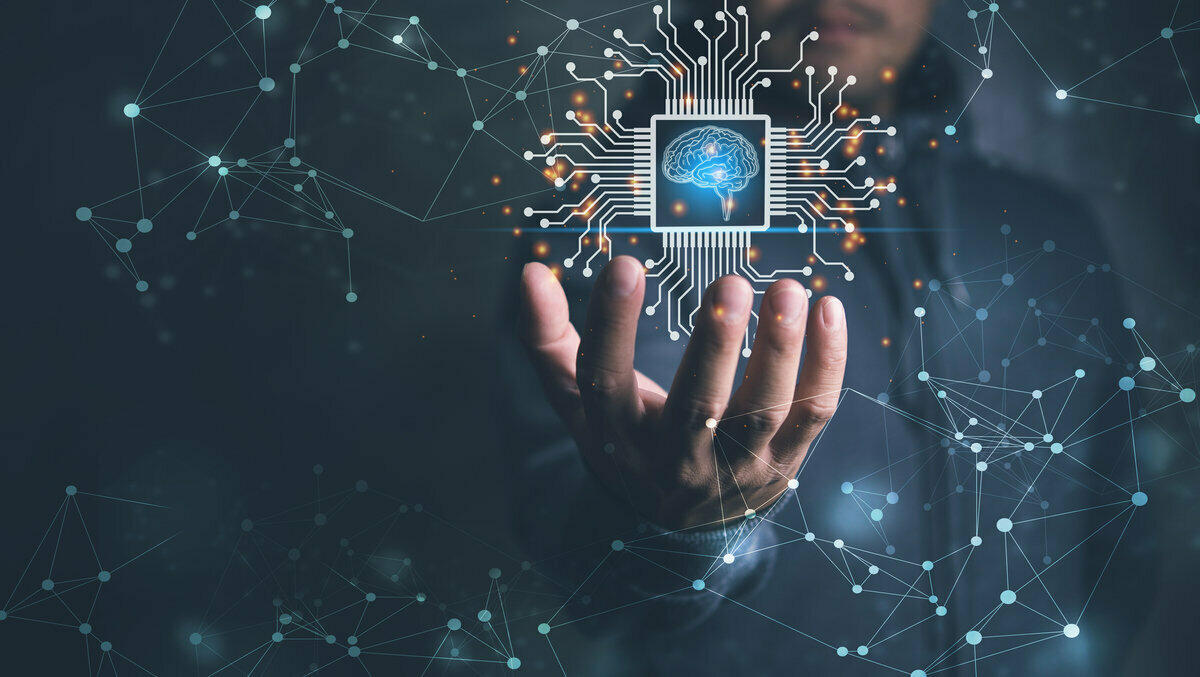
**DATATRAINED ACADEMY**

**BLOG ARTICLE-2**

**(I’m not saying it is going to be very easy, I’m saying it is going to be worth it.)**

**Insurance Claim Fraud Detection Project**

Submitted by: POOJA C

Batch number: 1831



**INTRODUCTION:**

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.

But now the question is how to predict wheather the insurance climbed is fraudulent or not. For that I have built a Machine learning model which can predict the claim is fraudulent or not. Using various features 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 and analysing the data I have achieved a good model that has **92% accuracy**. So let’s understand what all the steps we did to reach this good accuracy.



**Libraries used:**

* Python
* Numpy
* Matplotlib
* Seaborn
* Datetime

Now let’s get into the problem and build a best possible model to predict insurance claim is fraudulent or not. **In this perticular problem we are going to deal with 40 features and we have to be careful while analysing the problem.** Let’s have a look.

**1.Problem Definition:**

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 perticular problem we have to look into the insured person details and incident details and analyse the samples to know wheather the claim is fraudulent or not.

**Let’s do it step by step firstly analysing the dataset and doing exploratory data analysis, data visualization, data cleaning, pre-processing, model building, model saving and finally predictions.**

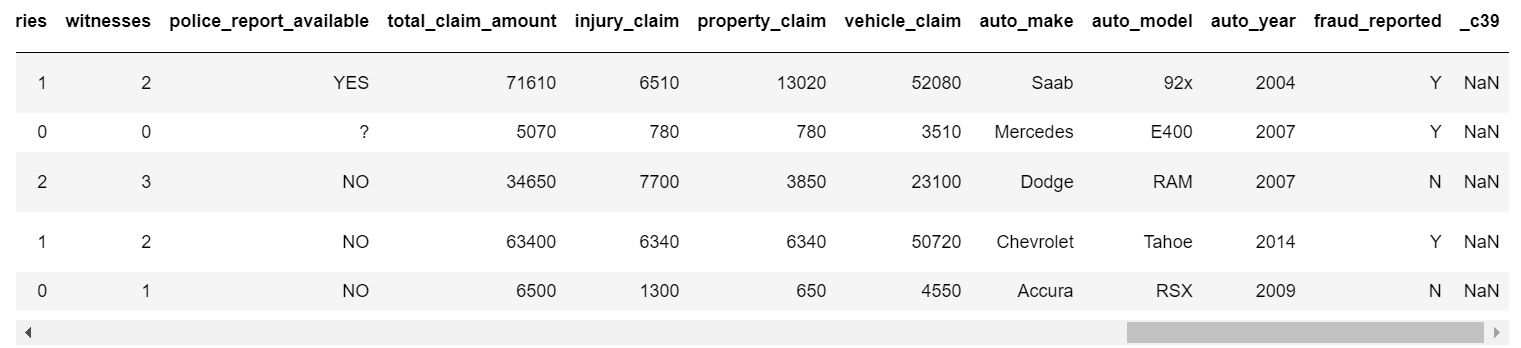
**2.Data Analysis:**

**** Let’s import the dataset first. I have about 40 features in the dataset. And I can use PCA to reduce the columns but I haven’t done that. Because being a data analyst our first duty is to avoid data loss. Since I have just 40 countable columns I can keep them and proceed with my steps. Now looking into the target ‘fraud\_reported’ and I have to make sure the data type of target column to decide the type of problem.

Since fraud\_reported is my target and it is a categorical column with string entries. So it looks quite clear that this perticular problem is a **Classification problem** and I have to use all classification algorithms while building the model.



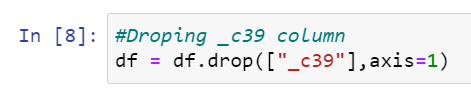
Firstly I have imported the dataset which was in csv format as df. Below is how the dataset looks.

****

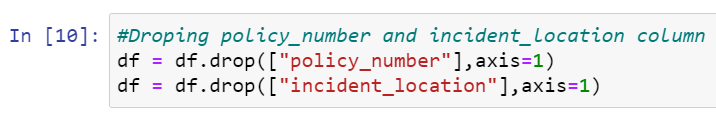
Above is my dataset. By looking into the features I can say that I have both numerical and categorical columns with some unnecessary entries. So now we have to clean this data.

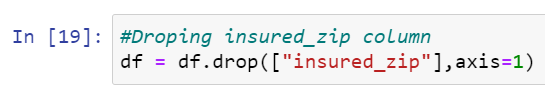
**Data Preparation and cleaning:**

* Firstly we have to do some statistical analysis like checking shape, nunique, value counts, info etc…..
* After reading 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. Keeping all entries NaN is useless so let me drop that column.

****

* After seeing 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 so I have dropped them.
* 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.

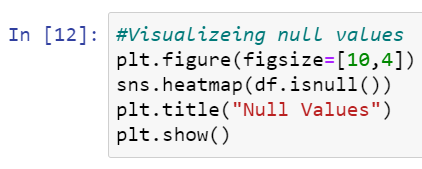
****

****

* 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 better let me drop this.

****

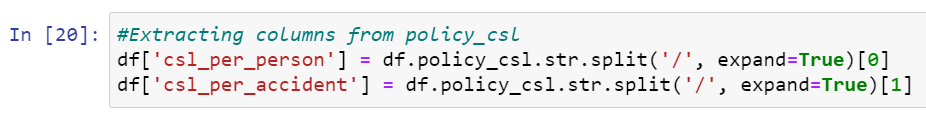
**Checking for Null values:**

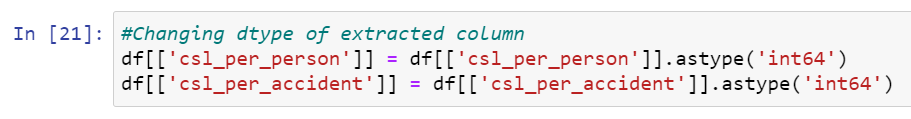
****

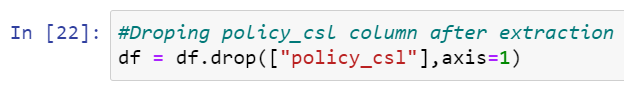
****

* I can observe there is no null values in any of the columns of dataset. So no worries let’s proceed.

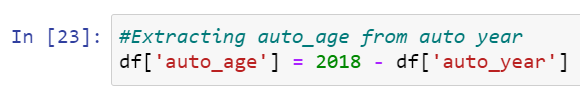
**Feature Extraction:**

****

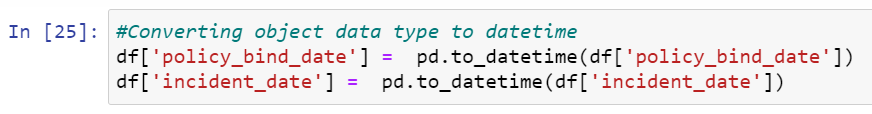
****

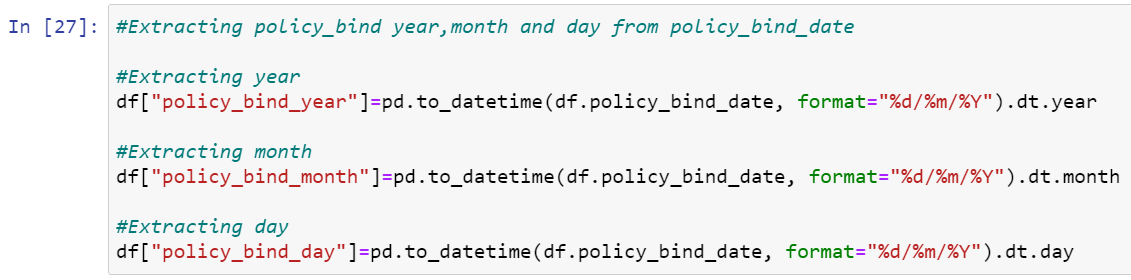
****

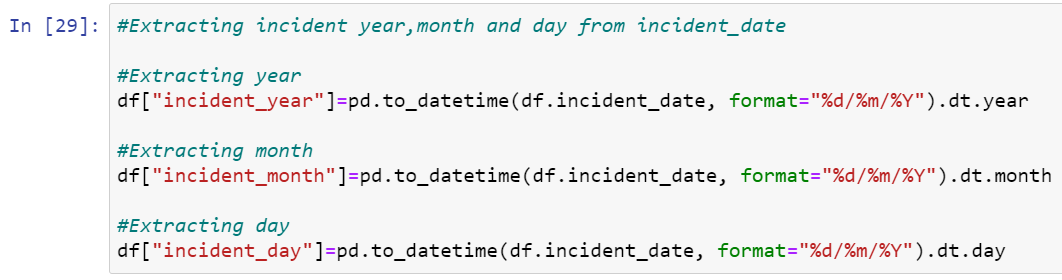
* I have extracted two columns csl\_per\_person and csl\_per\_accident from policy\_csl. Before policy\_csl was object type data since it was in the format (A/B), now I have extracted A and B separately (Where A and B are integers). And after extraction I have changed the data type from object to integer. And after extraction we have dropped policy\_csl column to avoid multicolinearity.

