

Shilpi Mohanty | Blog Submission | April 27, 2022

INSURANCE CLAIM FRAUD DETECTION

AUTOMOBILE INDUSTRY

## **1.INTRODUCTION**

According to the Insurance Information Institute (III), insurance fraud is a deception committed against an insurance company for financial gain. It can be anything from lying about a garaging address (the location where your vehicle is parked for most of the year) to exaggerating an accident to outright staging one.

What is Insurance Fraud?

Insurance fraud is an attempt to exploit an insurance contract. Insurance is meant to protect against risks, not to serve as a vehicle to enrich the insured. Although insurance fraud by the policy issuer does occur, the majority of cases have to do with the policyholder attempting to receive more money by exaggerating a claim.

## **Types of Insurance Claims Fraud**

Insurance claims fraud is a grave problem and its increasing at an alarming rate. Some classic examples of fraud to crop up during the claims settlement process in insurance are given below:

1. Faking an accident, injury, or damage to vehicle

2. Filing a claim against an incident that did not take place

3. Filing a claim by submission of incorrect details

4.Intentionally causing damage to vehicle

5. Filing false police reports for raising claims against fake incidents

6. Inflating the damage and filing additional claims which didn’t occur.

The blog I am presenting here is based on the insurance claim fraud detection where the goal is to find out the claims made by the insured is fraudulent or genuine. Insurance company must be very careful in analyzing every claim settlement process and it’s very difficult to manually analyse and examine each and every single claim as it takes lot of time and incur costs too. But with the help of machine learning models, we can easily verify that the claim is fraudulent or not within short duration of time .Thus insurance company can take extra precautionary measures and denied the fraudulent claim. Let’s first outline the process involved in the building of model for insurance claim fraud detection. Flow diagram of the process as below:

**2.Problem Definition**

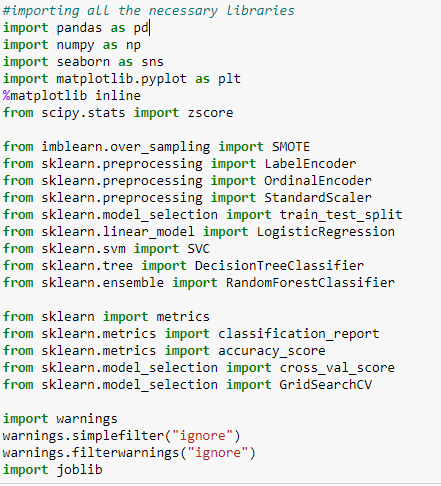
Business\_case:   
Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem.

In this project, 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.

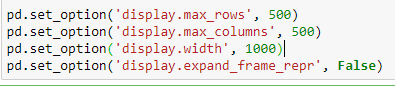
In this example, we will be working with some auto insurance data to demonstrate how we can create a predictive model that predicts if an insurance claim is fraudulent or not.

**Importing of libraries**

First I imported the required libraries that are needed for project and obtain the rest as and when required.



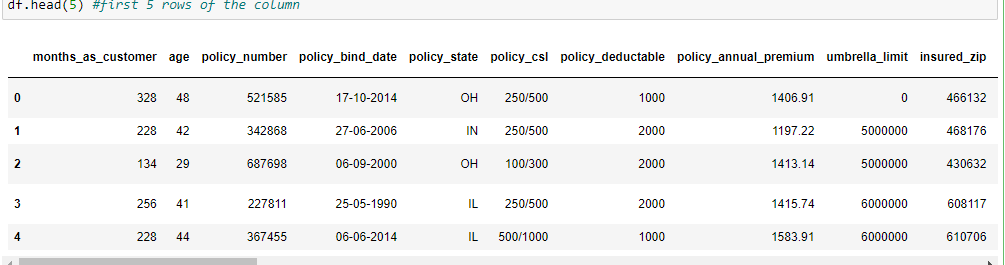
This is for displaying maximum number of row, columns that pandas will display while displaying a data frame. Also it align width and expand the frame.



