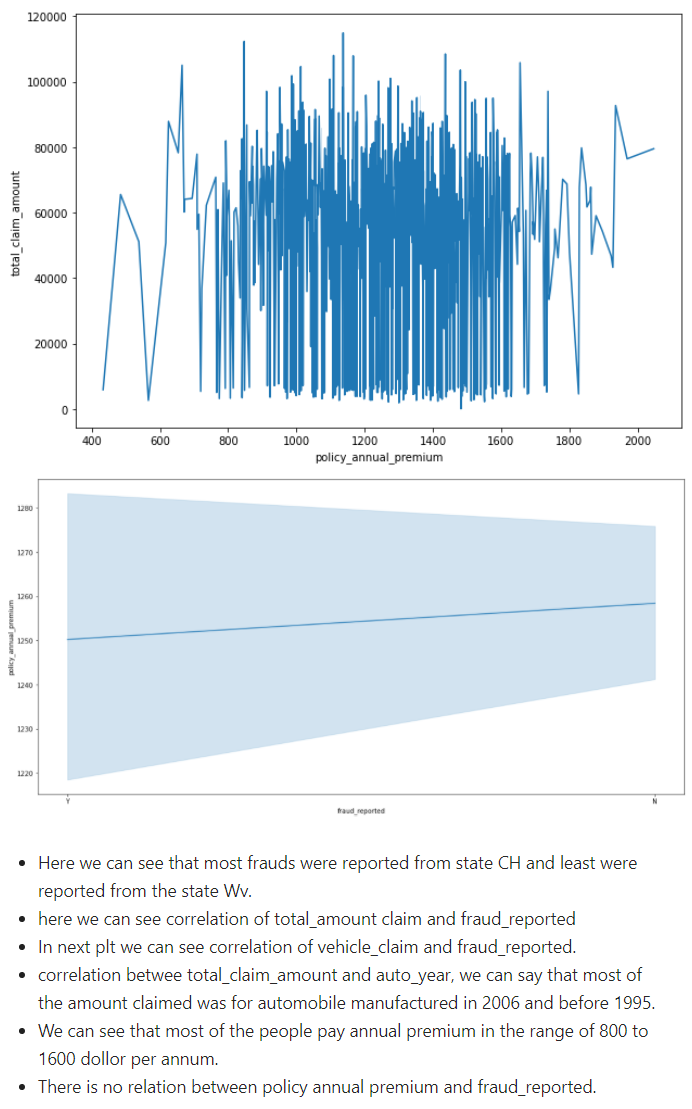








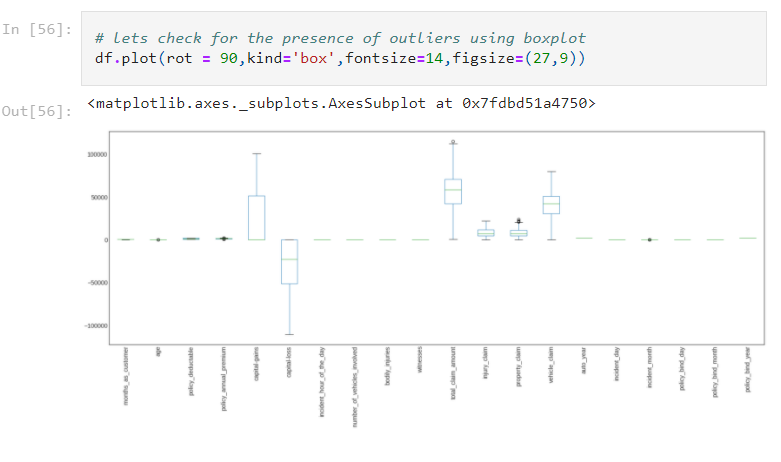




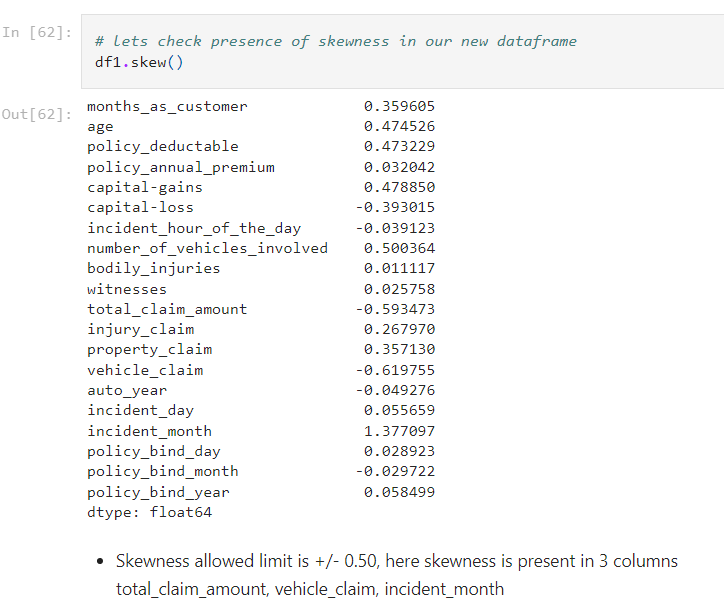
**3.EDA Conclusions:**

* I have checked for NaN values and I found there was no NaN /missing values in the dataset.
* I have extracted the necessary features from existing features to get better accuracy and dropped the old columns to avoid multicolinearity. If I keep the old columns as it is then they will act as duplicates in the model.
* I have also dropped the unnecessary columns. And also I replaced the ‘?’ entries with there suitable values.
* I have used both matplotlib and seaborn to visualize the data.
* To get better insight on the features I have used distplot, barplot, lineplot, scatterplot and boxplot since most of my columns were categorical I have used all categorical plots. For numerical columns I have used numerical ploting but I did not get any good pattern with numerical columns.

