

**Insurance Claim Fraud Detection**

**AUTHOR**

**ABHISHEK KUMAR**

**BATCH-1842**

****

# **Problem Definition**

* **The objective of this project is to build a predictive model that can detect fraud in insurance. The challenge behind machine learning fraud detection is that frauds are much less common compared to legitimate insurance claims. This type of problem is known as Imbalanced class classification.**
* **Dataset:** <https://github.com/dsrscientist/Data-Science-ML-Capstone-Projects/blob/master/Automobile_insurance_fraud.csv>

**INTRODUCTION**

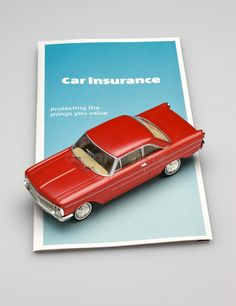
According to the Insurance Information Institute, “Insurance fraud is a deliberate deception perpetrated against or by an insurance company or agent for the purpose of financial gain.” Fraud may be committed at different points by applicants, policyholders, third-party claimants, or professionals who provide services to claimants. Insurance agents and company employees may also commit insurance fraud.

Common frauds include “padding,” or inflating claims; misrepresenting facts on an insurance application; submitting claims for injuries or damage that never occurred; and staging accidents.

People who commit insurance fraud include:

* organized criminals who steal large sums through fraudulent business activities,
* professionals and technicians who inflate service costs or charge for services not rendered, and
* ordinary people who want to cover their deductible or view filing a claim as an opportunity to make a little money.

Some insurance lines are more vulnerable to fraud than others. Healthcare, workers’ compensation, and auto insurance are generally considered to be the sectors most affected.



The auto insurance industry is complicated and involves millions of dollars changing hands every day. And whenever there is a large amount of money running through complex systems, there is opportunity for fraud. This fraud can be committed by professionals and companies working in the industry. But it can also be committed against them.

Insurance fraud can be broadly classified into 2 types.

* Soft Insurance Fraud
* Hard Insurance Fraud

**Soft Insurance fraud** : An example for this is, if the accident has taken place, but the amount of damage that has happened to the vehicle is very less. In such cases, the individual claims to the insurance company that a huge amount of damage has occurred to the vehicle with the goal of charging the insurance company a higher bill.

Hard Insurance fraud : An example for this is, an individual intentionally plans and invests the loss so that he can claim for the insurance from the company. A common example for this type of fraud is staging a car wreck with the goal of benefitting from the resulting claim.

In the project, we focus on the insurance claim data of an Automobile insurance company. Because of fraudulent claims, insurance companies lose large amounts of money, which indirectly affects the public. Therefore, it is important to know which claims are genuine and which claims are fraud.

In this article, we’ll check how to spot insurance fraud and the consequences of engaging in insurance fraud by building machine learning models and getting predictions of which claims are likely to be fraudulent.

**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 with 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 we can create a predictive model that predicts if an insurance claim is fraudulent or not.

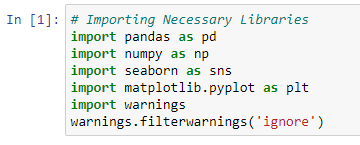
The problem statement explains that the target variable “fraud\_reported” contains the categories, so it is a “Classification Problem”, we need to predict whether an insurance claim is fraudulent or not.

**DATA ANALYSIS**

Data Analysis refers to the process of cleaning, transforming and extracting data to discover useful information for business decision making.

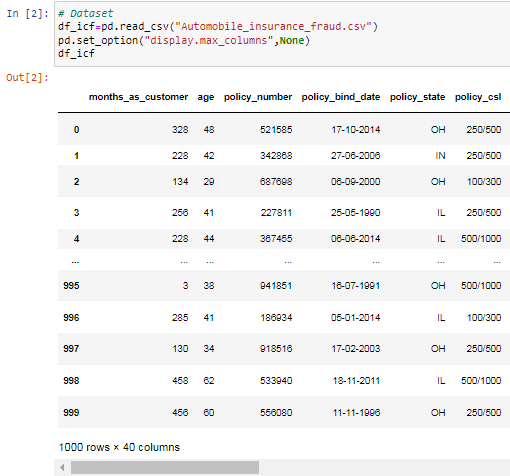
IMPORTING NECESSARY LIBRARIES

We import the libraries necessary for data analysis



IMPORTING THE DATASET

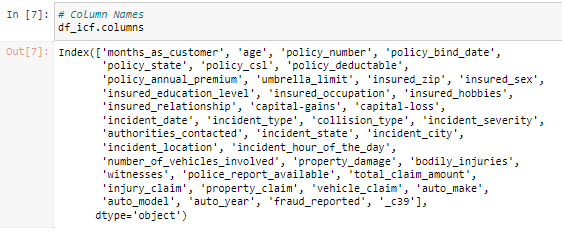
We import the Dataset



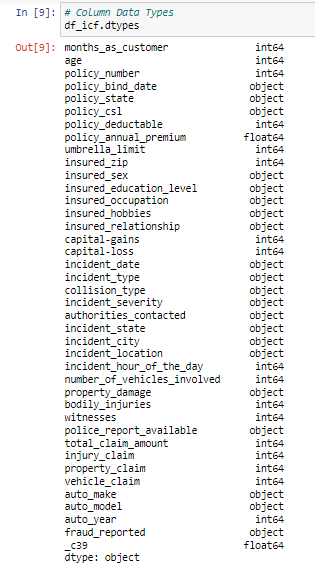
The dataset contains 1000 rows and 40 columns of numerical & categorical data. Next, we check the HEAD( ), TAIL( ) & SAMPLE( ) of the dataset. After this we do some Exploratory Data Analysis (EDA) of the given dataset.

DATA PREPARATION & CLEANING

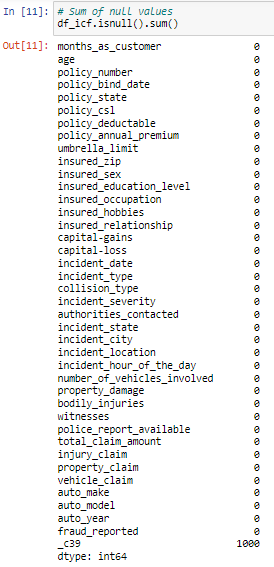
We check the columns present in the dataset

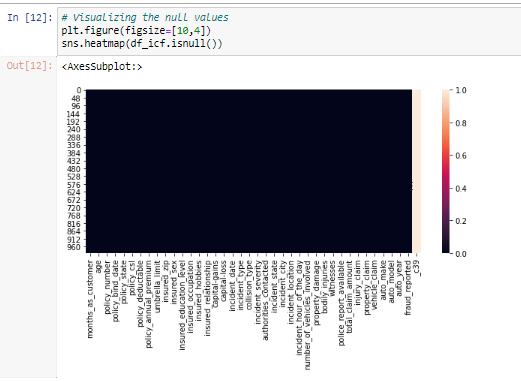


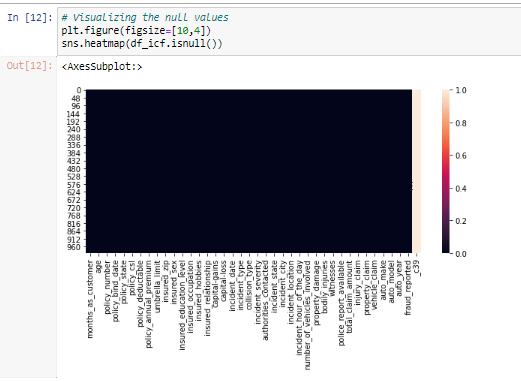
Data types of each column



* We Will check for null values present in the dataset, sum of such null values ( if present) in the dataset & a visual heat map of the null values.







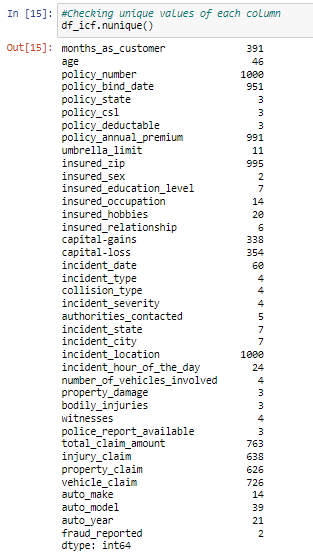
Next, we check the statistical information using “df\_icf.info( )”.