****

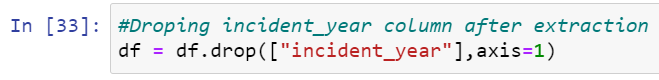
* I have also extracted auto\_age from auto\_year by taking difference of insurance climbed year to auto year. I felt auto age may help us more in prediction than auto year.

****

****

****

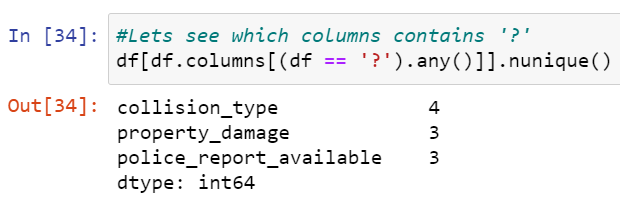
* Now it’s time to extract day, month and year from policy\_bind\_year and incident\_date. Before that we have to change the datatype of these two columns from object to datetime.

****

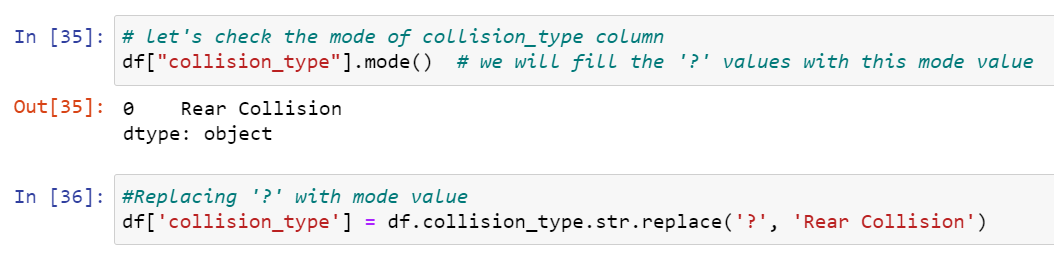
* I just had a look into value count of extracted columns I found single unique entries in incident\_year column which means all the entrices in this column are same so I don’t want to keep this unnecessary column in the dataset, so I have droped it.
* After extracting all the necessary columns from 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 ‘ ? ’ :**

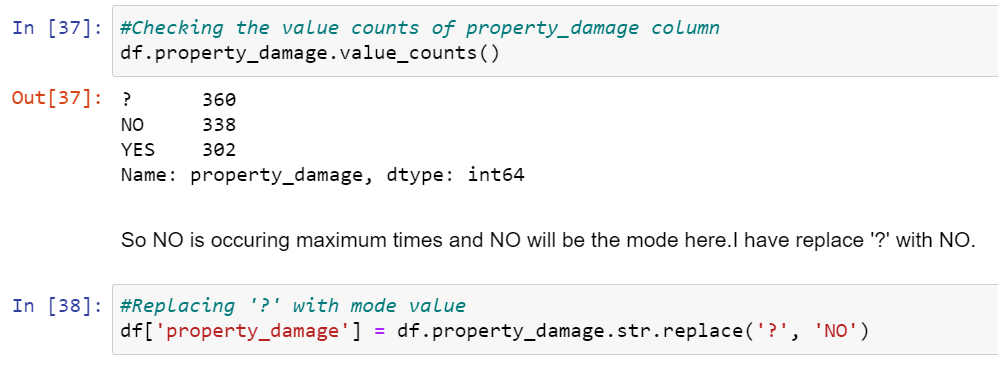
I have noticed some unnecessary entries in the dataset like ‘ ? ’. It may be because of some tying errors or some techinal error we got some entries as ‘?’. So now it’s time to replace those unwanted entries.



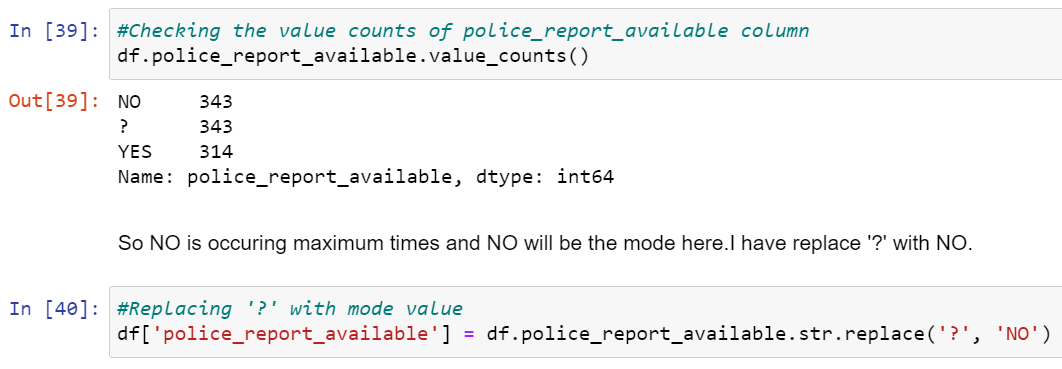
* Checking for ‘ ? ’ entries in all the columns. I found these entries in 3 columns.



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

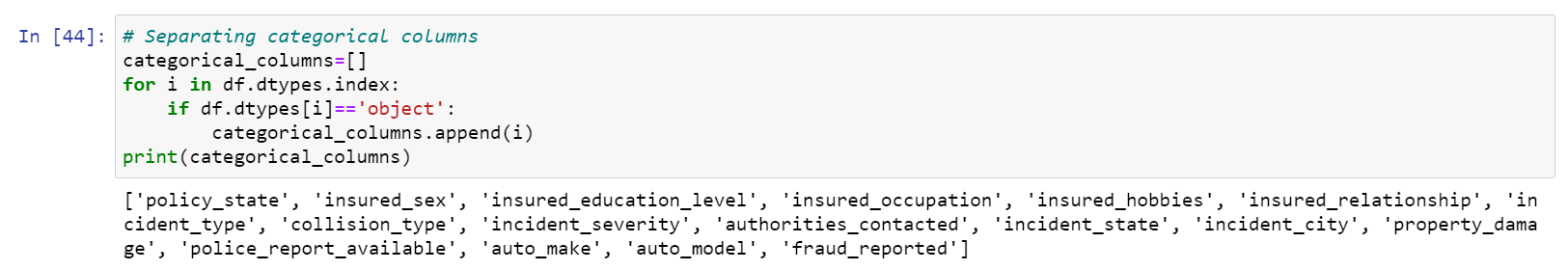


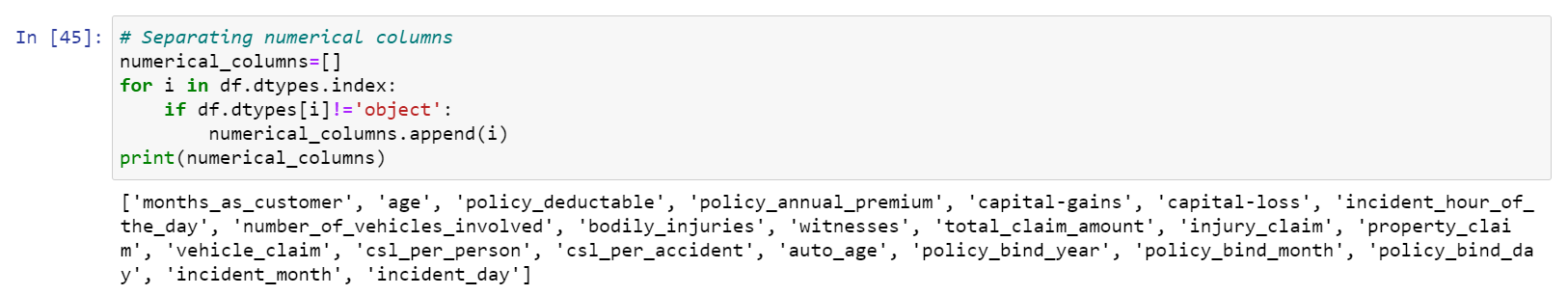
* And in property\_damage column NO has maximum count so I have replaced ‘?’ with NO.



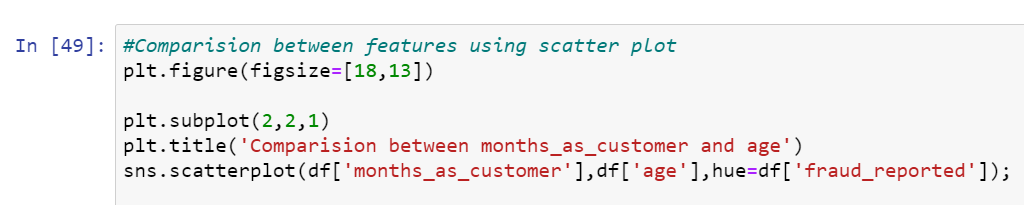
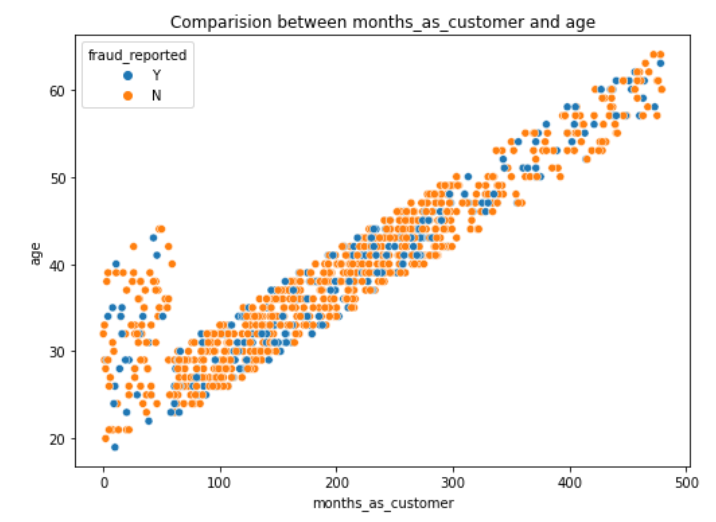
* In police\_report\_available column NO has maximum count so I have to replace ‘?’ with NO.
* Now all the feature extraction is complete and the data is set for analysis.