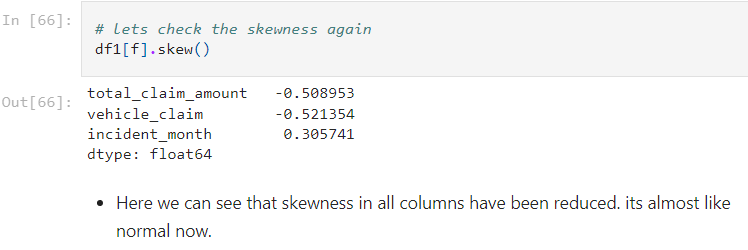
**Checking for Outliers and Skewness:**



* I have used box plot to check outliers. And I found outliers in age, policy\_annual\_premium, total\_claim\_amount, property\_claim and incident\_month. Now I have to remove outliers in these columns.
* To remove outliers I have chosen zscore with 0.4% dataloss. After removing the outliers using zscore I have saved the dataset as df1.
* I can notice skewness in total\_claim\_amount, vehicle\_claim, csl\_per\_accident and incident month.



* To remove skewness I have used Yeo-johson method.
* Now the skewness has reduced in almost all the columns. It looks good to proceed now.



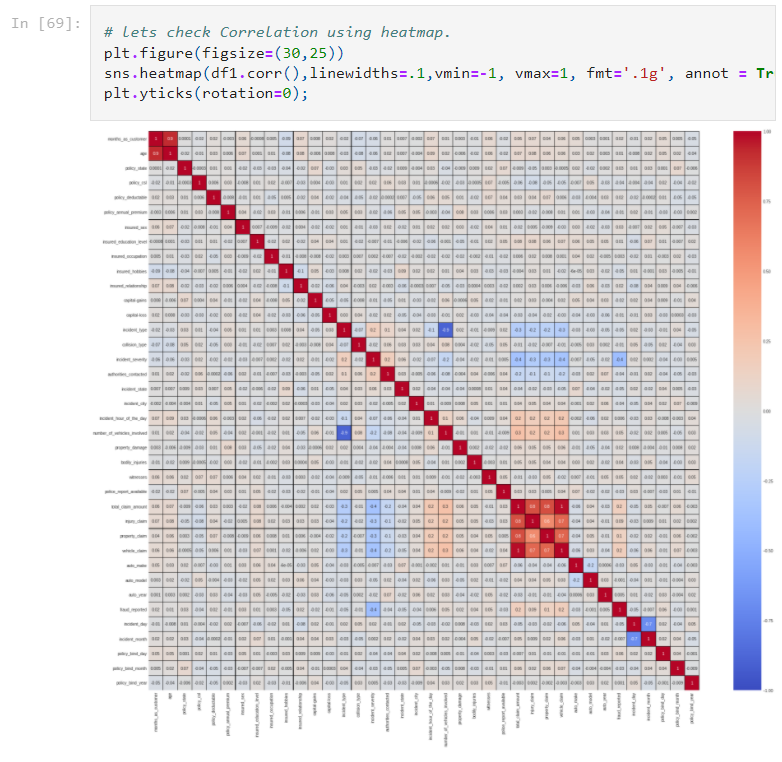
* Now it’s time to encode our categorical columns. For that we have to separate categorical columns from cleaned df1.