After running df\_icf.info( ), I found the column “c\_39” having one unique count as NAN throughout the dataset and it is of no use, so I dropped that column.



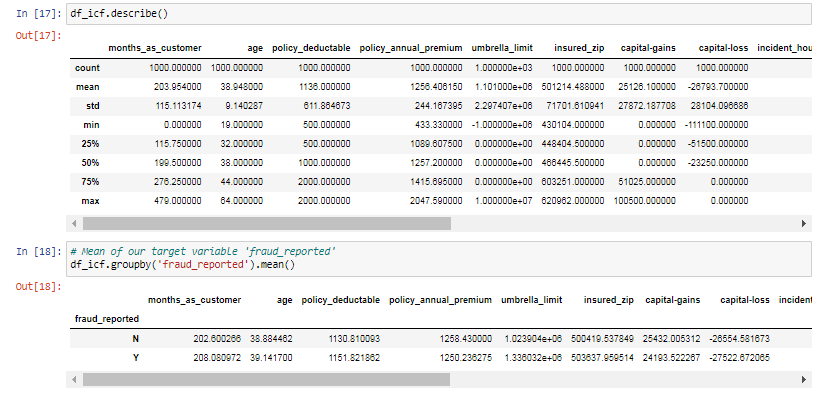
Next, we see the unique values present in each column of the dataset.



After running df\_icf.nunique( ), we see the columns “policy\_number” and “incident\_location” have 1000 unique counts which means all the values in these categorical columns are unique. We can drop these columns as they would be not of much use in model building.



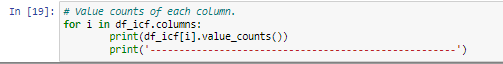
Next, we check the different statistical measurements of all the numerical columns, then specifically our target variable column.



From the above output we find the following observations:

* Here the counts of all the columns are equal which means there are no missing values in the dataset.
* In the columns “policy deductable”, “capital-gains”, “injury \_claim” etc we can observe the mean value is greater than the median (50%) which means the data in those columns are skewed to the right.
* And in the columns “total\_ claim \_amount”, “vehicle \_claim” etc we can observe the median is greater than the mean which means the data in the columns are skewed to the left.
* And in some of the columns the mean and median are equal, which means the data is symmetric and is normally distributed and no skew- ness present.

After this, we check the value counts of every column to see whether there is a need for feature extraction & feature engineering.



By running the above for loop, you will get the value counts of all the columns present in the dataset.

Looking into the value counts of each column, we see that the column “umbrella \_limit” contains about 80% of zero values. It might create a skew -ness problem in the data so it seemed better to drop this column.

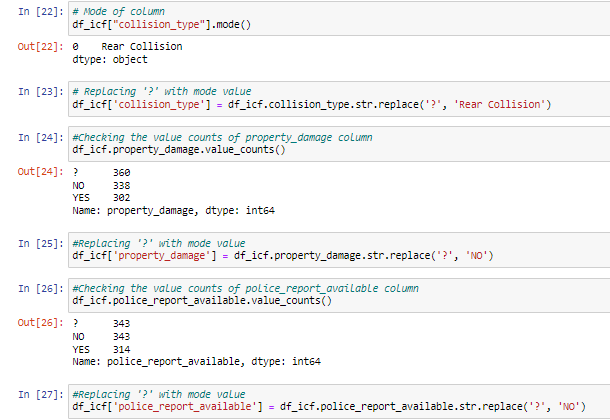


Also, the column “insured \_zip”, contains the zip code given to each person. If we take a look at the value count and unique values of the column, it contains 995 unique values that mean the 5 entries are repeating. Since it is giving information about the identity of the person, it is not important for the processing so we can drop this column as well.



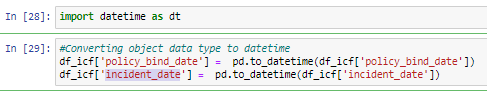
By looking at the dataset and value counts of the various columns, we see some columns having “?” signs. These are not to be considered as NAN values but we need to fill them.

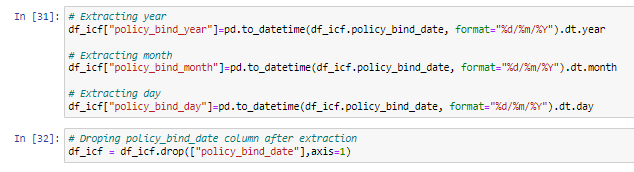
The columns, “collision type”, ”property\_damage” & “police\_report\_available” contain the “?” sign. Since these columns seem to be categorical, we will replace "?" values with most frequently occurring values of the respective columns that are their mode values. In some of these columns the mode and the “?” values are the same so we shall replace the “?” values with the second highest occurring values in the respective columns.

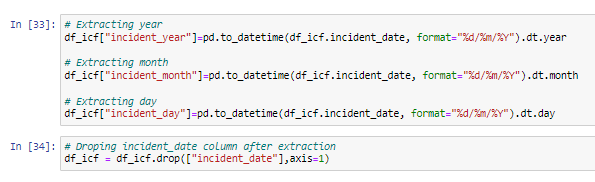


We have now replaced all the “?” values with the respective modes of the respective columns.

Now let us do some feature extraction, we shall first convert the columns, “policy \_bind \_date” & “incident \_date” from object data type to Date Time data type, and extract the year, month and day from these columns. And finally we dropping these columns after extraction.



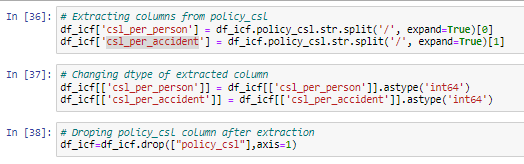




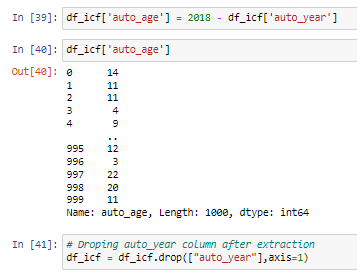
The column “incident \_year” has only one unique value, so we shall drop this column as it will not be useful for model building.



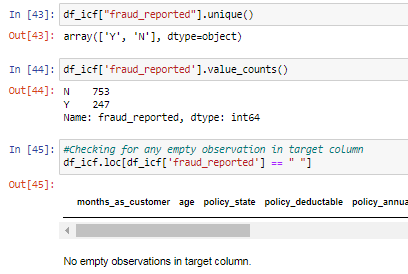
Next, we shall extract the “ csl \_per \_person” & “csl \_per \_accident” from the column “policy \_csl”. After extraction we shall change the data type of these columns to “integer“ data type. Finally we shall drop the “policy \_csl” column.



We then shall extract the “auto \_age” from the column “auto \_year”. Since the data belongs to the year 2018 we shall subtract the auto year from the year 2018 to get the auto age.

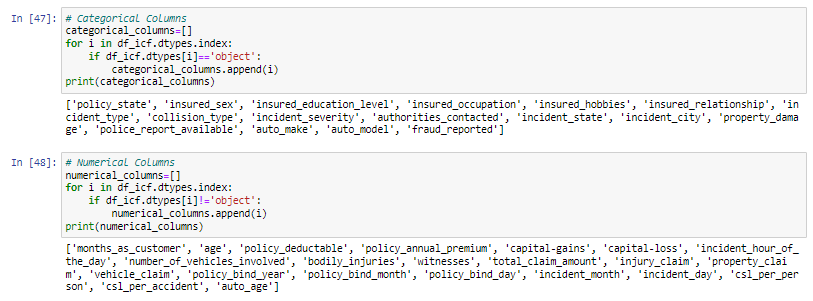


Lastly, before Data Visualization we shall see the unique values present in our target variable column, the value counts of these unique values & lastly, if there are any empty observations in the target variable column.