**Visualization:**

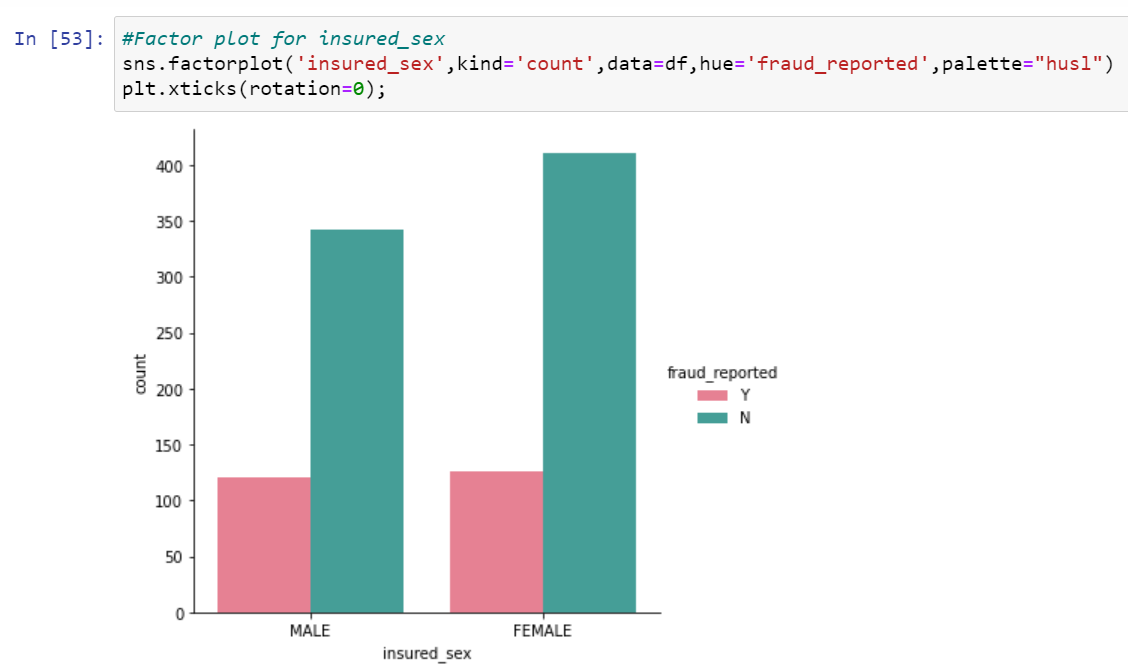
****

****

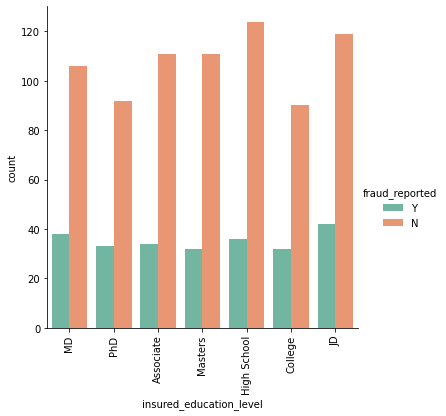
* As a first step we have to separate all numerical and categorical columns.

****

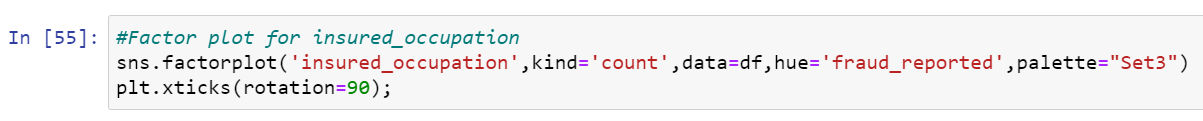
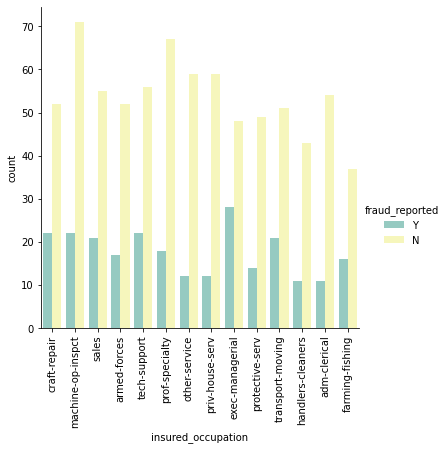
* I have used scatter plot to know the relation between age and month\_as\_customer. I have observed a linear curve with less count of fraud. Which means as the month\_as\_customer is increasing, age is also increasing and the person is getting older the density of both Y and N are decreasing. Which means younger people has the high rate of frauduling.



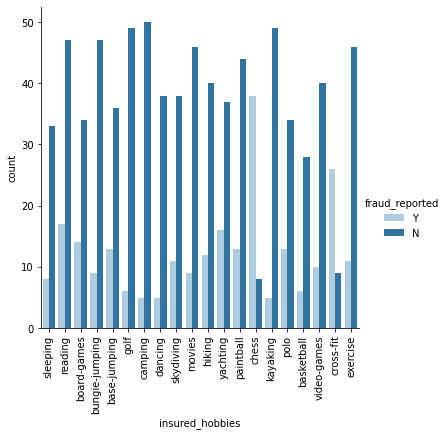
* Above is the factor plot for Insured sex I noticed that in both the genders the count of fraud reported is same. But the fraud not reported is high with females. Which means females are more trust worthy than men.



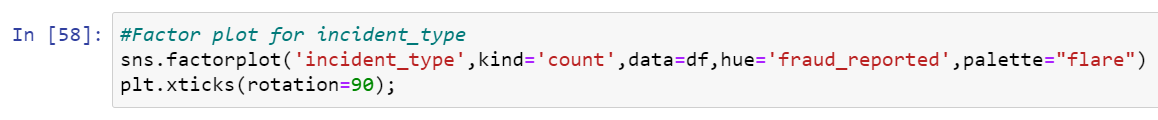
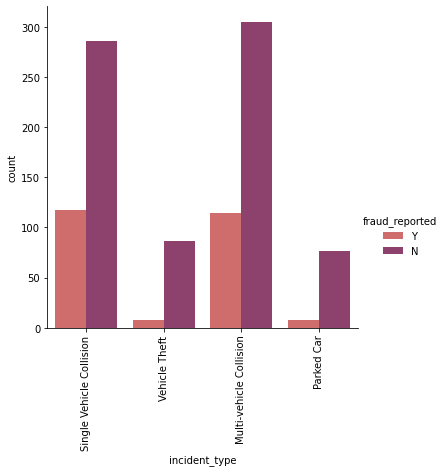
* Look at the factor plot for insured\_education\_level. The fraud count is less with college level educated and fraud not reported count is high for high school level educated which means less educated people are more trust worthy comparatively.
* Next plot is for insured occupation column. And the 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.



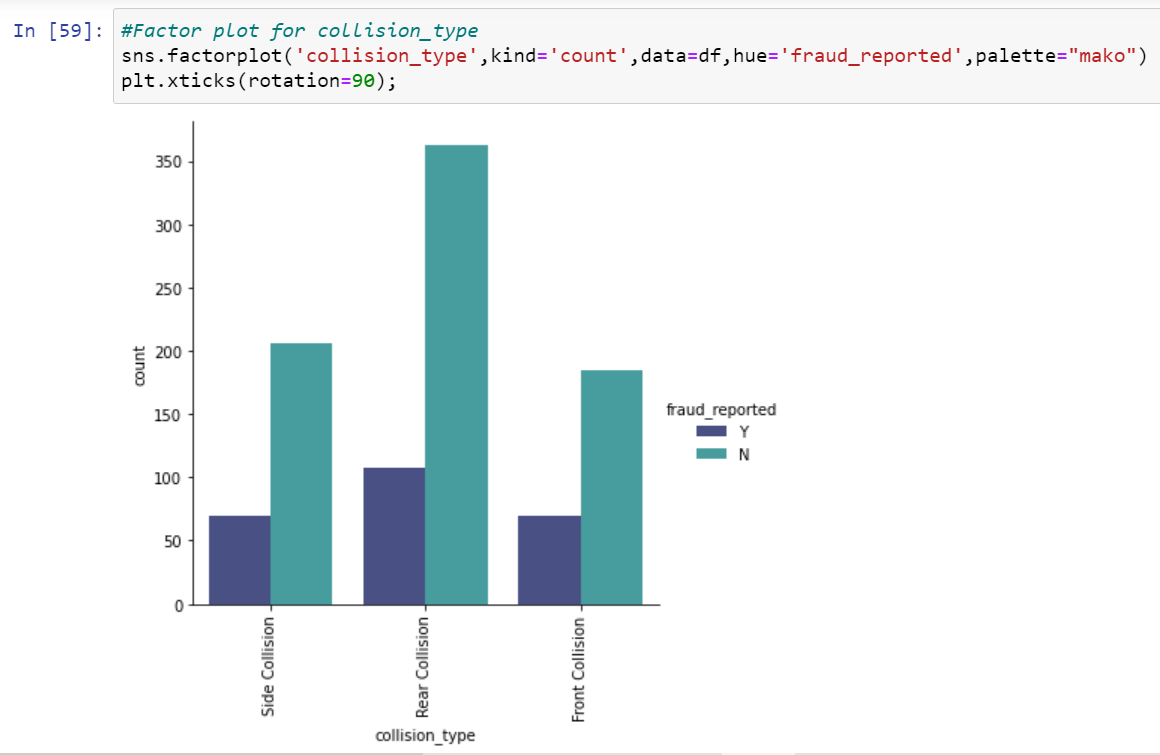




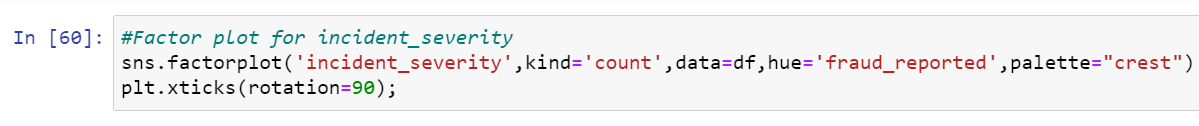
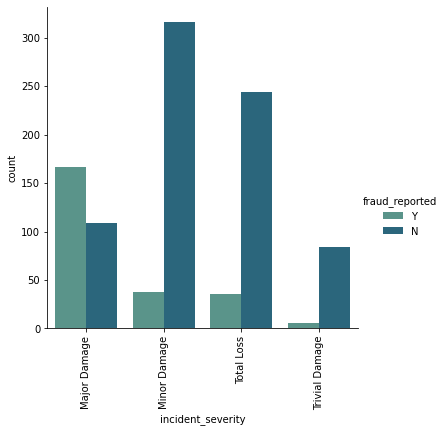
* Have a look into the plot, most of the insured people having chess and cross-fit as a habit are more fraudulent. And fraud not reported count is high for insured people having camping, golf and kayaking as there habit.