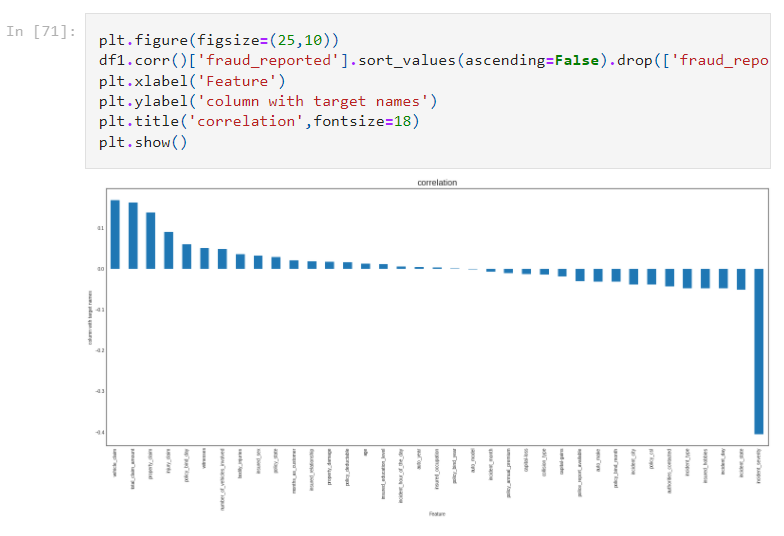
* For all the categorical columns in cleaned dataset df1 I have applied label encoding.

**Checking for correlation using heat map:**

After checking the correlation, to get better insight on the corr values I have plotted heat map. And this **correlation** has to be checked for **cleaned dataset**.



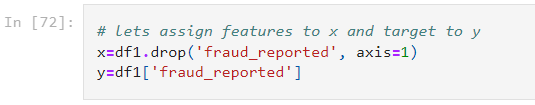
* Looking into the heat map I can say that there is multicolinearity issue and to get better insight on targets correlation with other features I have ploted bar graph.



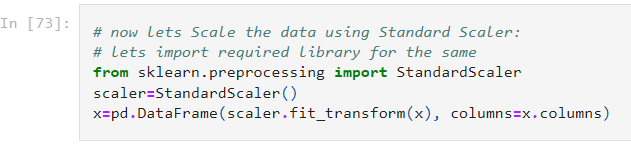
* Policy\_bind\_year, auto\_model, insured\_occupation, auto\_age and incident\_hour\_of\_the\_day are very less correlation with target but let me keep the columns and build the model. Since I don’t want to loose any data so first keeping all the columns let me build the model. After looking into the accuracy if I feel I can increase the accuracy by deleting these less correlated columns then again let me come back and delete these columns.
* Now my dataset is ready for preprocessing.

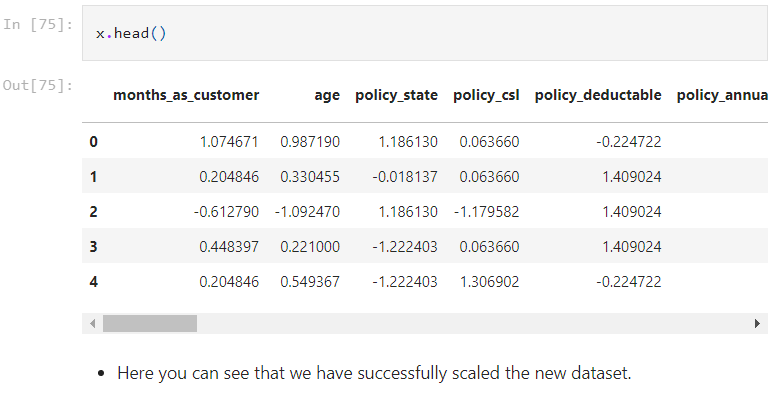
**4.Preprocessing:**

* As a first step I have to separate the dependent and independent features.