Then I loaded the dataset which was in csv format in jupyter notebook by creating an instance ‘df’.The short representation is as below:



3. **DATA ANALYSIS :**

**Data analysis** is the practice of working with data to glean useful information, which can then be used to make informed decisions.Thus with the help of head(5) function ,it shows me first five rows of each column.****

Initial Observations:

1.We can see Null values are there which need to be imputed with mode or mean or dropping od column as case to case basis .

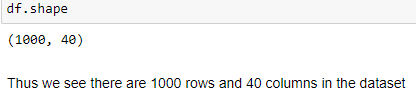
2.There are some question marks in the columns which has to be replace with nan then suitable imputation technique.

**3.fraud\_reported column is our target column which is in binary form so this makes the case study as classification problem.**

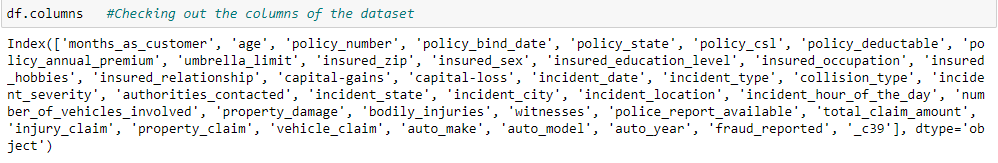
**4.Exploratory Data Analysis (EDA)**

It is the process to know about your dataset in depth and detail. This is called Exploratory Data Analysis. We know about data different characteristics with useful visual patterns and helps in in-depth analysis

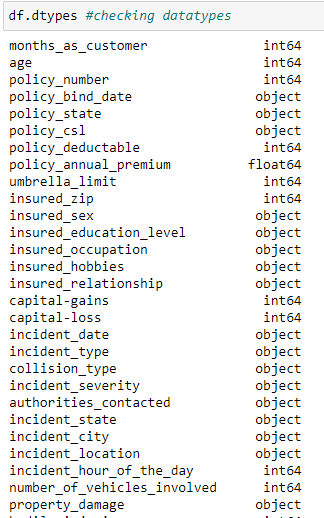
With the help of python code, I carry out initial analysis first is dataframe ‘shape’.The dataframe contains 1000 rows and 40 columns.



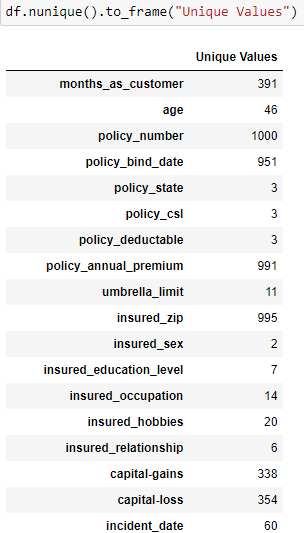
Then I checked for columns’ name, datatypes and unique values. Below are short snap view:



This shows names of all the columns present in the dataset.



Thus we see there are 2 float datatype,17 int datatype and 21 object datatype in the given dataset. The object datatype has to be converted into numerical datatype before building a model.

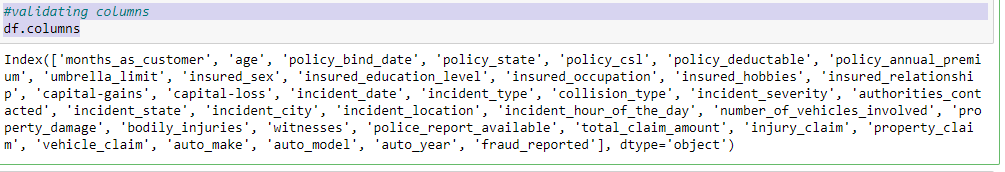


Here we see the number of distinct values present in each column.

Thus from above,we see policy\_number is an identifier and insured\_zip is a pincode number which is irrelevant for model building purpose so dropping the same.