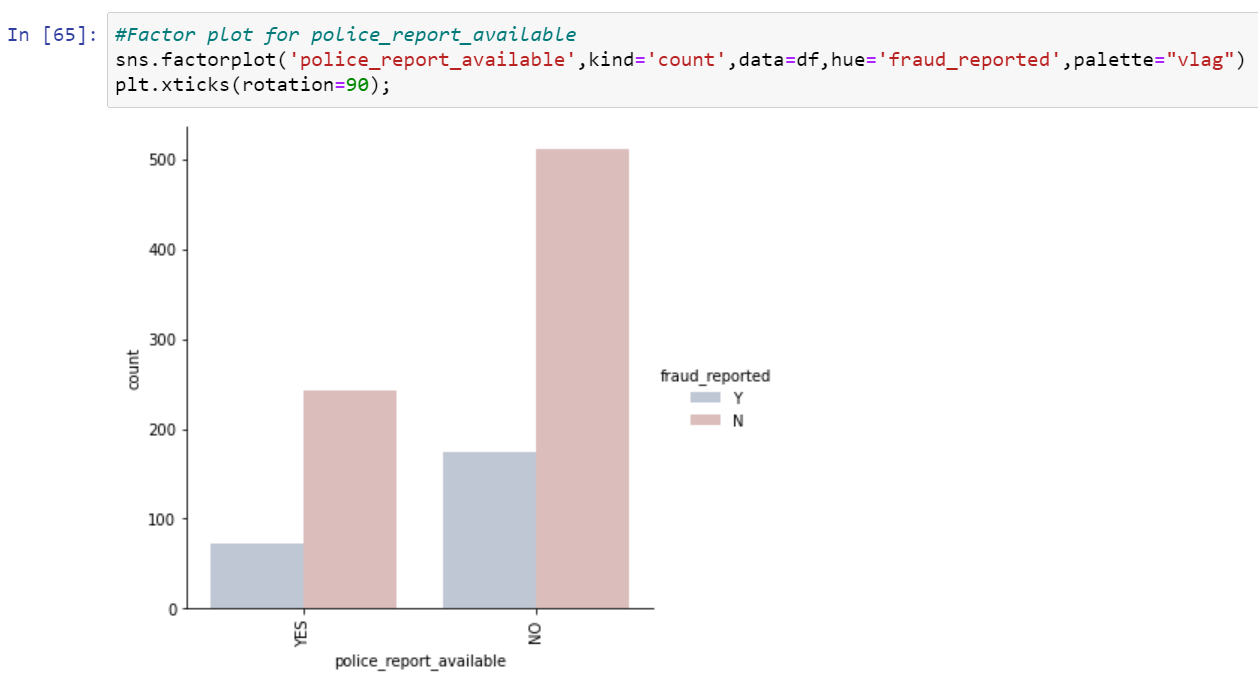
* The count is high for single and multi vehicle collision. The fraud reported and not reported is almost same for these two types of collision.



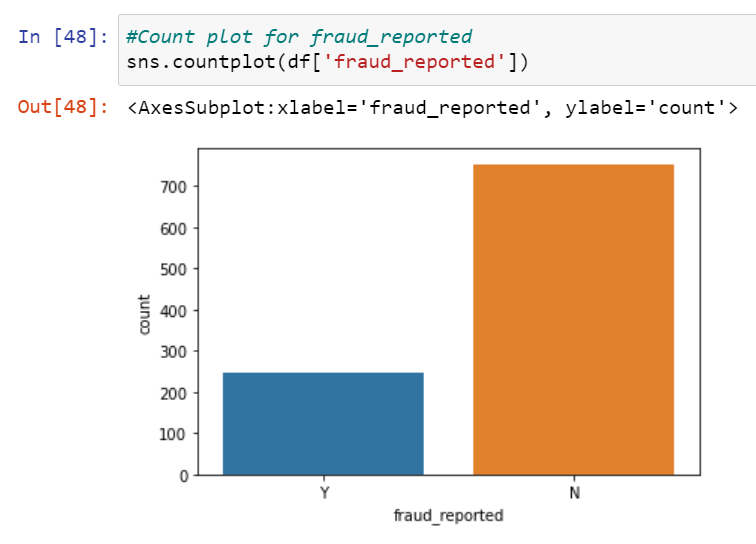
* The count is high for Rear collision and in that case the fraud detected is less compared to fraud not reported.



* This perticular plot is for incident\_severity. If a person is climbing insurance on the condition of major damage then the chance of fraud reporting is very high than any other cases.



* If the police report is not available then the chances of fraudulent is high. So while assuring the claim checking police report may help the insurance company.



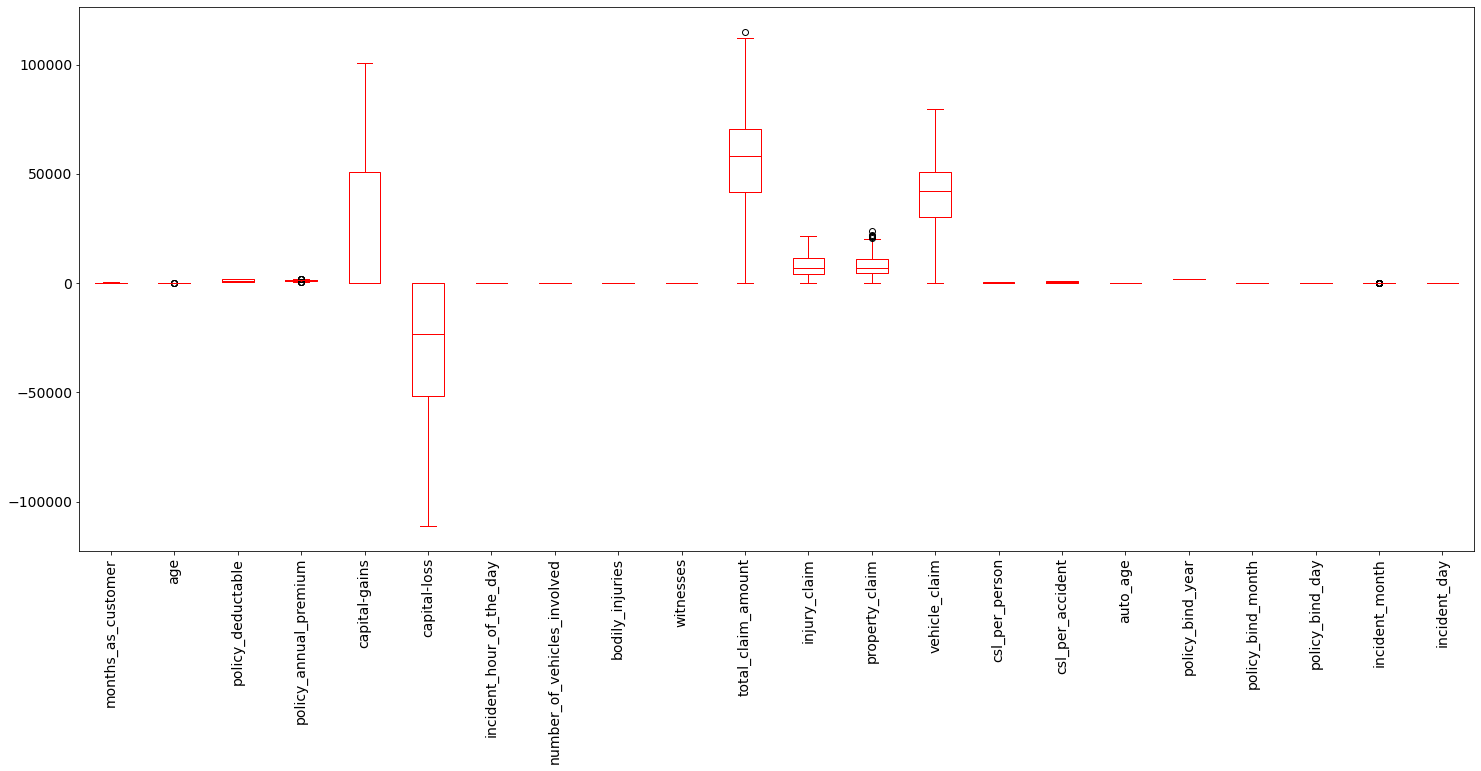
* Since our problem was classification. And the target column fraud\_reported is imbalanced. The count of Y is less compared to N. We have to balance these counts to get a good model.

**3.EDA Concluding Remark:**

* I have checked for NaN values and I found there was no 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, 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:**

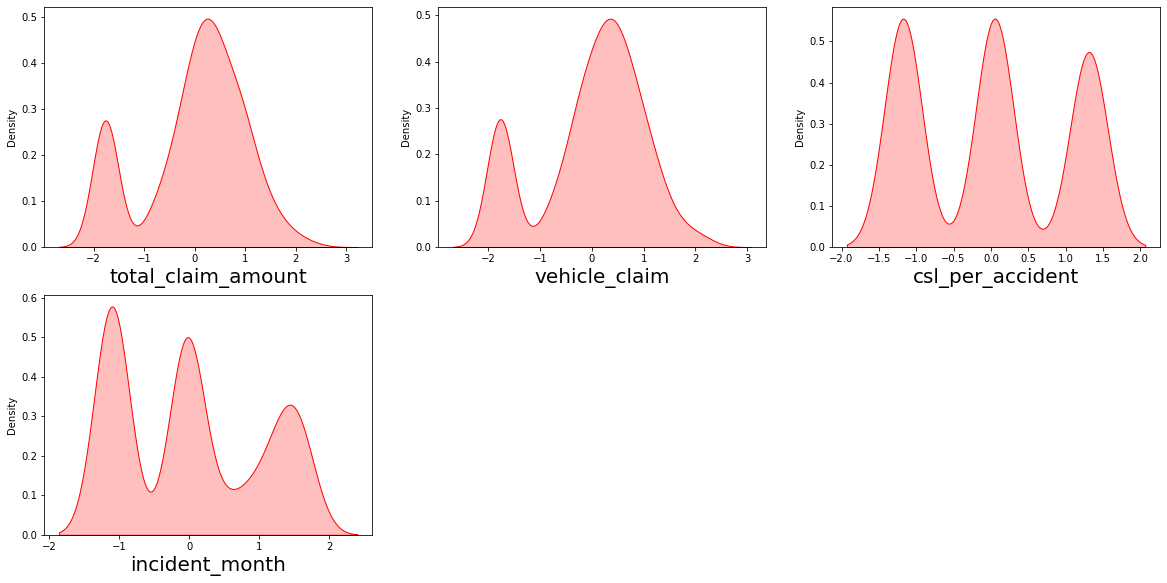
****

****

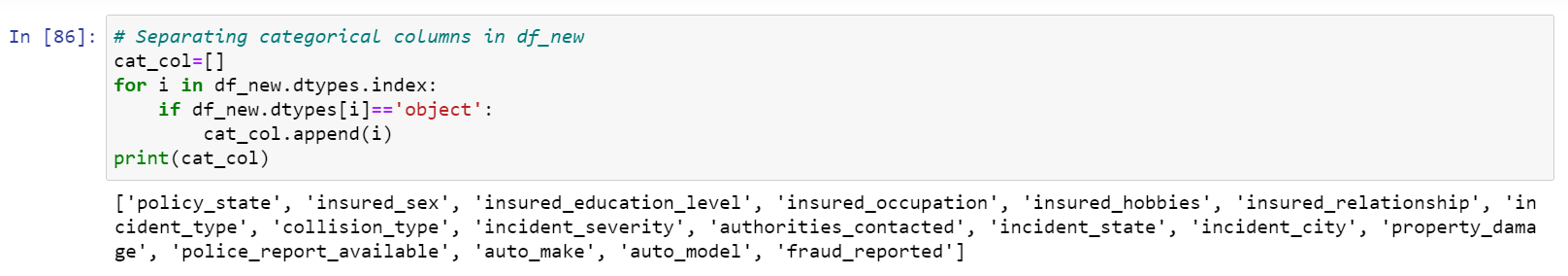
* I have used box plot to check outliers. And I found outliers in age, policy\_annual\_premium, total\_claim\_amount, property\_claim and incident\_month. Now I have to remove outliers in these columns.
* To remove outliers I have chosen zscore with 0.4% dataloss. And also I have gone through IQR but IQR was giving me 5.8% dataloss which was greater than zscore method. So I came to a conclusion to use zscore. After removing the outliers using zscore I have saved the dataset as df\_new.