* I have taken x as features and y as dependent/target feature.
* Then I have to scale my features to get the same range in all the columns. If I don’t scale my independent columns then there is a chance that my model may get biased. So In this particular case I have used Standard scaling as I have removed all outliers and skewness from the dataset it is good to use standard scaler.





* Now scaling part is done.
* But I have left out with multicolinearity.
* I have to check VIF(variance inflation factor) now.

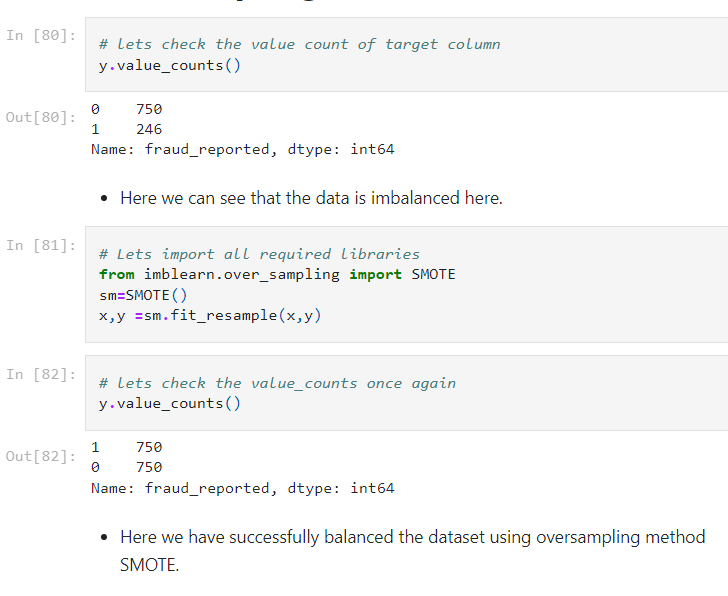


* I can notice a high VIF for total\_claim\_amount, so I have dropped this column first. After that again I checked for VIF and got the highest value for csl\_per\_accident so I dropped this column. Then my multicolinearity issue was solved.



**Data Balancing:**

* Since as observed before my target is imbalanced so now I have to balance it using over sampling. I can also use under sampling but I haven’t because of dataloss.



* Now the balance issue is solved and my target is beautifully balanced

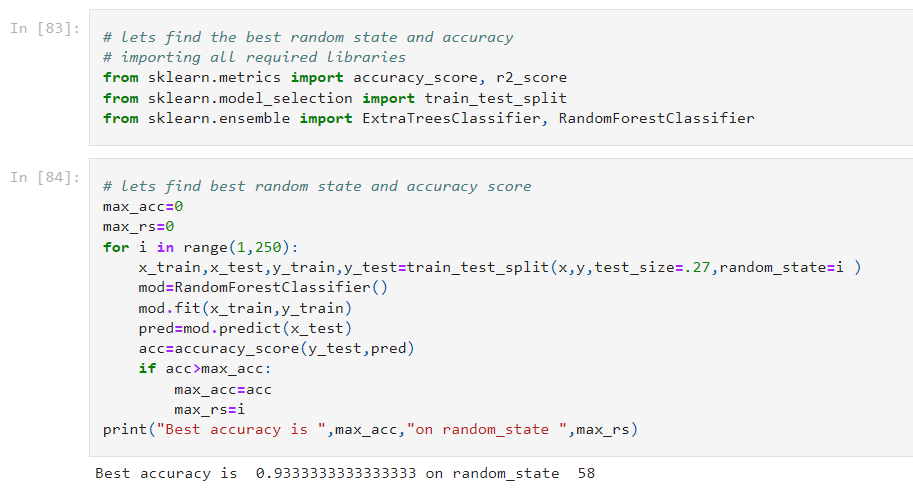
.

* My data is all set for model building. Let’s go ahead with classification algorithms since this is a Classification Problem

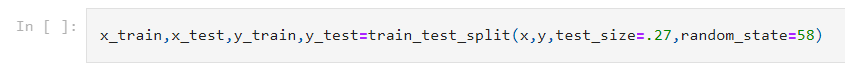
**5.Building ML Models:**

1. **Finding best random state and accuracy:**

Let’s find the best Random state and accuracy first.