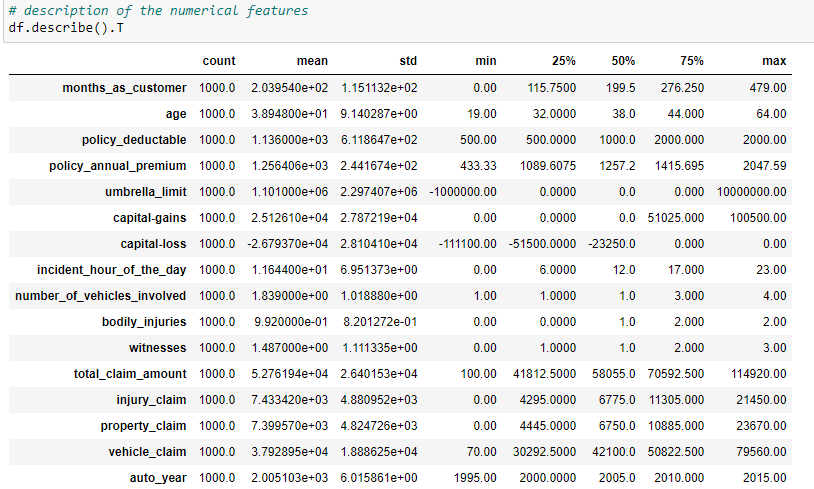


Dropping \_c39 columns as it consists of Nan Values.

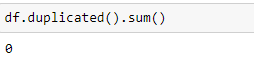


Validating column to see changes has been made.

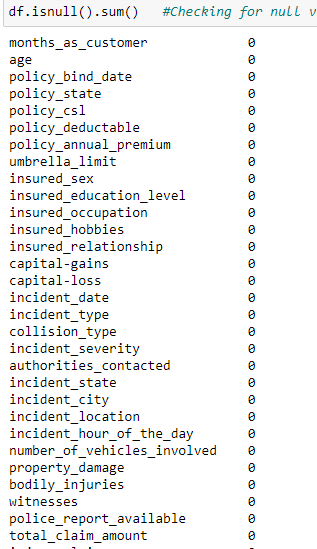
Describe function: The Describe function returns the statistical summary of the dataframe or series. This includes count, mean, median (or 50th percentile) standard variation, min-max, and percentile values of columns.

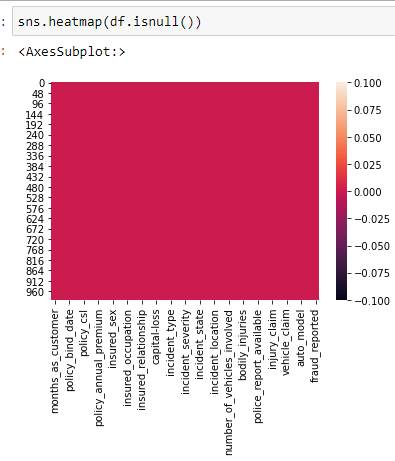


Duplicate function to see any duplicated values, thus there are no duplicate values in the dataset so it showing zero.



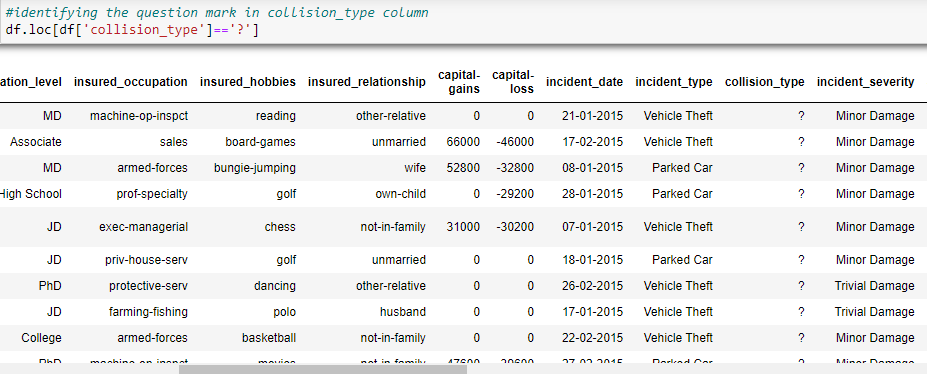
Missing value:

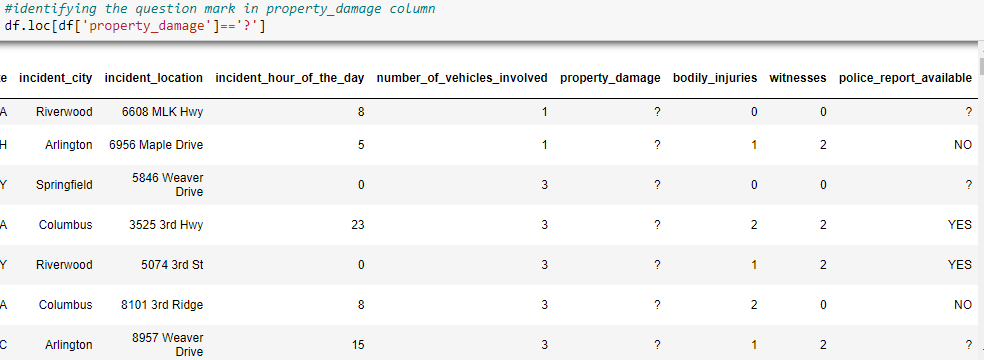




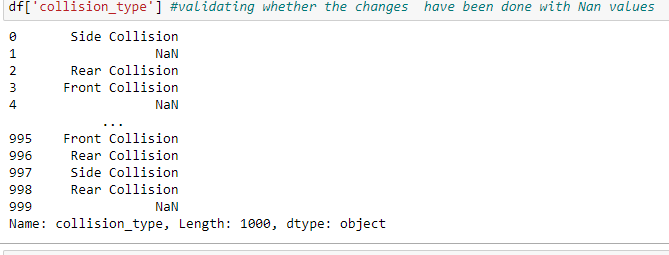
Thus there is no missing value in the dataset.

Data cleaning: we can see question marks ‘?’ in collision type, property\_damage,police\_report\_available. Thus, I imputed it with Nan values first, then all being object datatype so imputed it with mode imputation technique.

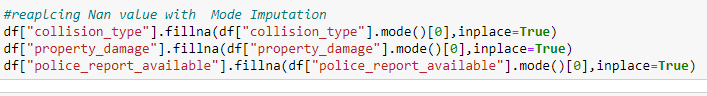




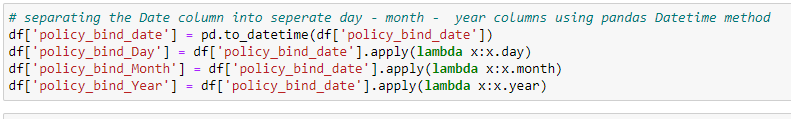


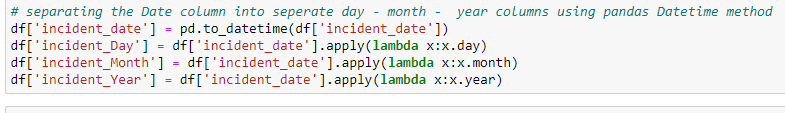


Replacing Nan values with mode imputation in collision\_type,property\_damage and police\_report\_available.

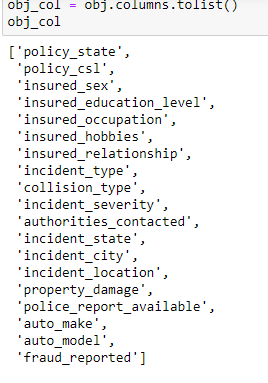


Creating new variable for policy\_bind for day,month,year and incident for day,month,year and deleting the date column and incident\_year column as it pertains to only one year 2015.





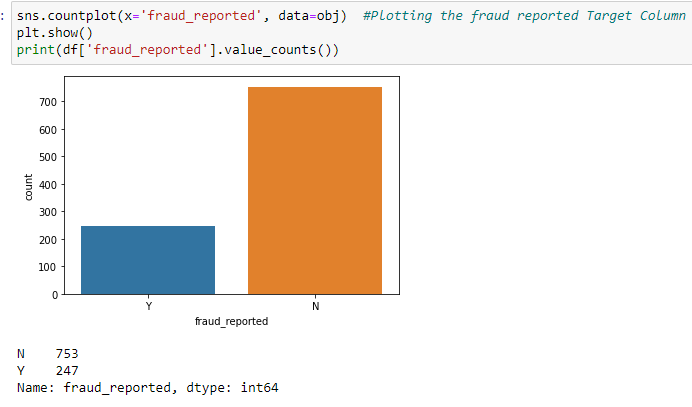




Thus we have separated the object datatype column names and numeric data type column names. We will be checking outliers and skewness only in continuous column not in categorical column.This makes our job easy.