* 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. I have also tried working on log, log1p, cbrt, sqrt but skewness was not at all reducing. After removing the skewness the distplot of skewed columns is shown below.



* 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 df\_new.

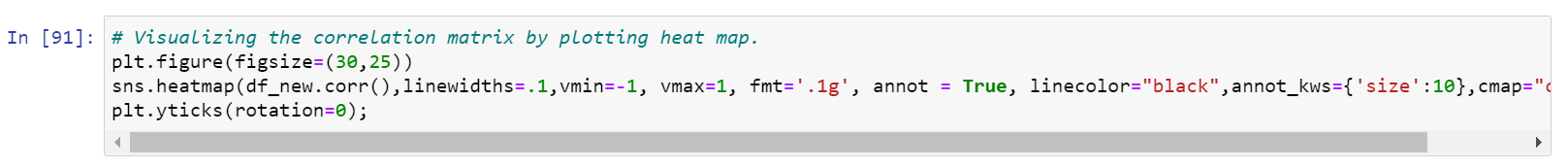


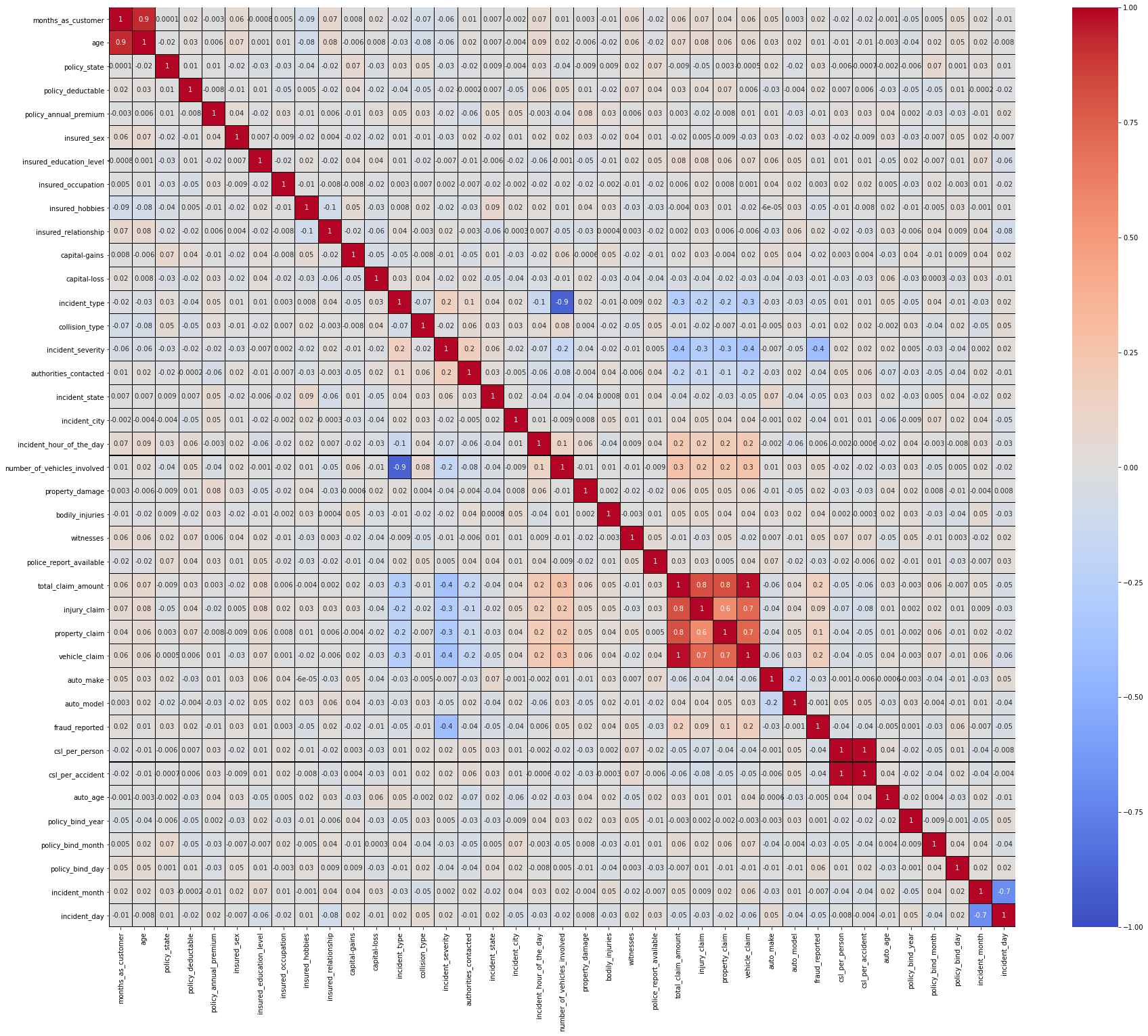


* For all the categorical columns in cleaned dataset df\_new I have applied label encoding (Also tried with ordinal encoder but it decreased the model accuracy).

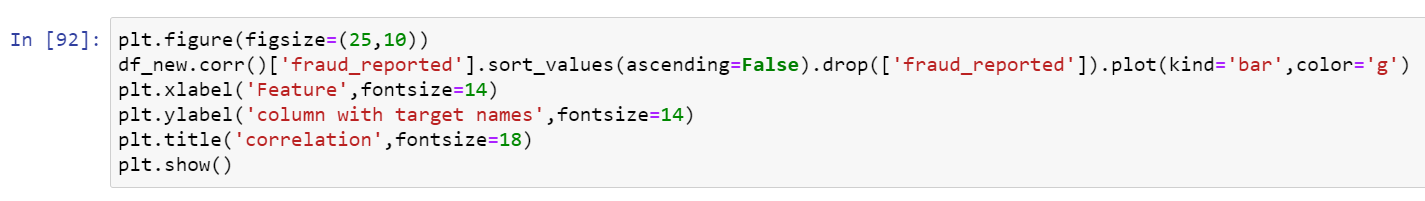
**Checking for correlation using heat map:**

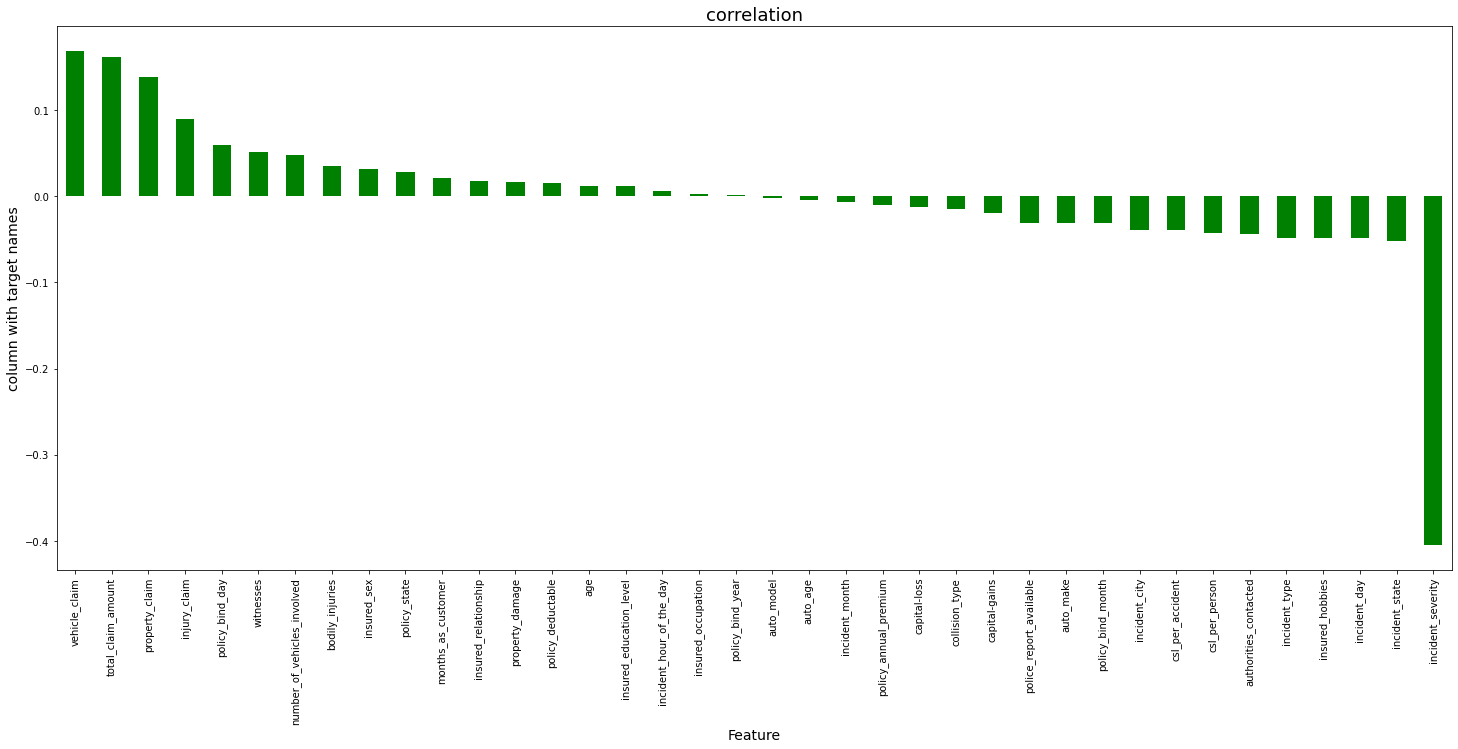
After checking the correlation, to get better insight on the corr values I have plotted heat map. And this **correlation** has to be checked for **cleaned dataset**.



****

* 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.

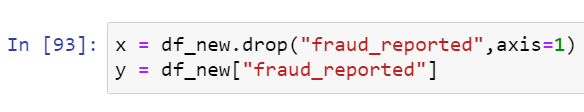
****

****

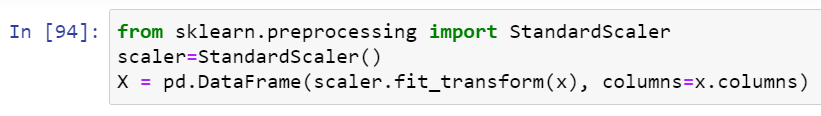
* Policy\_bind\_year, auto\_model, insured\_occupation, auto\_age and incident\_hour\_of\_the\_day are very less correlation with target but let me keep the columns and build the model. Since I don’t want to loose any data so first keeping all the columns let me build the model. After looking into the 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.