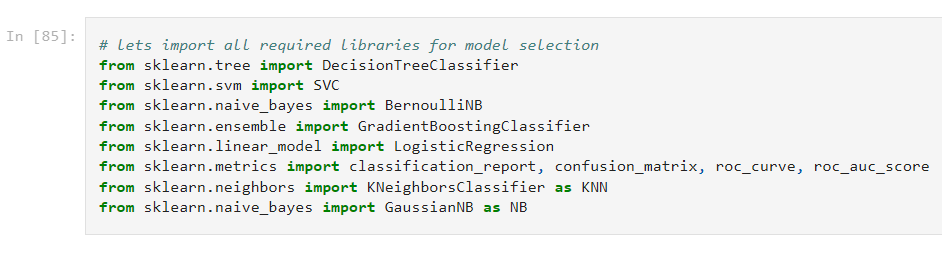
* I got best random state = 58 and accuracy = 93.33%. Now the task is to find the best fitting model.



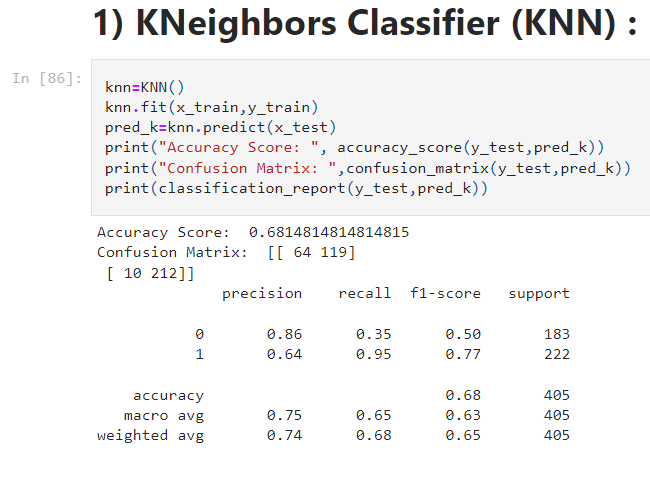
* Created train and test data as x\_train, x\_test and y\_train, y\_test.

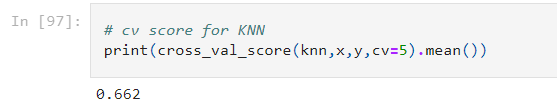
1. **Classification Algorithms:**

As a first step we have to import all the necessary libraries.

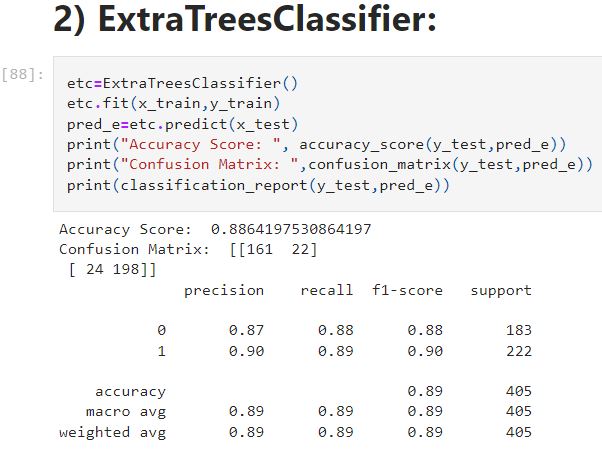


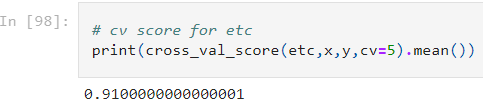
* I have used Cross validation as model evaluation metrics for all the algorithms. And I have used accuracy\_score, Confusion metrics as metrics in model building.