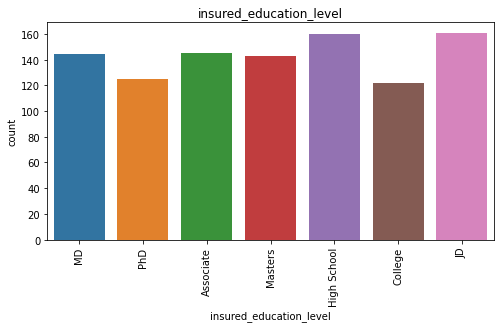


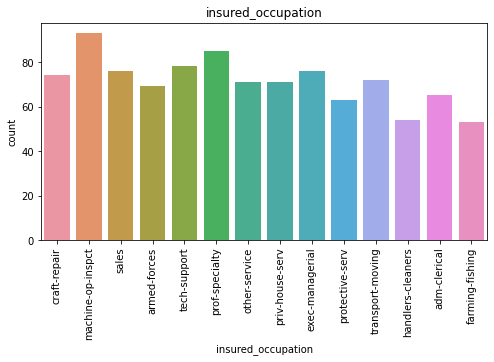
**DATA VISUALISATION**

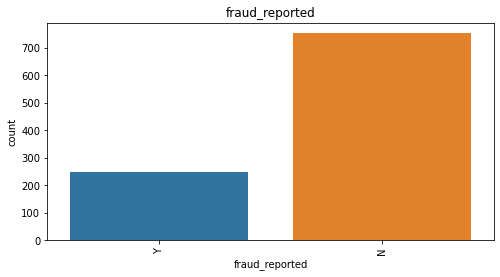
Now we visualize our data. For visualization we shall divide the columns into categorical columns and numerical columns to make visualization better.



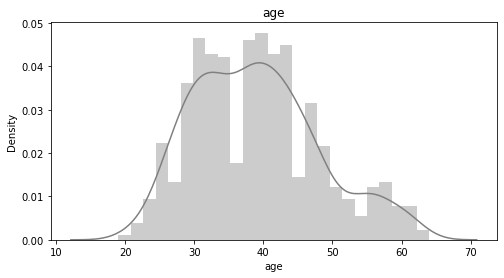
Now for Univariate analysis, we use count plots for plotting the categorical columns of our dataset. Here are a few plots of categorical data:

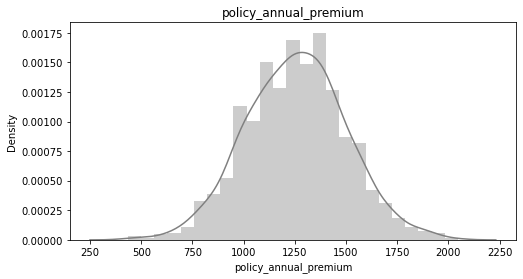


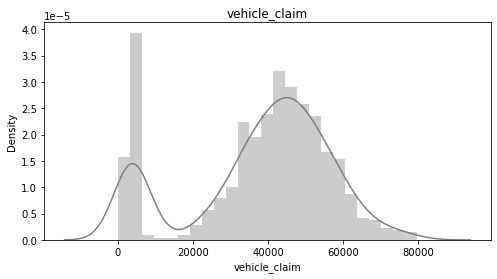
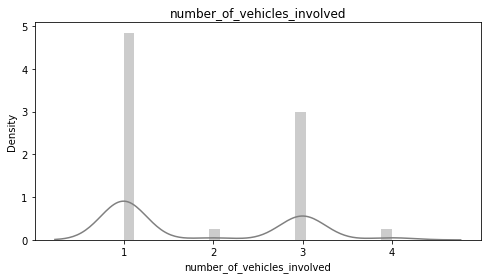




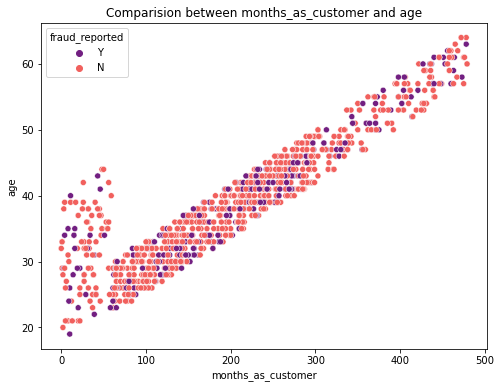
We use dist plots for the numerical columns in our dataset .Here are a few plots of numerical data:



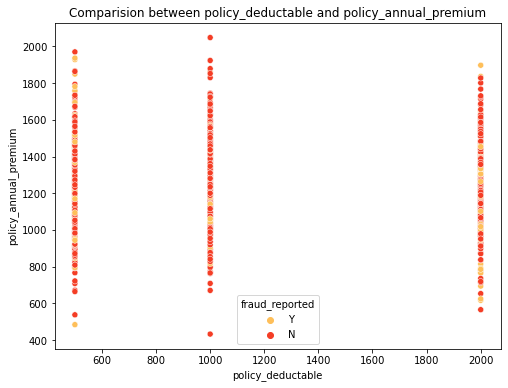




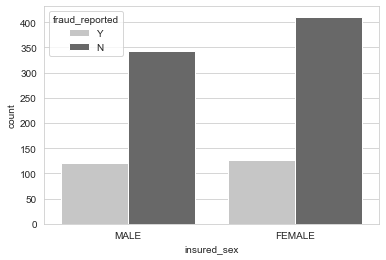
Next we conduct bivariate analysis. For bivariate analysis we used scatter plots & count plots for visualisation.



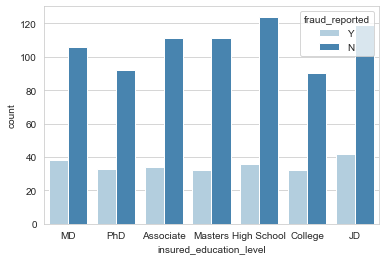
From the above scatter plot, we can observe a strong linear relationship between the “age” and “month \_as\_ customer”. As, month \_as\_ customer increases, the age of the person also increases. Also, as the person gets older, the frequency of the both fraud reported classes are vanishing slowly. That means, the people having young age are more likely to have high fraud reports.



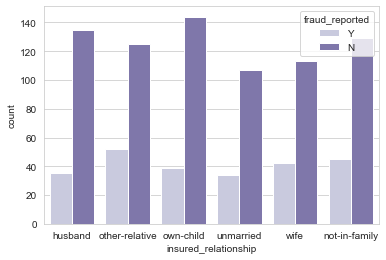
From the above scatter plot, we can observe that in the “policy \_ annual \_premium” range of 400 to 2000 the “policy \_ amount \_deductable” of 1000, which is the highest count among the policy deductible amounts, has the least amount of “fraud \_reported” as Y and a higher number of “fraud \_reported” as N compared to other policy deductible amounts.



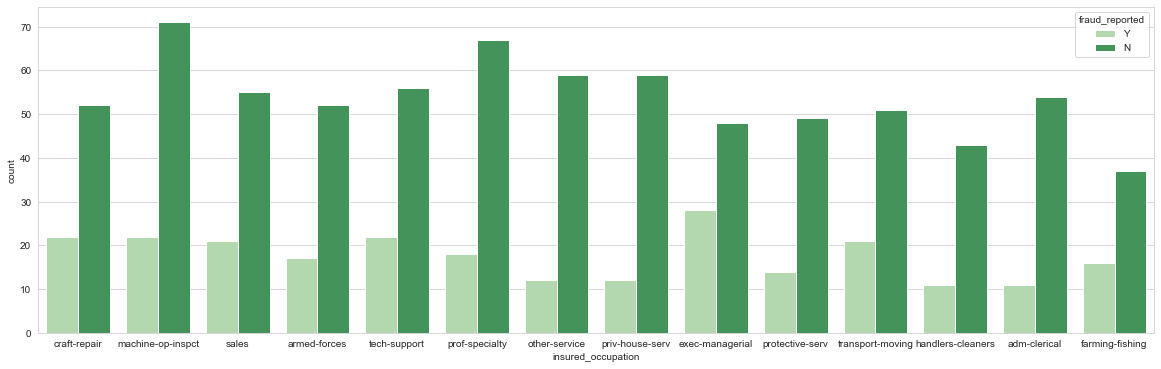
Above is the count plot to compare “insured \_sex” and “fraud \_reported”. We notice both male and female customers have insurance but the count for females is a bit higher than male counts. The fraud reported data are almost the same in both genders but the non-fraud reports are a bit high in case of female , graph shows that the female customers are more trustworthy than male customers.