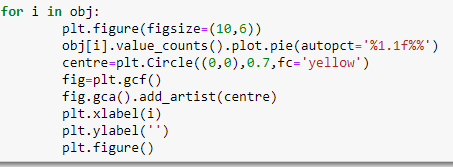
Data Visualization:

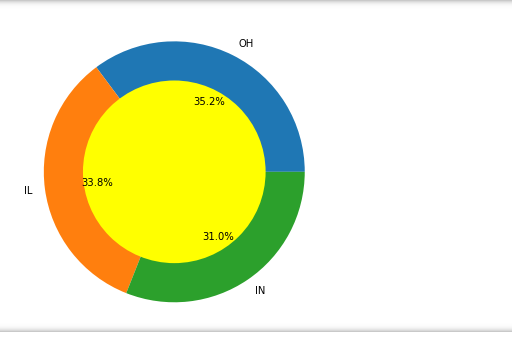
Target Column:

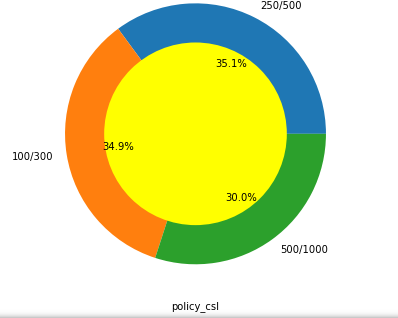
* 

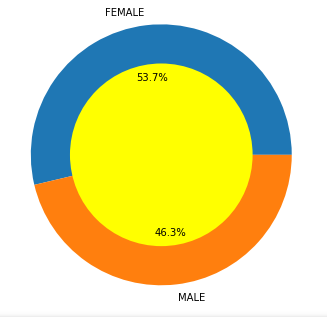
Thus we see target column is imbalanced and we should be applying smote technique to balance it. We observed that there are 753 cases not reported as fraud claims and 247 cases which are reported as fraud claims.