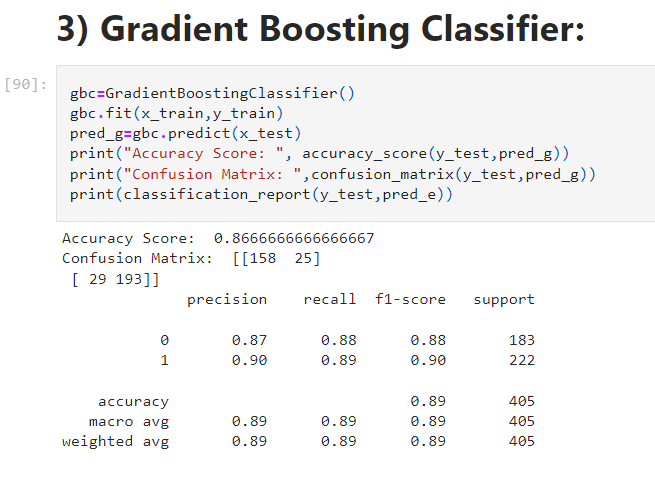


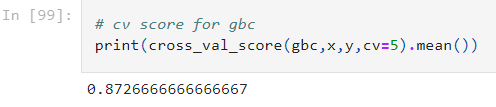
* KNeighbors Classifier model is giving me 68.15% accuracy\_score and the cross validation is 66.2%.



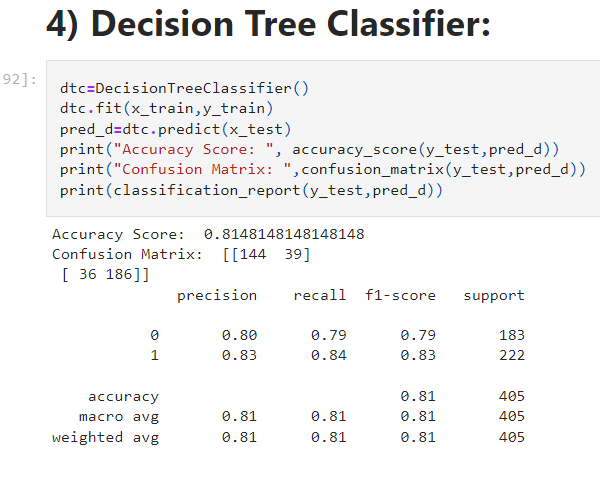


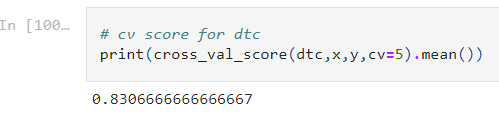
* Extra Trees Classifier model is giving me 88.64% accuracy\_score and the cross validation is 91%.Extra Trees classifier is give very good score, but first lets check scores of other models too.



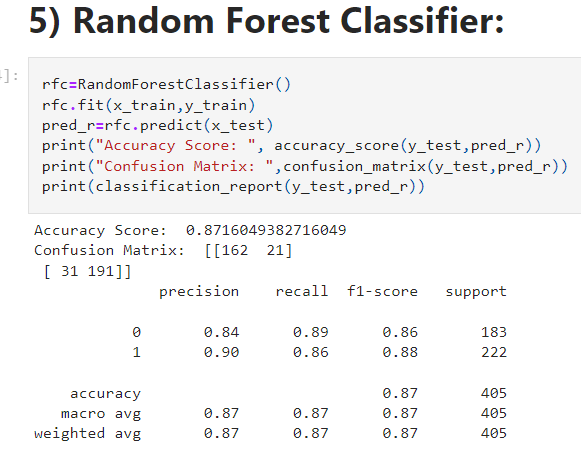


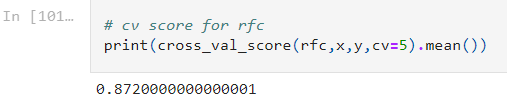
* Gradient Boosting Classifier model is giving me 86.67% accuracy\_score and the cross validation is 87.26%. This is also good score but not better than Extra Trees Classifier.





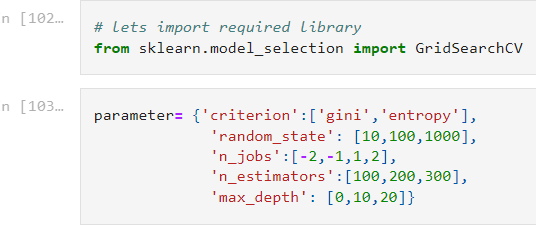
* Decision Tree Classifier model is giving me 81.48% accuracy\_score and the cross validation is 83.06%.



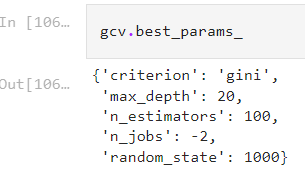


* Random Forest Classifier model is giving me 87.16% accuracy\_score and the cross validation is 87.2%. This is also a good score but not better than Extra Tress Classifier.