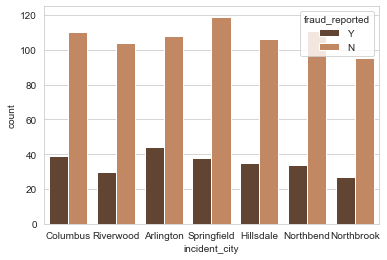
From the above count plot we can observe that the fraud \_reported is very less for the people who have high school education and the people who have completed their "JD" education have high fraud \_reported among others. That means the people with less education level are more trustworthy.



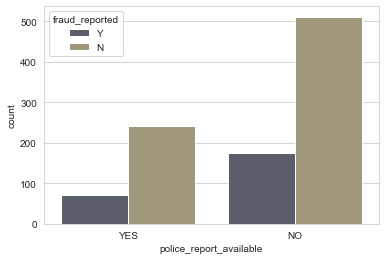
From the above count plot we observe the fraud \_reported is very less in people who take insurance for their own-children, followed by people who take insurance for their husband. The fraud \_reported is highest in people who take insurance for their ‘other-relative’, followed by insurance taken for ‘not-in-family’. It concludes that insurance taken for own-children, husband & wife are usually trustworthy.



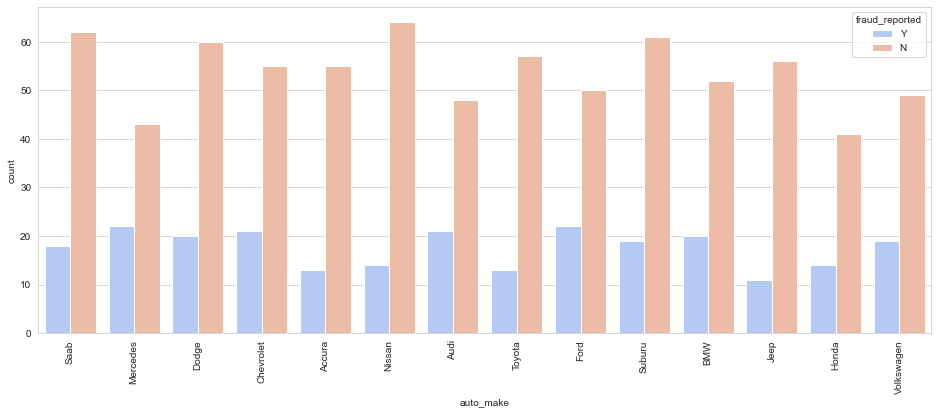
From the above count plot, we observe less fraud \_reported if the occupation of the insured is machine operation inspector followed by professional uniqueness. Apart from this all the other insured occupations have almost the same counts. The people whose occupation is exec-managerial have high fraud reports compared to others.



From the above count plot we see the highest number of fraud \_reported as no in the city of Springfield, followed by Columbus & North bend. The highest number of fraud \_reported as yes is in the city of Arlington.



From the above count plot we see that the police report is not available in most cases when the fraud\_ reported is no, compared to when the fraud\_ reported is yes. If there are no police reports available then the fraud\_ reported is very high.



From the above count plot, the fraud\_ reported was the least with the automaker being Nissan, followed by Saab, Subaru & Dodge. The fraud\_ reported was high when the automaker was Ford, Mercedes, Audi or BMW.

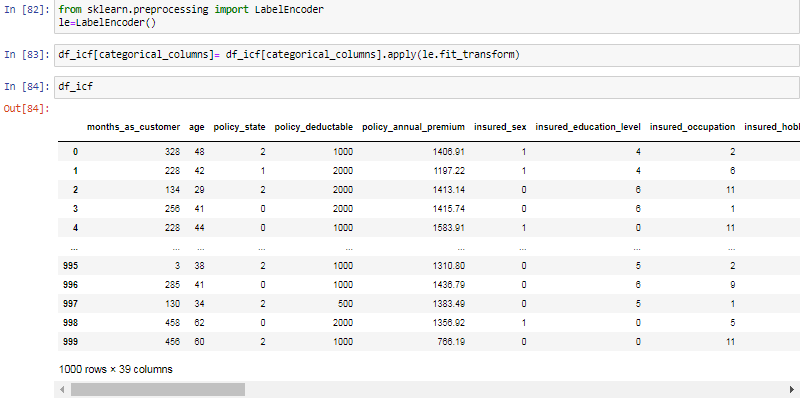
In a similar manner, we plot the graphs for other columns and make good data visualization.

**EDA CONCLUSION**

* We have checked the null values in the dataset and there was no missing values found. ( One column “\_c39” with only “NaN” values was dropped)
* We have dropped some of the irrelevant columns ( “policy\_ number”, “incident\_ location”, “umbrella\_ limit”, “insured\_ zip”) to overcome the multicollinearity problem.
* Replaced the corrupted entries “?” in the columns with their respective mode values.
* Extracted some new features from the existing features to get better results without any hindrance. And dropped the old columns, if I keep them as it is they will act as duplicates and that leads to a multi collinearity problem.
* Coming to the visualization part, we have found when and where the fraud reports are high in number.
* To get the better insights about the features, I have used count plots, box plots, pair plots, pie charts, scatter plots and distribution plots.

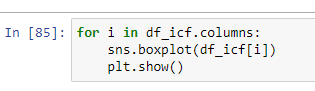
**ENCODING THE DATA FRAME**

Since our dataset contains many columns with object data type, we need to encode them using any of the encoding methods. Here we apply the label encoding method.

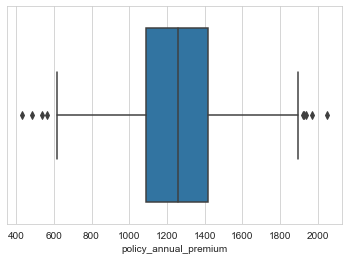
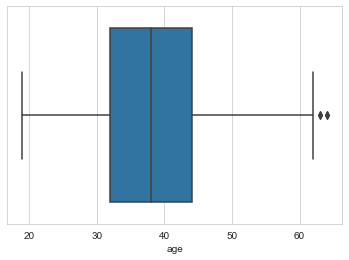


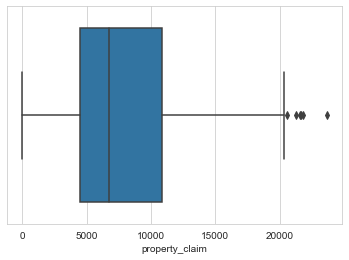
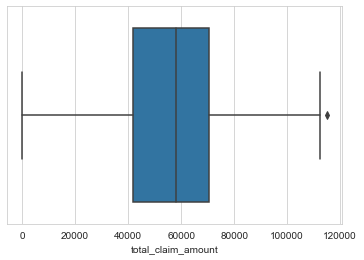
**CHECKING OUTLIERS & SKEWNESS**

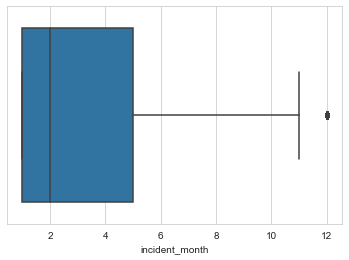
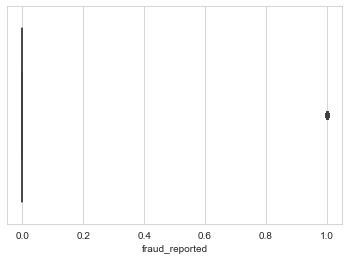
We checked the data for outlier, box plots were used to check each column to find outliers.