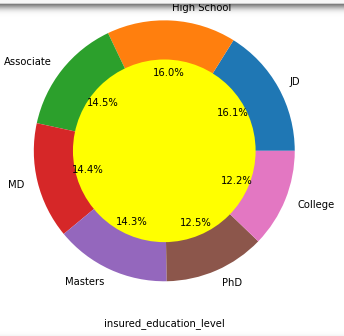
Categorical Column Analysis

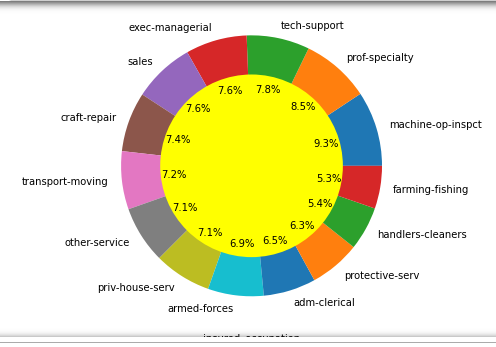


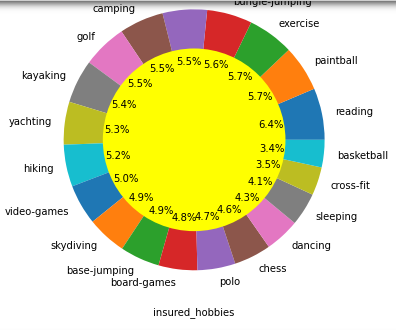


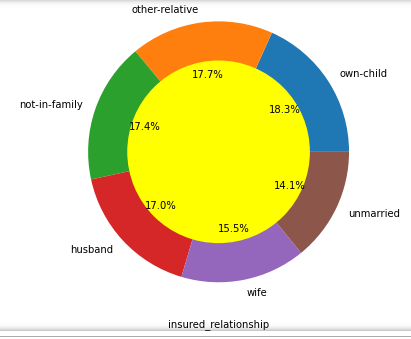


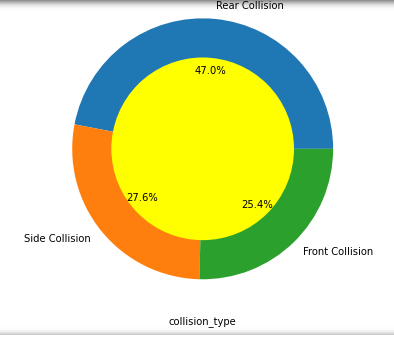


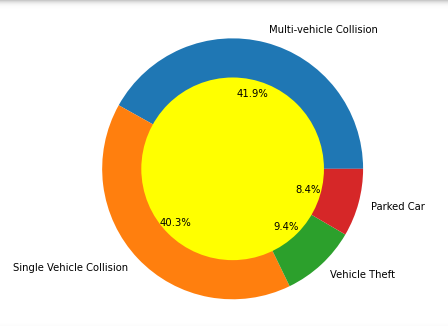


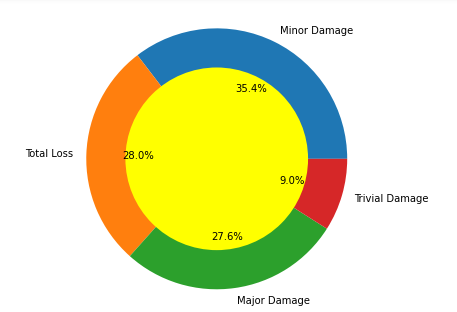


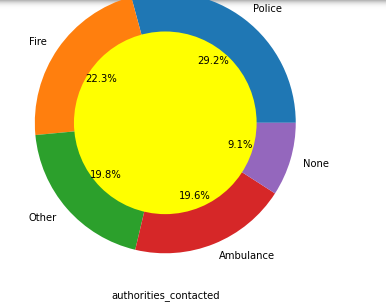


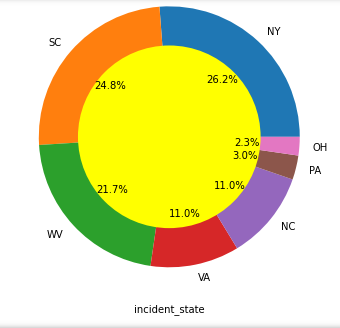


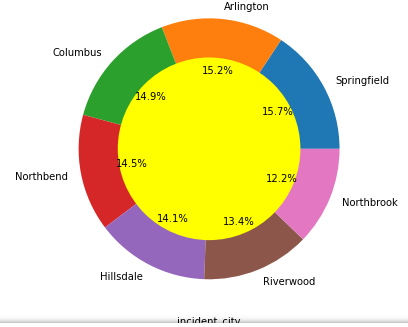


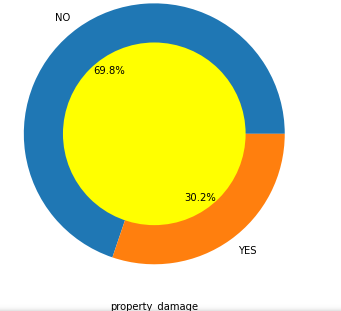


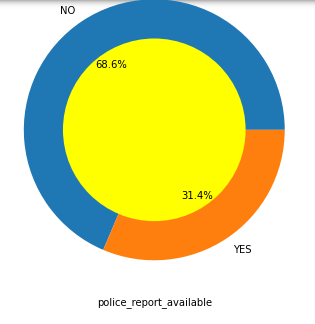


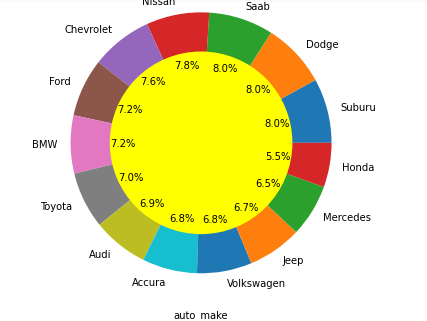


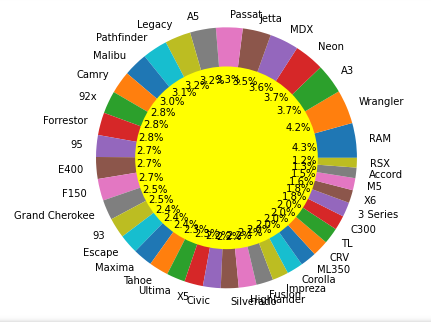








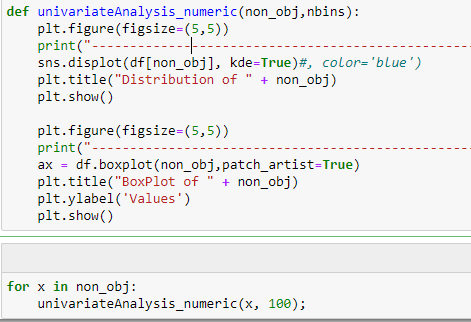