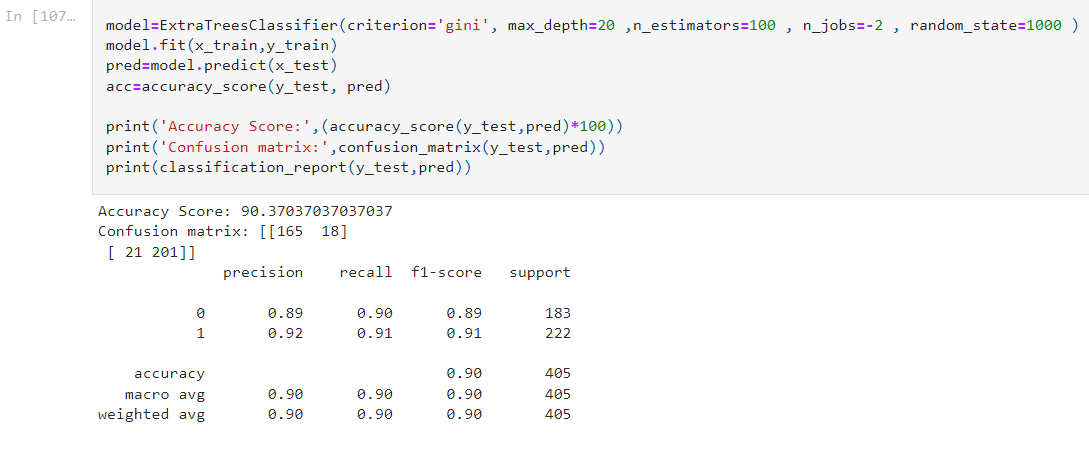
**Hyper Parameter Tuning:**



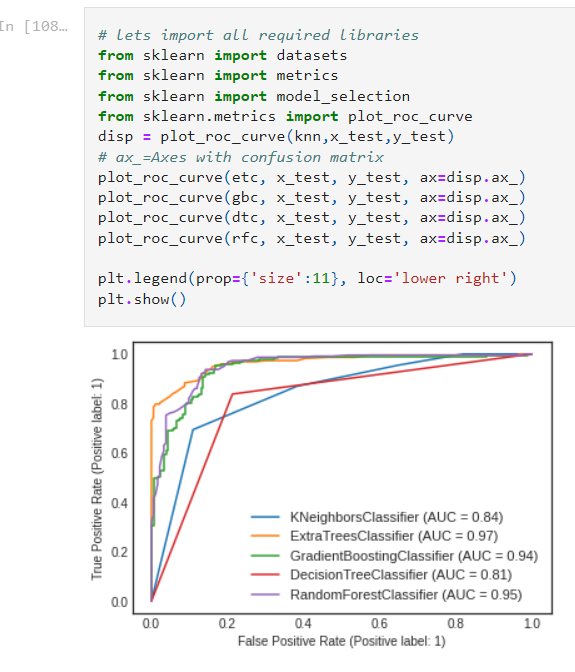
* Using the above parameters list I’m tuning my best model i.e., Extra Trees Classifier. And I have to choose the best parameters in above parameter list, with those parameters I have to build the best model.



* After knowing the above best parameters I gave the inputs for improving model accuracy.My models accuracy improved by almost 2%.

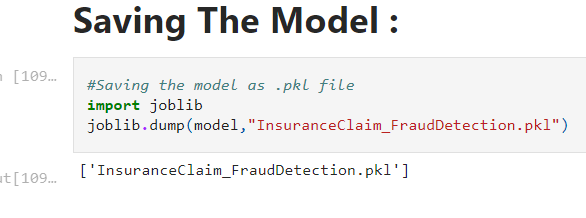


* After tunning the model accuracy is improved by almost 2 % which means the default parameters used by the model were giving the best accuracy. And the model is now ready with 90.37% accuracy which looks good!!!!.
* The ROC-AUC curve for all the above model is as shown below.