Outliers were found in the following columns:

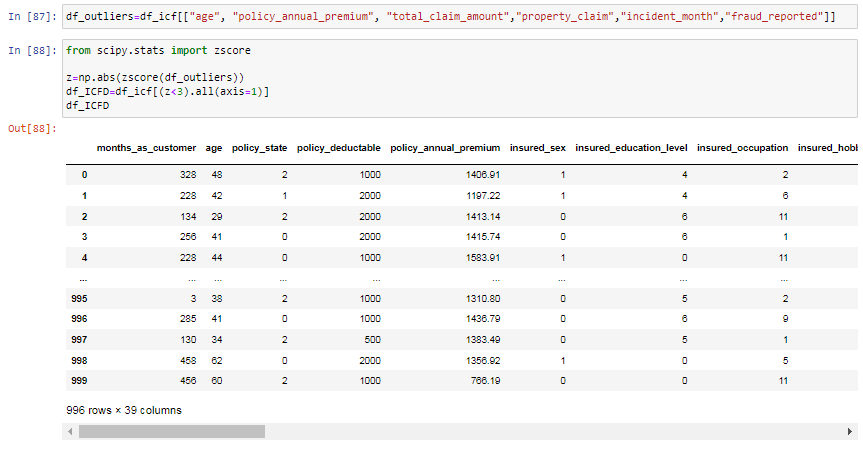






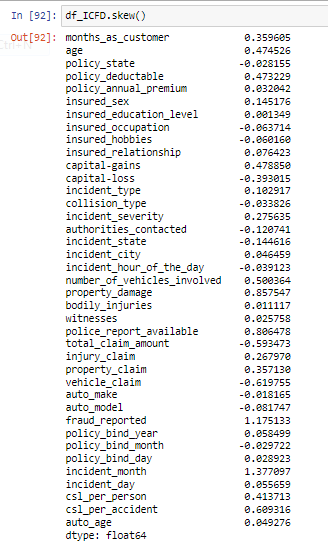
Outliers were found in “age”, “policy \_annual\_ premium”, “total \_claim \_amount”, “property \_claim”, “fraud \_reported” and “incident \_Month”.

So, we removed the outliers using the Z-Score Method.



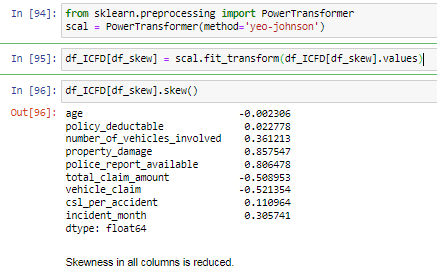
After removing the outliers our data loss was 0.4 %. Which was affordable.

Then we checked for the skew ness of all the columns of the dataset.



There was skewness in `age, policy \_deductable ,number \_of \_vehicles \_involved , property \_damage, police \_report \_available, total \_ claim \_amount , vehicle \_claim , csl \_per \_accident, incident \_month.`

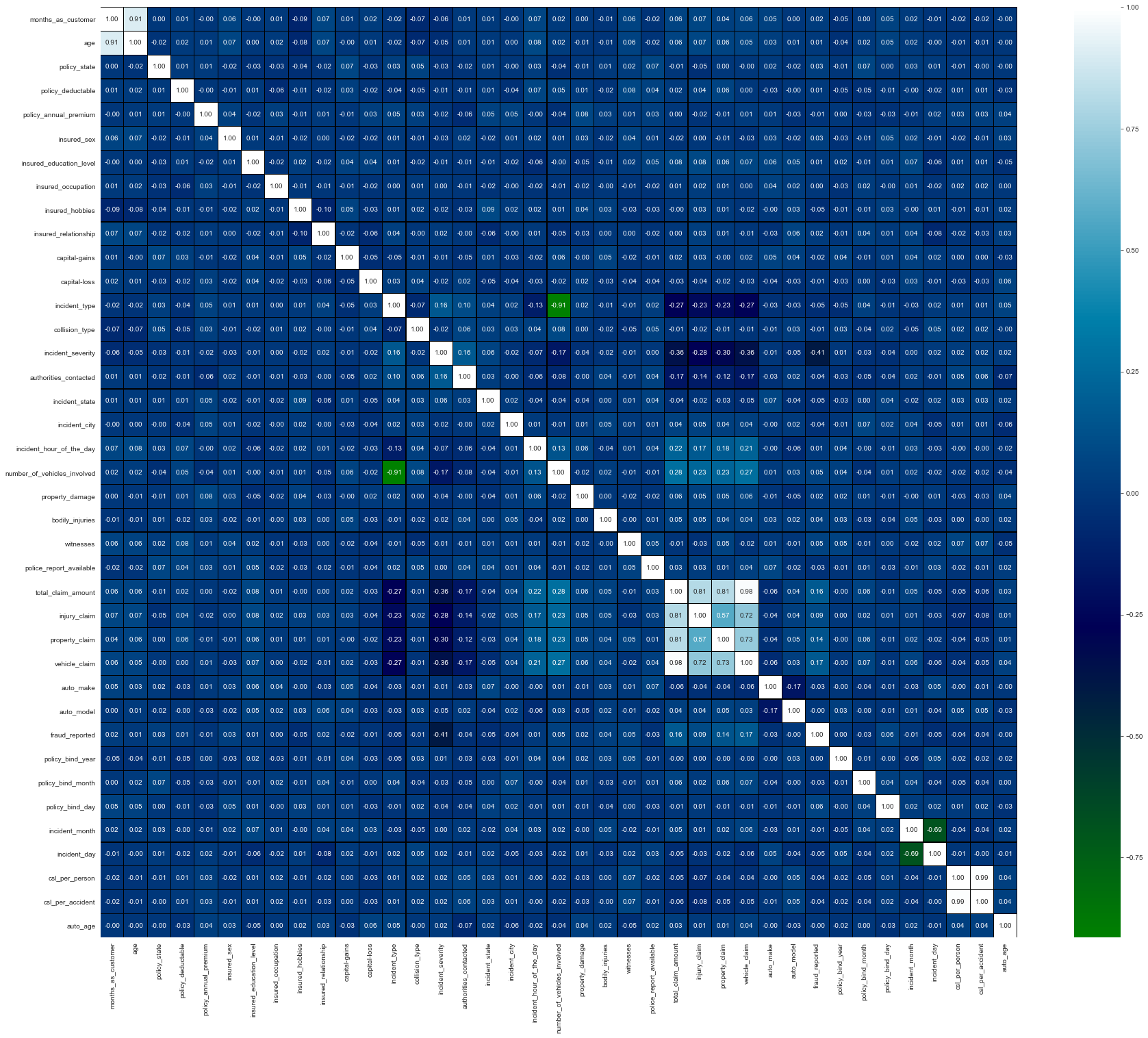
Now,We used the power transformation method ( yeo -johnson method ) to remove the skewness in the dataset. After using it, the skewness has almost been reduced.



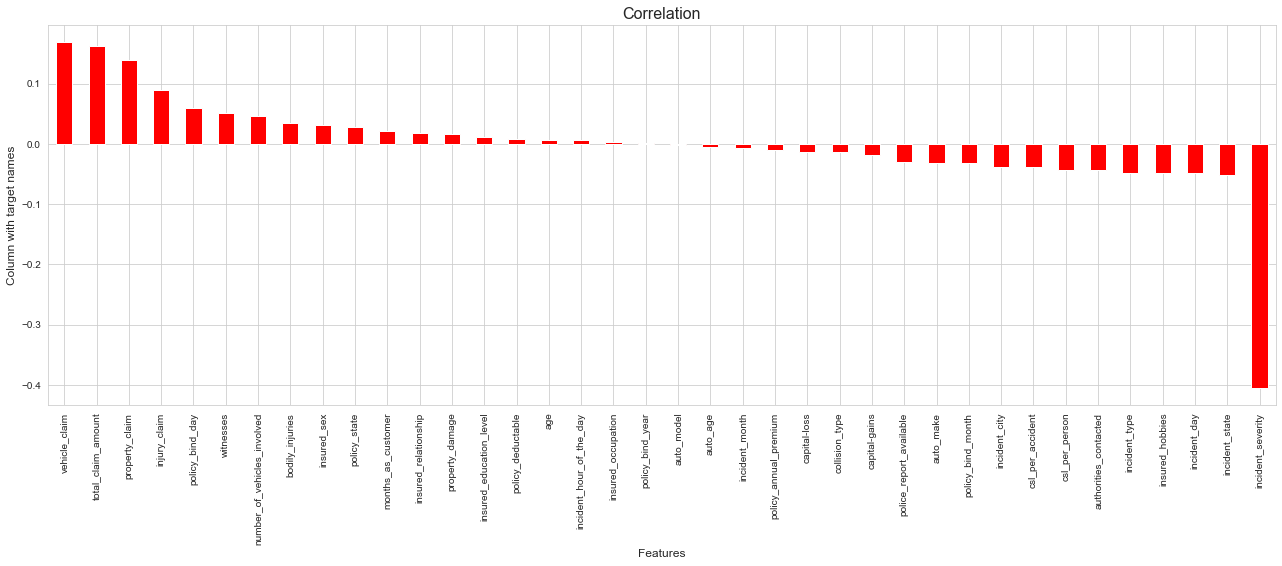
**CORRELATION MAP**

This Heatmap shows the correlation matrix by visualizing the data. we can observe the relation between one feature to another. This heat map contains both positive and negative correlation.