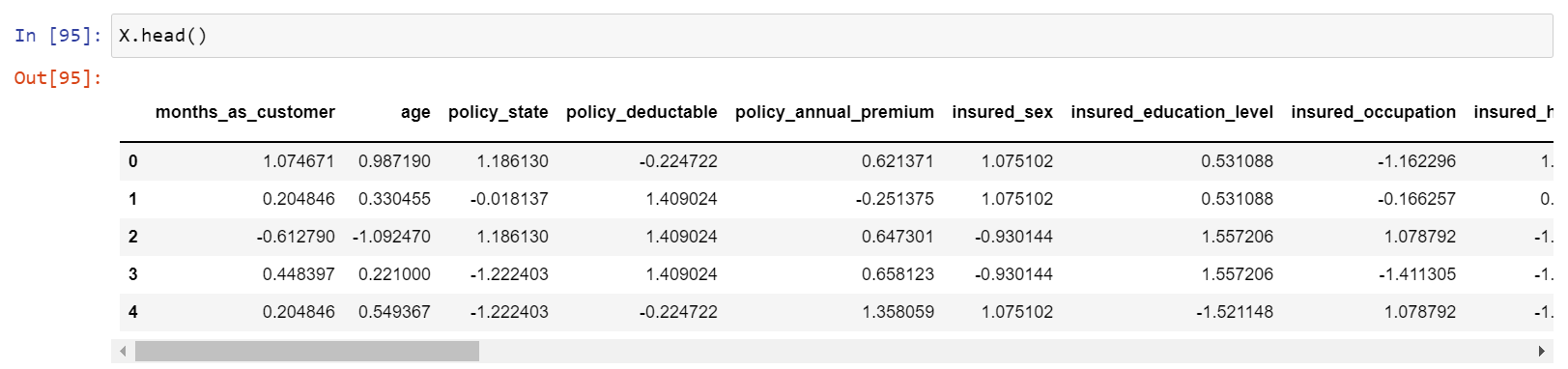
**4.Preprocessing Pipeline:**

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

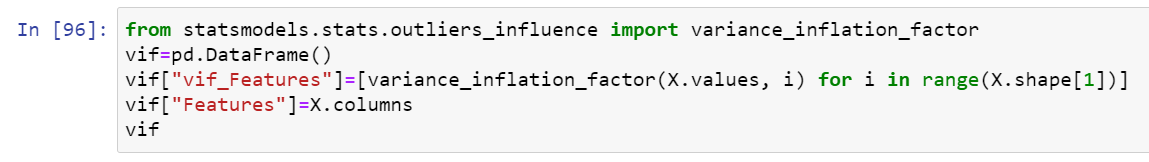


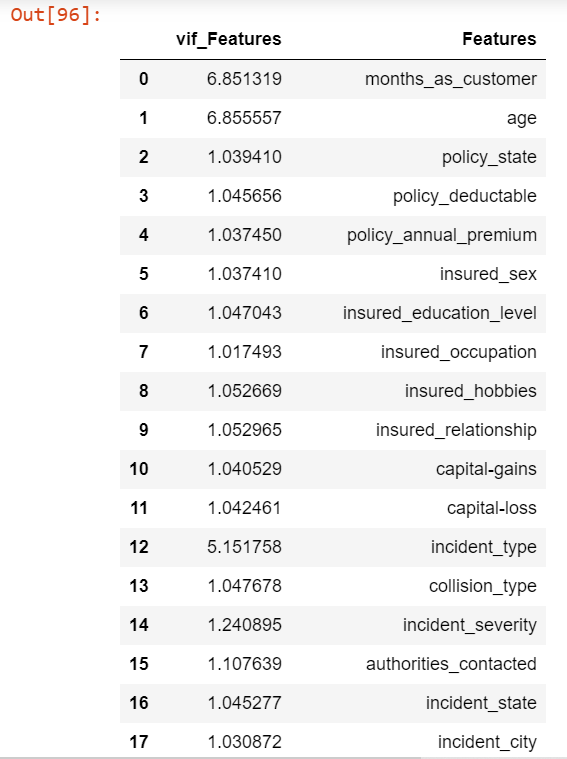
* I have taken x as all independent features and y as dependent/target feature.
* Then I have to scale my independent features to get the same range in all the columns. If I don’t scale my independent columns then there is a chance that my model may get baised. So In this perticular case I have used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

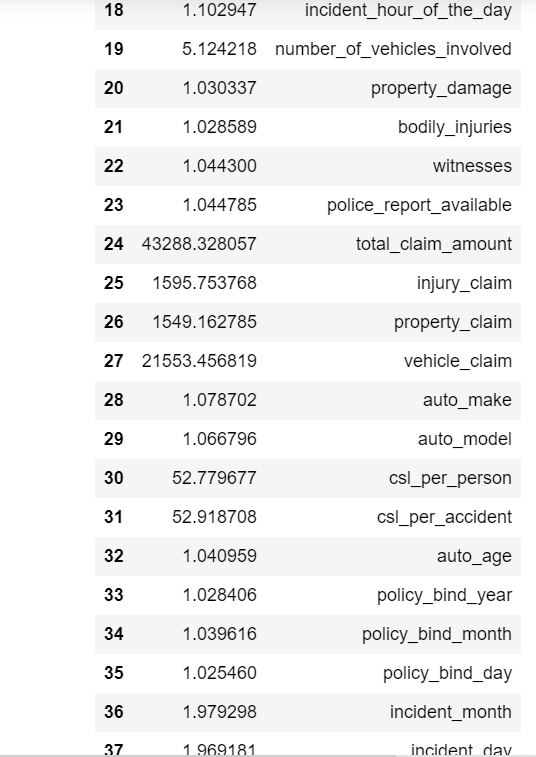




* Now scaling part is done. But I have left out with multicolinearity. I have to check VIF(variance inflation factor) now.

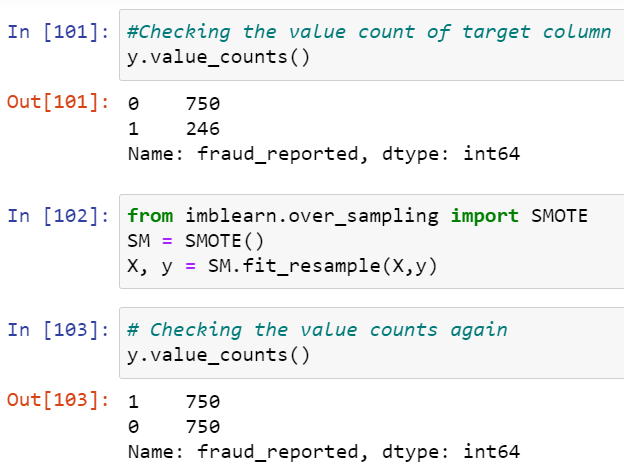


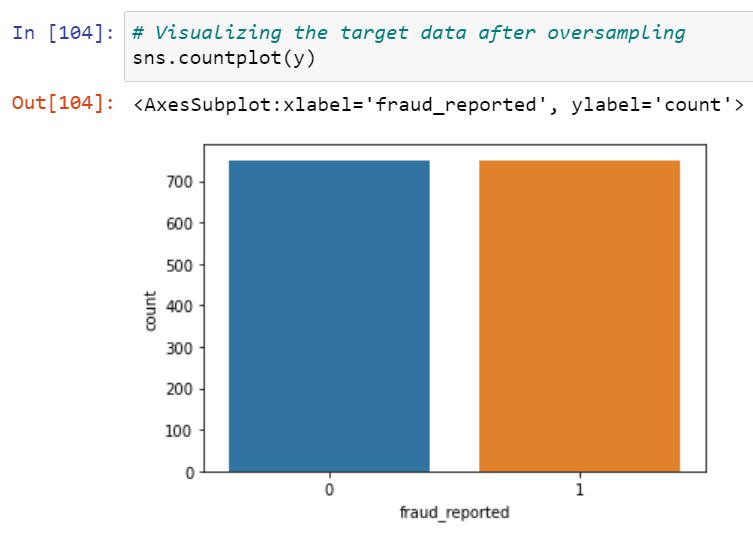




* I can notice a high VIF for total\_claim\_amount, so I have dropped this column first. After that again I checked for VIF and got the highest value for csl\_per\_accident so I dropped this column. Then my multicolinearity issue was solved.

**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.

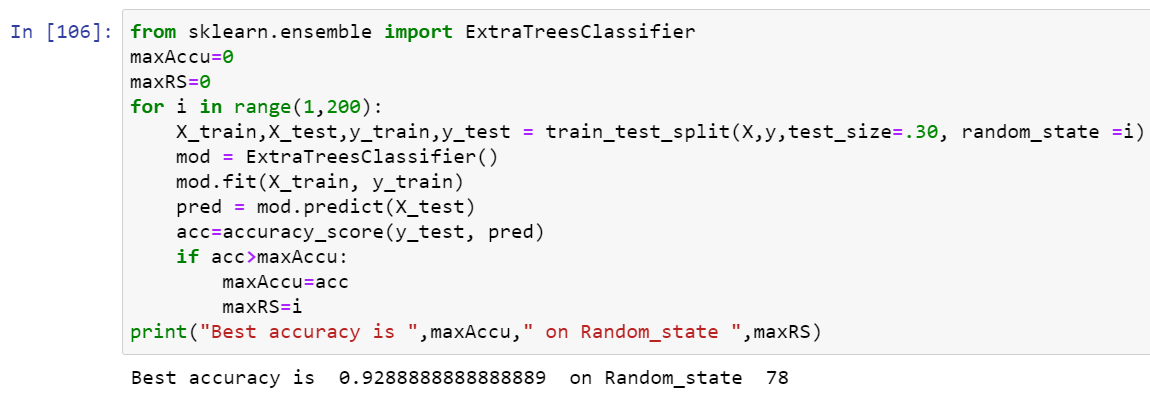


* Now the balance issue is solved and my target is beautifully balanced.
* My data is all set for model building. Let’s go ahead with classification algorithms since this is a Classification Problem.

**5.Building Machine Learning Models:**

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

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



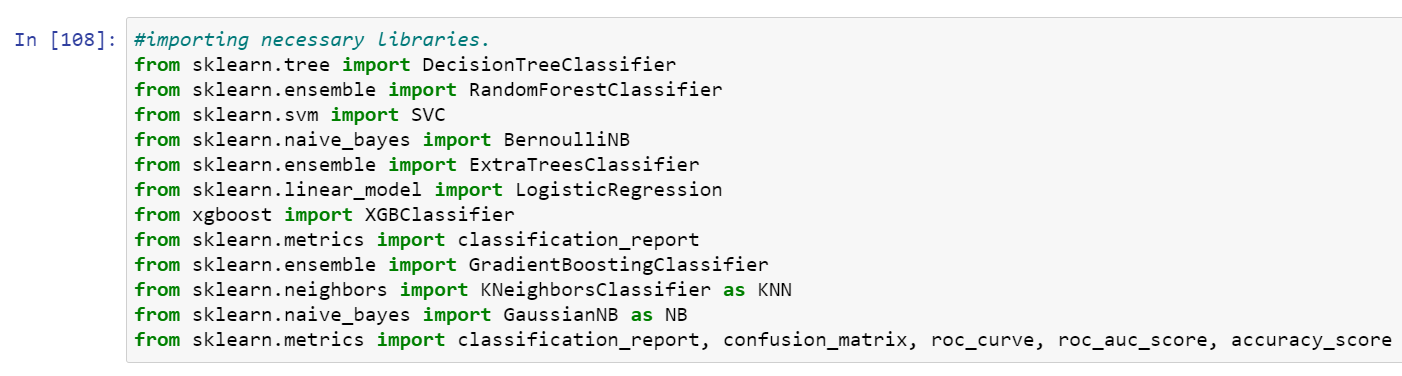
* I got best random state = 78 and accuracy = 92.89%. Now the task is to find the best fitting model.