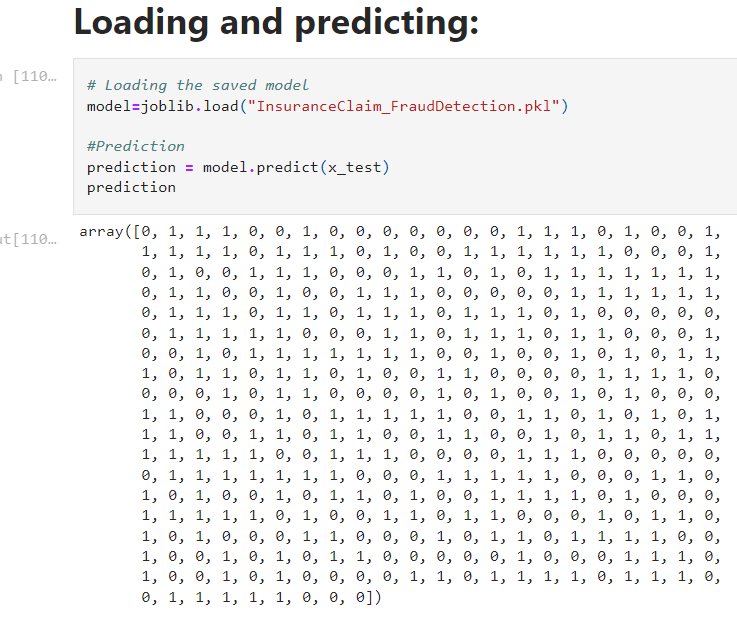
**The AUC value is high for Extra trees and also I found the least difference of model accuracy and cross validation score for Extra Tree Classifier. So I’m choosing Extra Tree Classifier as the best model with difference of 2%. And the model accuracy is 90.37% which is good.**

After getting this best model I have saved it using .pkl. As InsuranceClaimFraud.

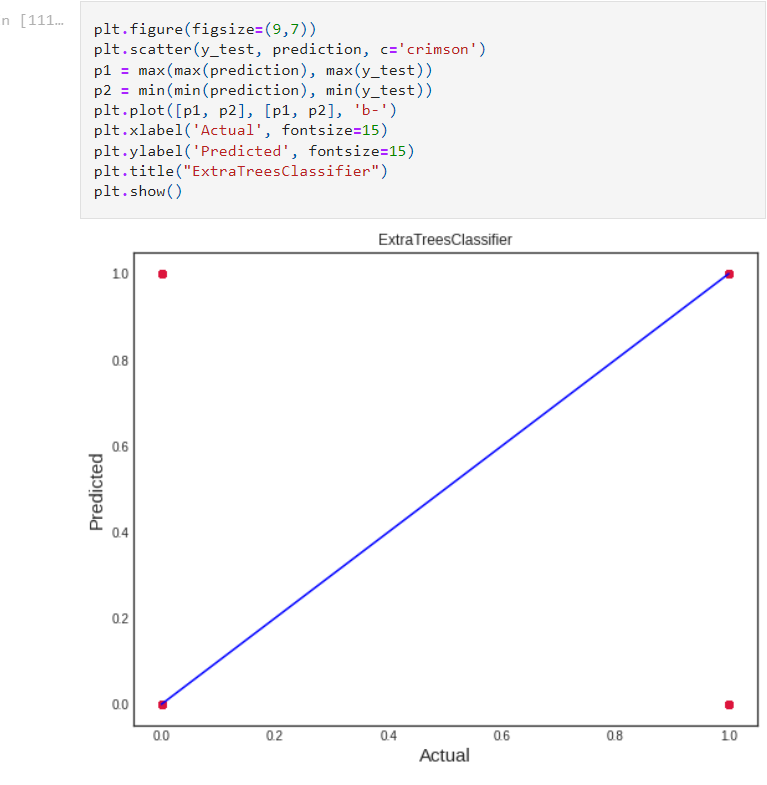


**Predictions:**

* Now using the saved model I can predict whether the insurance claim is fraudulent or not.



* After saving the best model, lets load the saved model and check for the actual values verses predicted values.



* Blue line is the actual values and red dots are predicted values and it feels really good to see my model is working good!!!.

**6.Conclusion:**

* This particular problem needs a good understanding of data, and in this problem, Feature Engineering is the one of the most important step which actually helped in our model performance.
* Here you can find the way we have handled numerical and categorical data and also how we build different machine learning Algorithms on the same dataset.
* Using hyperparameter tuning we Tried to improve the our best model’s accuracy, which to an extent we achieved as after hyperparameter tuning our final model showed improvement of almost 2 %.

With the help of this machine Learning Model anybody can predict and decide whether the insurance claim is fraudulent or not.

* 