****



From the above correlation map, we see that there is very less correlation between the target variables and the other variables. We can observe that most of the columns are highly correlated with each other which result to the multicollinearity problem. We will check the VIF value to overcome this multicollinearity issue.

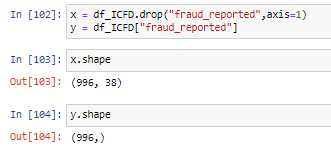


To get the better insights from the heat map we have used bar plots to show the positive and negative correlation between the target variable and other columns.

* Policy \_ bind \_year & auto \_model are the least correlated with target column.
* Next, insured \_occupation is slightly correlated with the target variable. Auto \_age and incident \_hour \_of \_the \_day are also less correlated with target.
* Vehicle \_claim, total \_claim \_amount & property \_claim are highly positively correlated with the target.
* Incident \_severity , is highly negatively correlated with the target.

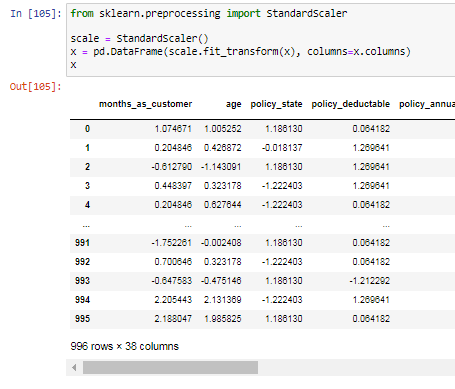
**PRE-PROCESSING PIPELINE**

First of all , We have to separate the target variable “fraud \_reported” and features to process the dataset for model building.



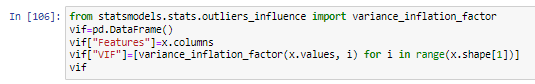
I have separated independent and dependent features and stored them in x and y respectively.

Now, we have to scale the data containing independent variables (x) in order to overcome the data biasness. Since I have removed the skewness and outliers and my data is also normal so I can use the Standard Scaler method to scale the data. If it is not the case then we could apply Min Max Scaler.

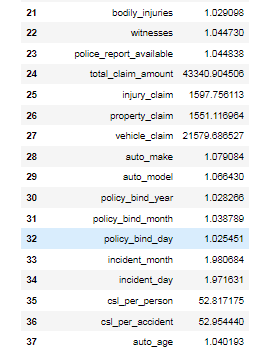
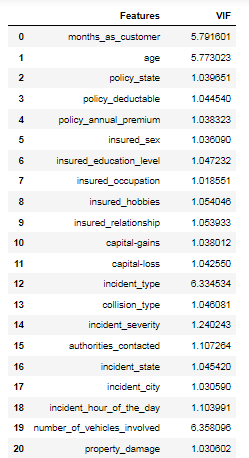


I have scaled the data using the standard scaler method to overcome the issue of data biasness.

In the heat map we found some features having high correlation with each other which means that there is a multicollinearity problem, so let's check the VIF values to solve the multicollinearity problem.



We got the VIF values as :



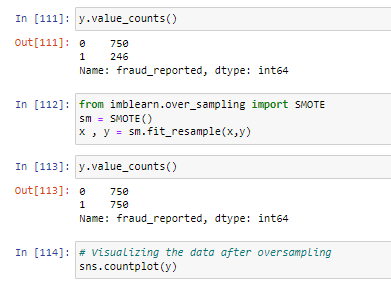
We see very high VIF values in “total \_claim \_amount”, “injury \_claim”, “property \_claim”, “vehicle \_claim”, “policy \_bind \_year” & “csl \_per \_person”.

The acceptable range of VIF is below 10. We observed the highest VIF in total \_claim\_ amount, so we dropped this column first and again checked the VIF to confirm whether the multicollinearity issue was solved or not. Again, we found a high VIF in the csl \_per \_accident column. So, we dropped that column too. After removing 2 columns our multicollinearity got solved by giving VIF values below 10 in all the columns.

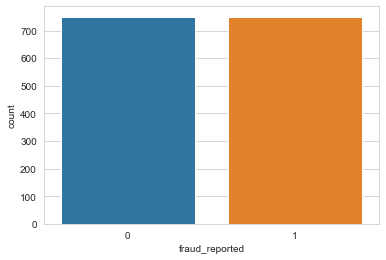
**OVER SAMPLING :**

Now, since we have come across the data imbalance issue, we need to fix it by either oversampling or undersampling the data. Oversampling is preferred, because undersampling causes a huge data loss.

Oversampling was done as follows:



The data is now balanced that we can observe in the count plot

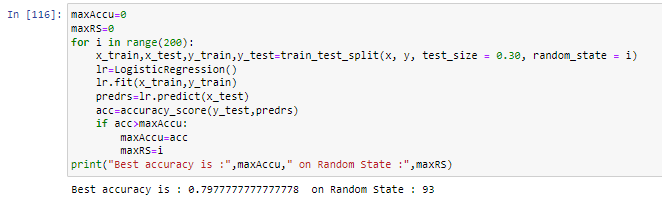
.

**BUILDING MACHINE LEARNING MODELS**

Since all the pre-processing and data cleaning is done, now our data is ready for model building process. Let’s get the predictions by creating some classification algorithms.

Before building the models, we first need to find the best random state and accuracy using any one of the classification models.

**FINDING THE BEST RANDOM STATE & ACCURACY**



* We have got the best random state as 93 and best accuracy as 79.77% using the Logistic Regression model. Now let’s create new train sets and test sets and fit them into the models to find our ideal model.

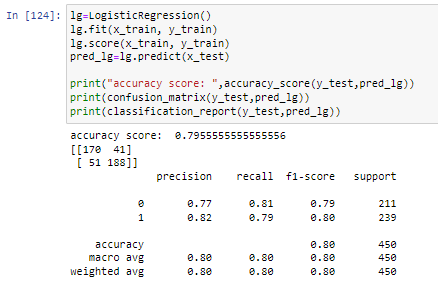


CLASSIFICATION ALGORITHMS :

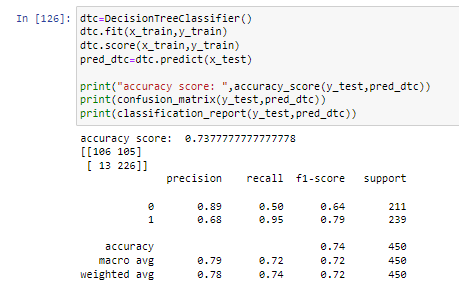
We have used 9 different classification algorithms for our predictions, they are : Logistic Regression Model, Decision Tree Classifier, Gaussian NB Classifier, Gradient Boosting Classifier, K-Nearest Neighbors Classifier, SVC Model, Random Forest Classifier, XG Boost Classifier & Extra Trees Classifier.

We have used evaluation metrics like classification report, confusion matrix, roc score and accuracy score. And we also used a cross validation score (cvs) to get the difference from the model accuracy for better result.