****

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

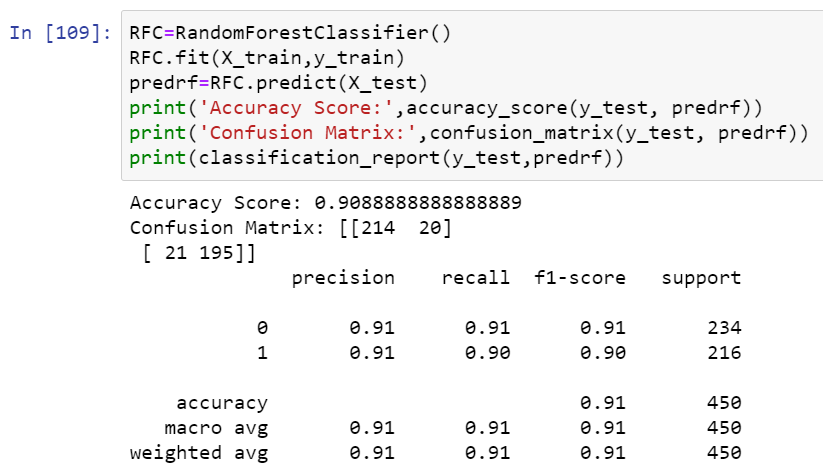
1. **Classification Algorithms:**

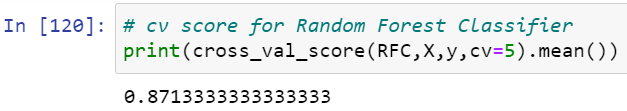
As a first step we have to import all the necessary libraries.



* 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.

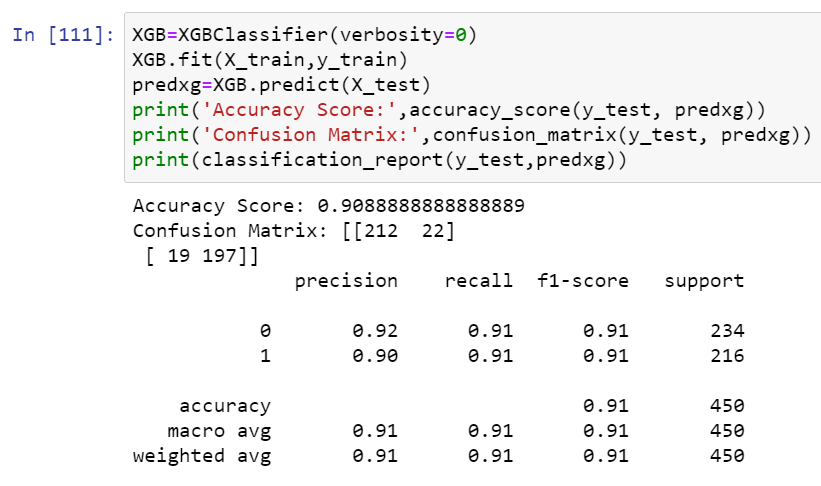
1. **Random Forest Classifier:**

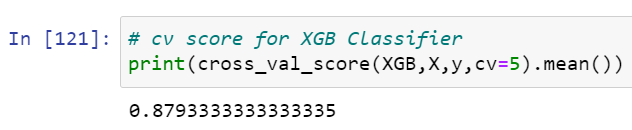
****

****

* Random Forest Classifier model is giving me 91% accuracy\_score and the cross validation is 87.13%. RFC is working good but I can not conclude it as good model before looking into multiple models.

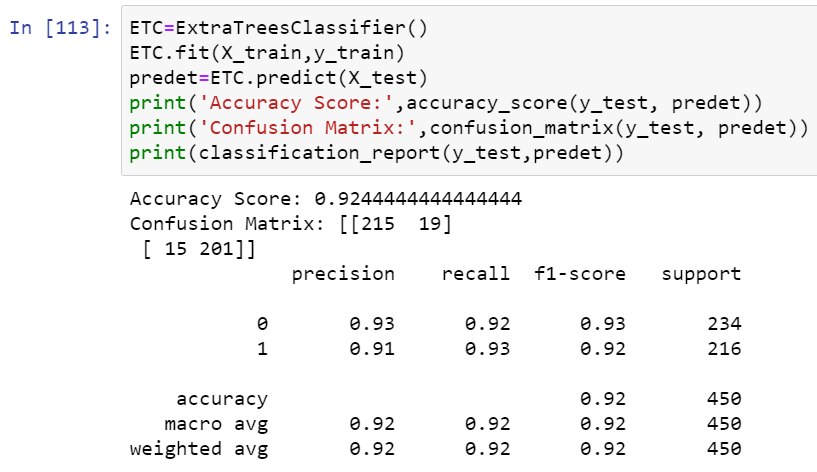
1. **XGB Classifier:**

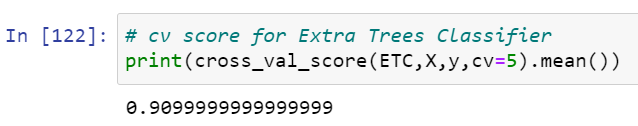
****

****

* XGB Classifier model is giving me 91% accuracy\_score and the cross validation is 87.93%.

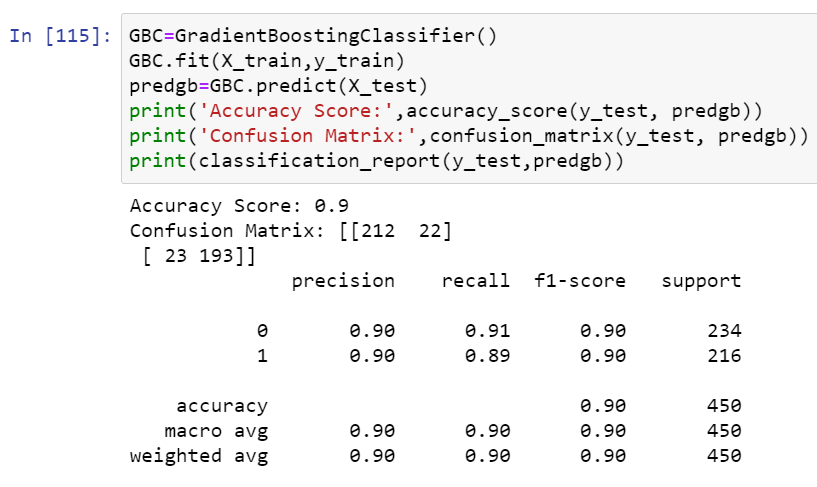
1. **Extra Trees Classifier:**

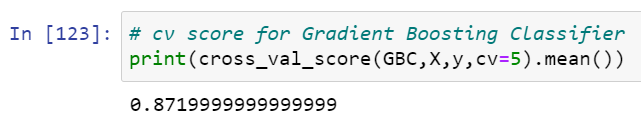
****

****

* Extra Trees Classifier model is giving me 92% accuracy\_score and the cross validation is 91%.

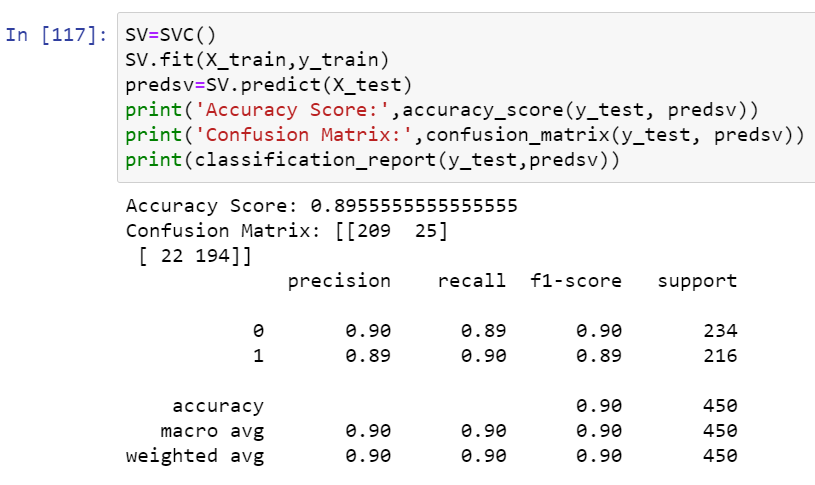
1. **Gradient Boosting Classifier:**

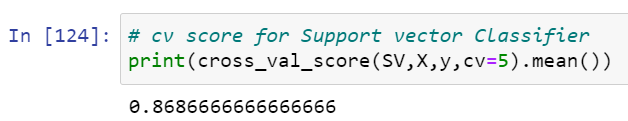
****

****

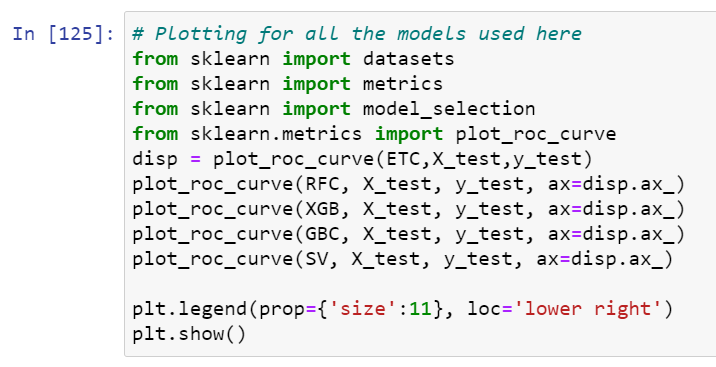
* Gradient Boosting Classifier model is giving me 90% accuracy\_score and the cross validation is 87.20%.

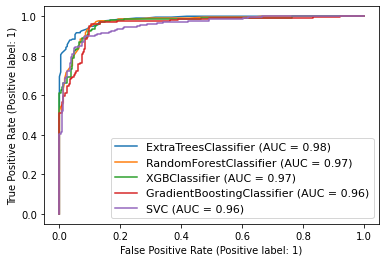
1. **Support Vector Classifier:**

****

****

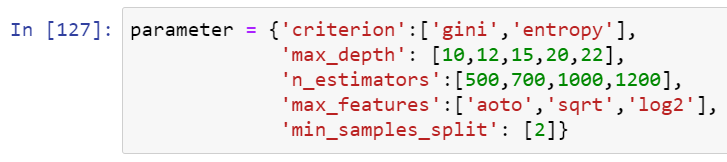
* Support Vector Classifier model is giving me 90% accuracy\_score and the cross validation is 86.87%.
* The ROC-AUC curve for all the above model is as shown below.