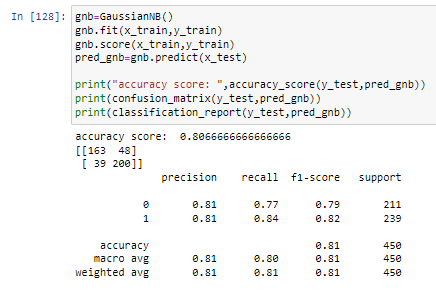
* LOGISTIC REGRESSION MODEL :



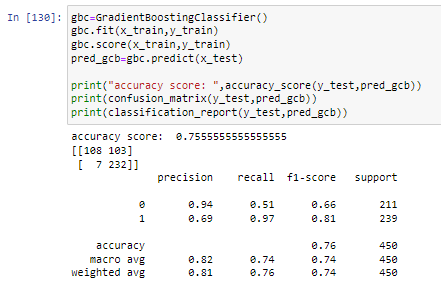
* The Logistic Regression Model gave us an accuracy score of 79.55 %.
* **DECISION TREE CLASSIFIER :**



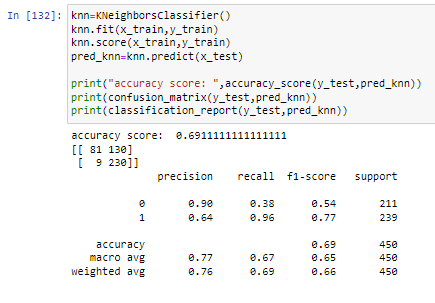
* The Decision Tree Classifier Model gave us an accuracy score of 73.77 %.
* **GAUSSIAN NB CLASSIFIER :**



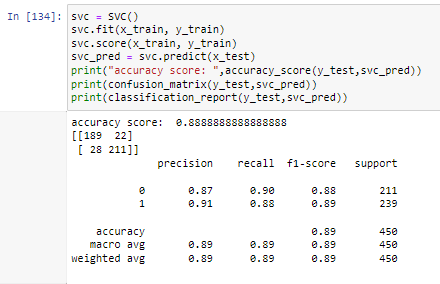
* The Gaussian NB Classifier Model gave us an accuracy score of 80.66 %.
* **GRADIENT BOOSTING CLASSIFIER :**



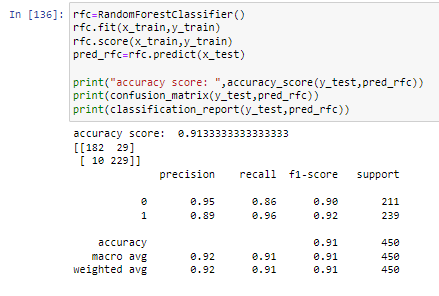
* The Gradient Boosting Classifier Model gave us an accuracy score of 75.55 %.
* K NEAREST NEIGHBORS CLASSIFIER :



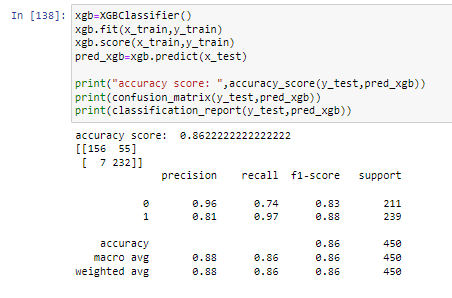
* The K Nearest Neighbors Classifier Model gave us an accuracy score of 69.11 %.
* **SVC(support vector classifier) MODEL:**



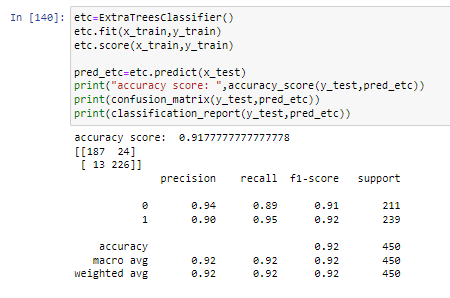
* The SVC Model gave us an accuracy score of 88.88 %.
* **RANDOM FOREST CLASSIFIER :**



* The Random Forest Classifier Model gave us an accuracy score of 91.33 %.
* **XGBOOST CLASSIFIER :**



* The XG Boost Classifier Model gave us an accuracy score of 86.22 %.
* **EXTRA TREES CLASSIFIER :**



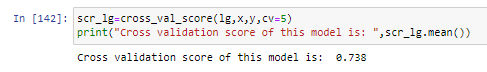
* The Extra Trees Classifier gave us an accuracy score of 91.77 %.
* From the above Classification Models, the highest accuracy score belongs to Extra Trees Classifier followed by Random Forest Classifier, then by SVC model & XG Boost Classifier.

,And then Gaussian NB Classifier, Logistic Regression Model, Decision Tree Classifier and Gradient Boosting Classifier.

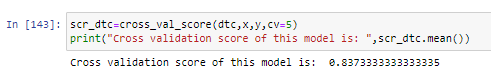
The lowest Accuracy score belongs to K Nearest Neighbors Classifier.

**CROSS VALIDATION SCORES:**

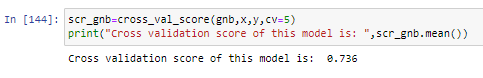
* We now checked the cross validation score of each of the models mentioned above.



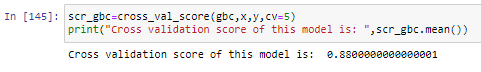
* The Cross Validation Score of the Logistic Regression Model is 73.8 %.



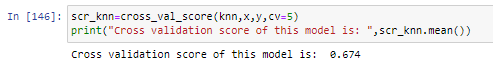
* The Cross Validation Score of the Decision Tree Classifier Model is 83.73 %.



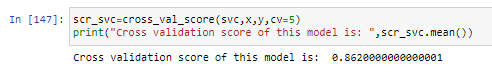
* The Cross Validation Score of the GaussianNB Classifier Model is 73.6 %.



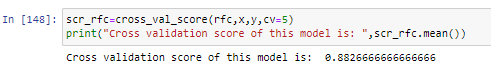
* The Cross Validation Score of the Gradient Boosting Classifier Model is 88.00 %.



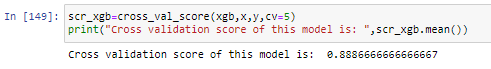
* The Cross Validation Score of the K Nearest Neighbors Classifier Model is 67.4 %.



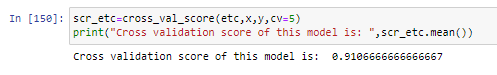
* The Cross Validation Score of the SVC Model is 86.20 %.



* The Cross Validation Score of the Random Forest Classifier Model is 88.26 %.



* The Cross Validation Score of the XG Boost Classifier Model is 88.86 %.



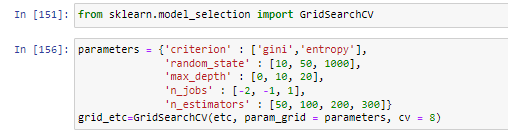
* The Cross Validation Score of the Extra Trees Classifier Model is 91.06 %.
* The highest Cross validation Score belongs to Extra Trees Classifier, followed by XG Boost Classifier, Random Forest Classifier, Gradient Boosting Classifier & SVC( support vector classifier) model.

Followed by Decision Tree Classifier, Logistic Regression model, Gaussian NB Classifier . and lastly, K Nearest Neighbors Classifier.

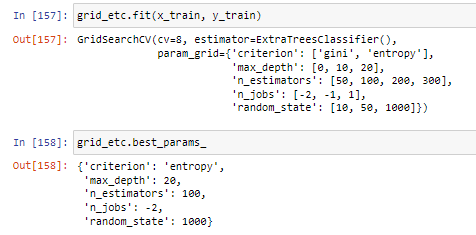
* HYPER PARAMETER TUNING :

Since the Cross Validation Score and the Accuracy Score of Extra Trees Classifier are both high, we shall consider this model for hyper parameter tuning.

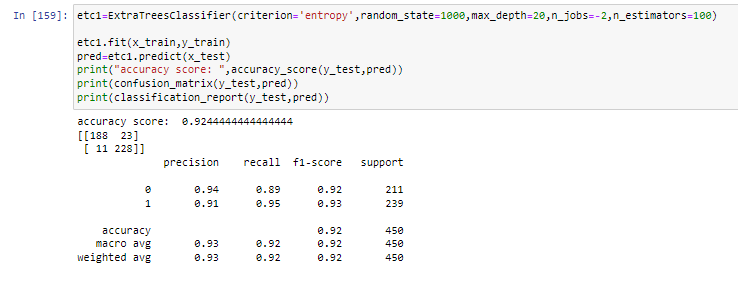
* We will use Grid Search CV for hyper parameter tuning.



* By using the above parameters, we are tuning the best model (Extra Trees Classifier) and after tuning we have to choose the best parameters from the above list.

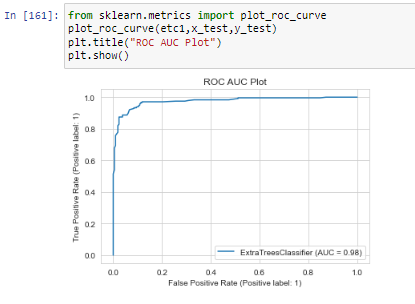


* These were found to be the best parameters after tuning, now let us use these parameters to improve our model.



The model after hyper parameter tuning has an improved accuracy score of 92.44 %.

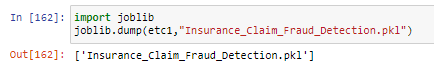
Now we will plot the ROC curve and compare the AUC for the best model.



We have plotted the ROC-AUC curve, AUC score is 98 %.

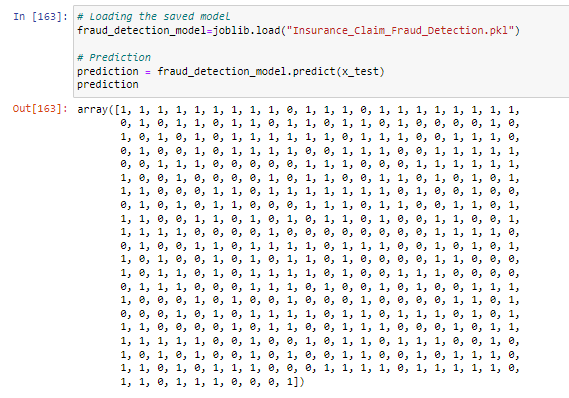
* **SAVING THE MODEL :**

Finally, we saved the model by using library “joblib”.

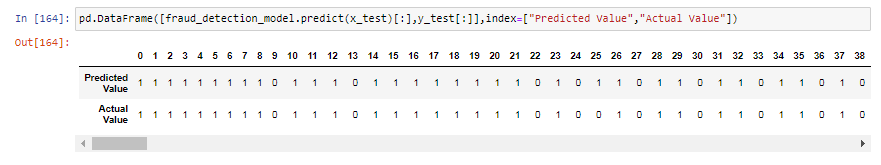


* **PREDICTION:**

By loading the saved model, we can now check predict value whether the insurance claim is fraudulent or not.



* Comparing the Predicted Values & the Actual Values.



* The above shows the predicted values and the actual values. The values are almost similar.

**CONCLUDING REMARKS**

In this project we went through the different processes involved in building a machine learning model. We started with Exploratory Data Analysis, did some Data Cleaning, conducted some Feature Extraction & Feature Engineering which were crucial in making our data ready for visualization and model building.

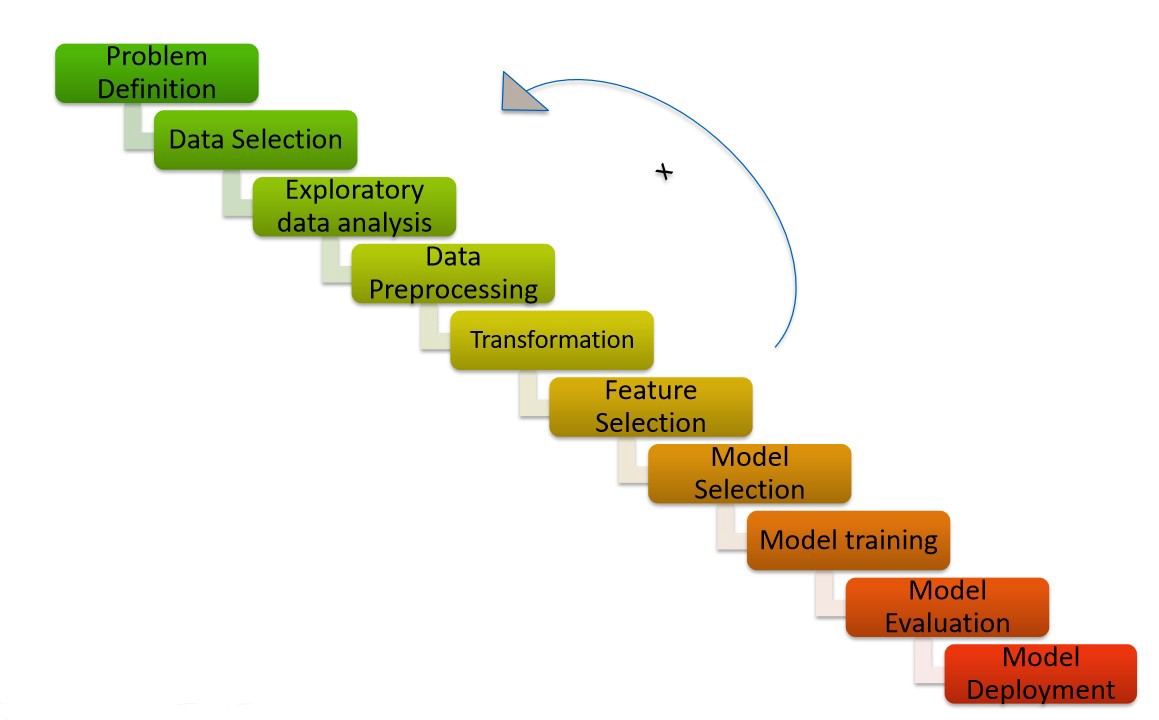
We did some Data Visualization using count plots, scatter plots, bar plots & dist plots. After visualization we encoded the data frame using Label Encoder. Next we checked for outliers present in the data and removed them using the z-score method. We checked the skewness of our data and reduced it for better model building.

And lastly, we built different classification models to predict whether the insurance claim is fraudulent or not and performed the hyper tuning to improve the best model by using different parameters.

With the help of above techniques, our model is able to predict the fraudulent report with the accuracy of 92.44%. Also, we have seen that the actual and predicted values are almost the same, which means our model worked correctly.

## **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.**

**Building machine learning models for such problems can help the insurance companies to choose the correct insurer. So, Machine learning techniques are very useful to solve these kinds of problems.** **This project has built a model that can detect auto insurance fraud. In doing so, the model can reduces loses for insurance companies. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims.**

****

**About the Author:**

**Pursuing Post Graduate Diploma in Data Science, Machine Learning and Neural Network from Datatrained . I am an aspiring Data Scientist whose purpose is to learn in detail all the concepts needed for Data Science. I am passionate about Data Science and have skills that help me derive valuable insights from data, such as Data Manipulation, Data Visualization, Data Analysis, EDA, and Machine Learning.**

**LINKEDIN: https://www.linkedin.com/in/abhishek-kumar-a68b06131**

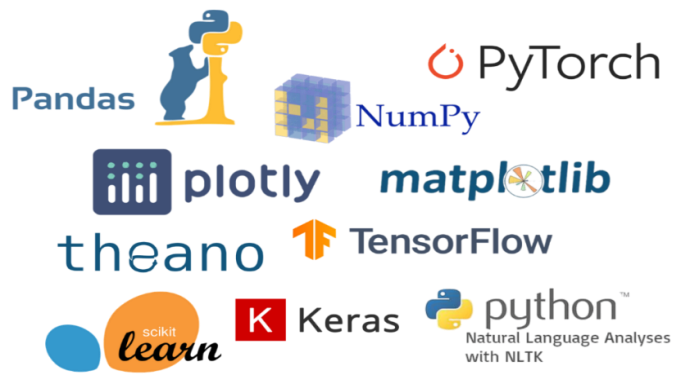
# Hardware & Software Requirements & Tools Used:

### Hardware required:

* Processor: core i5 or above
* RAM: 8 GB or above
* ROM/SSD: 250 GB or above

**Software requirement**:

* Jupiter Notebook

**Libraries Used**:

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