****

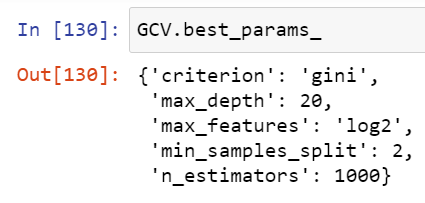
****

* **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 1.44%. And the model accuracy is 92.44% which is good but I can improve the model accuracy by tuning it. Let’s try to improve the model accuracy now.**

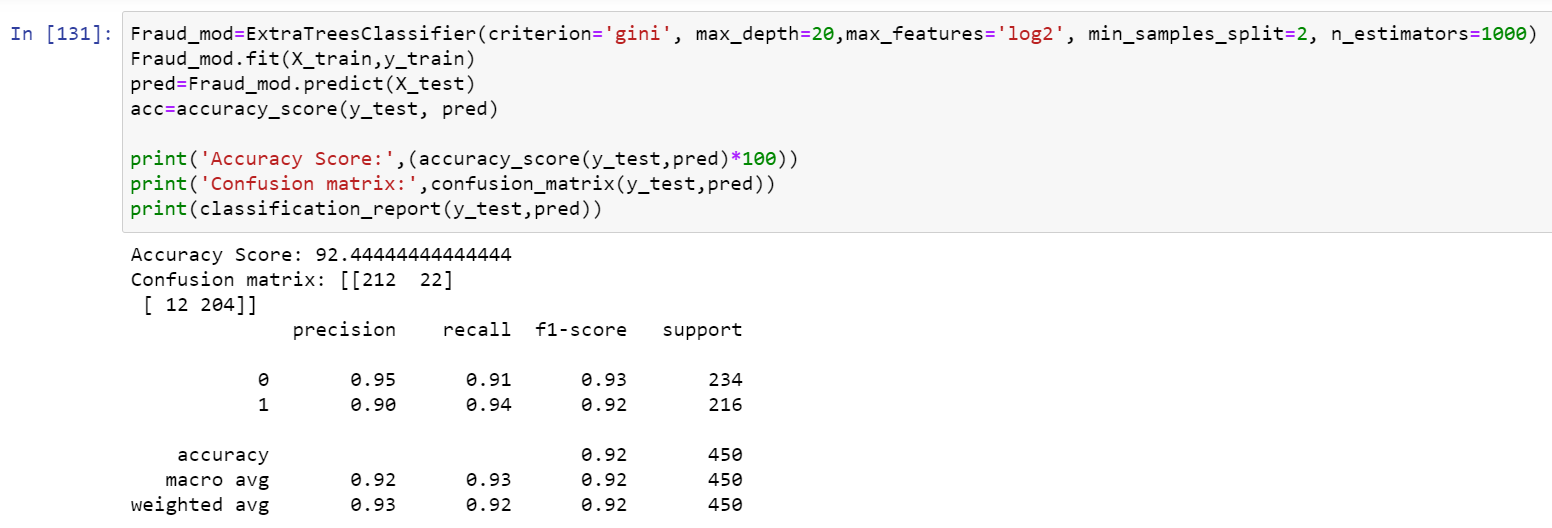
**Hyper Parameter Tuning:**



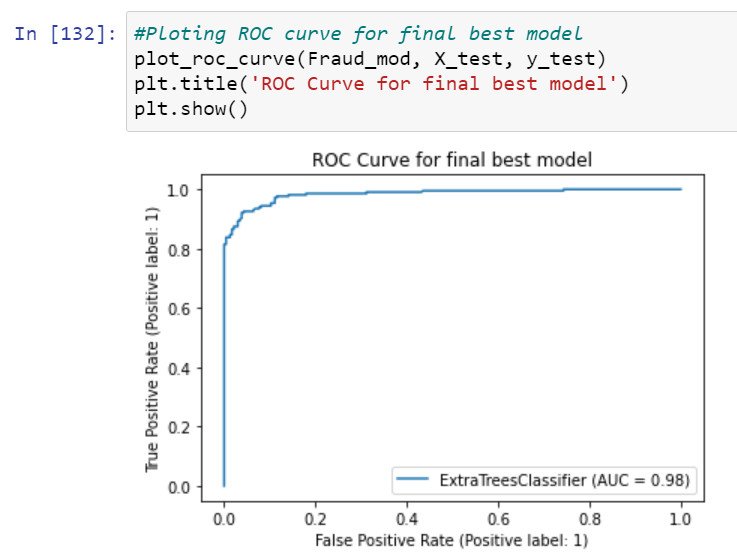
* Using the above parameters list I’m tuning my best model i.e., Extra Trees Classifier. And I have to choose the best parameters in above parameter list, with those parameters I have to build the best model.



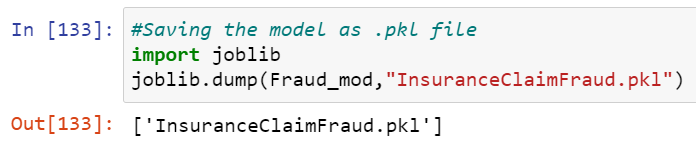
* After knowing the above best parameters I have to run for improving model accuracy.



* Even after tunning the model accuracy is same which means the default parameters used by the model were giving the best accuracy. And the model is now ready with 92.44% accuracy which looks good!!!!.

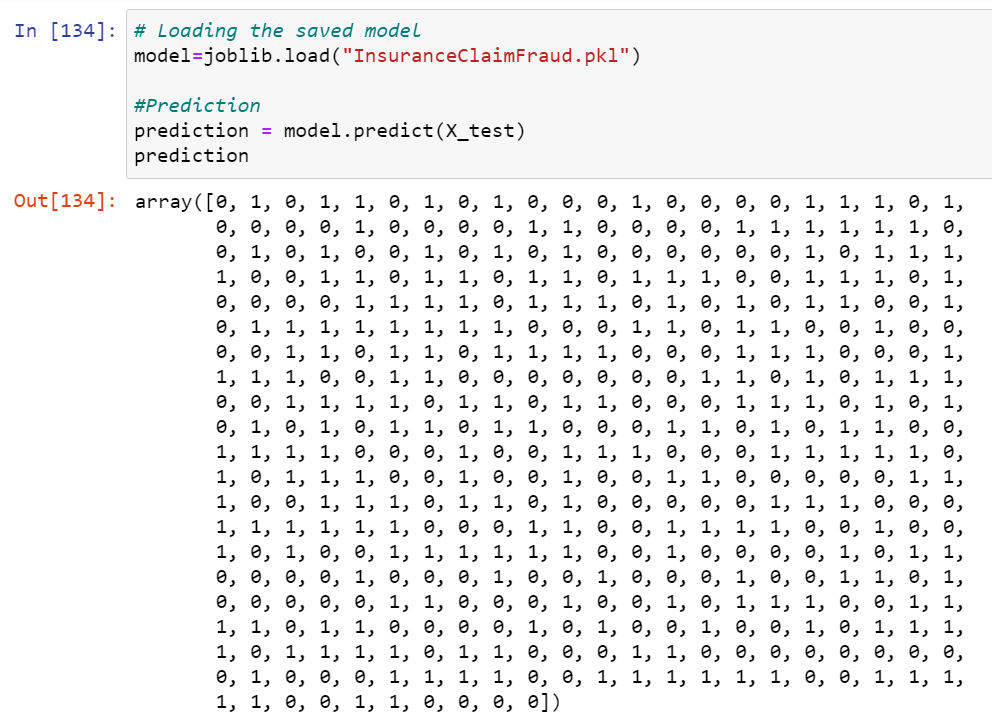


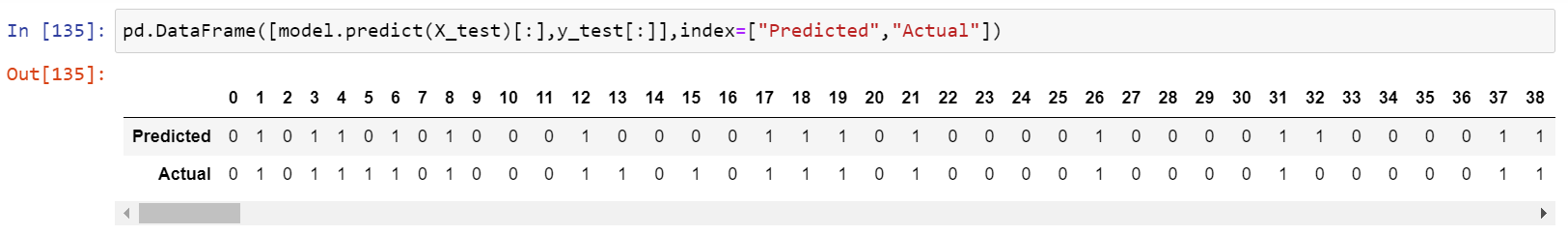
* Above is the ROC curve for best model. And the AUC value also remained same.
* After getting this best model I have saved it using .pkl. As InsuranceClaimFraud.



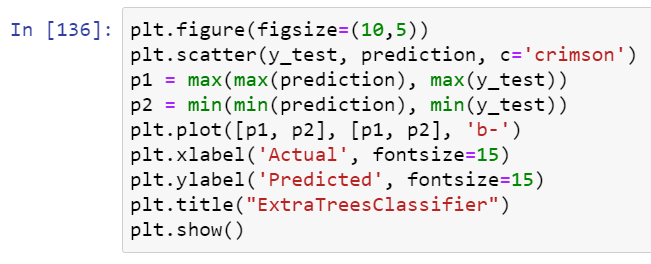
**Predictions:**

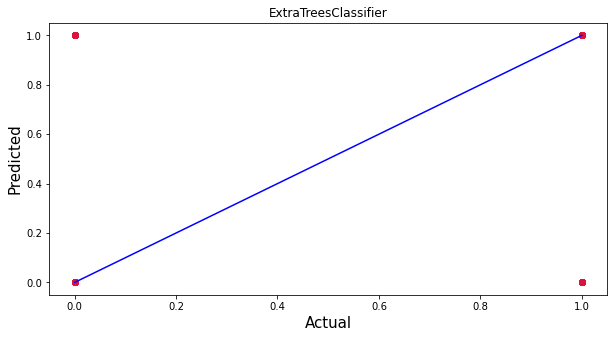
* Now using the saved model I can predict wheather the insurance claim is fraudulent or not.





* After saving the best model we have to load it and check the actual verses predicted values.





* Blue line is the actual values and red dots are predicted values and it’s pleasure to see my model is working good!!!😊.

**6.Concluding Remarks:**

* ****This perticular problem needs a good vision on data, and in this problem Feature Engineering is the most crucial thing.
* You can see how we have handled numerical and categorical data and also how we build different machine learning models on the same dataset.
* Using hyper parameter tunning we can improve our model accuracy, for instance in this model the accuracy remained same.
* Using this machine Learning Model we people can easily predict the insurance claim is fraudulent or not and we could reject those application which will be considered as fraud claims.

****