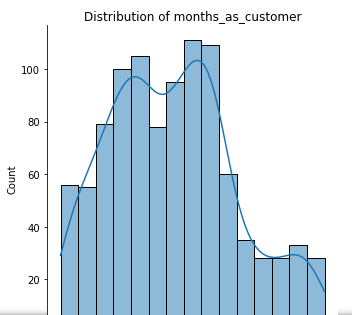
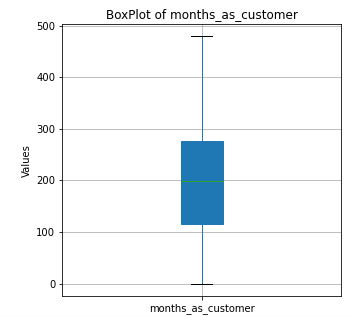
hus,we see that with the help of pie-plot,we get a quick snapshot of various columns showing percentages of each category in the column.

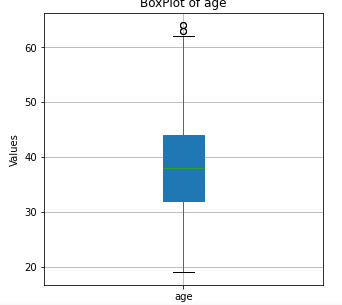
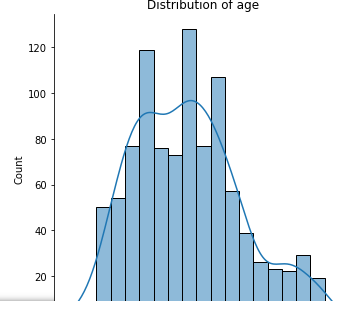
Non\_object Column Analysis

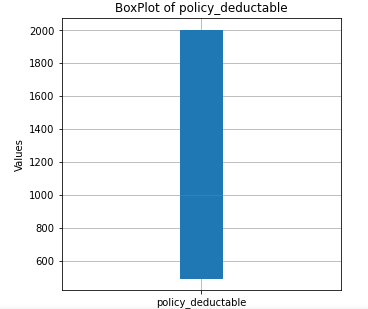
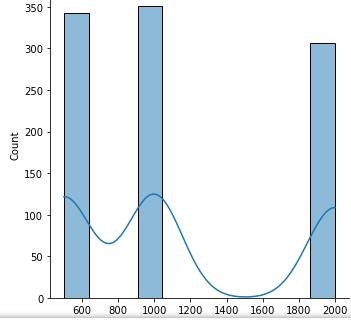
Then I created a function definition which will take all the non\_obj columns one by one for frequency distribution to see whether normalized or not and box\_plot for outliers checking and show the output.

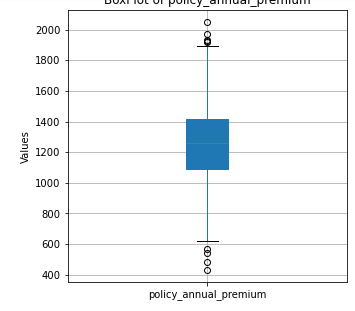
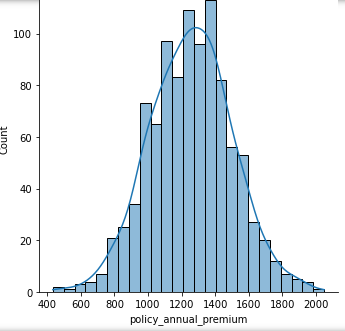


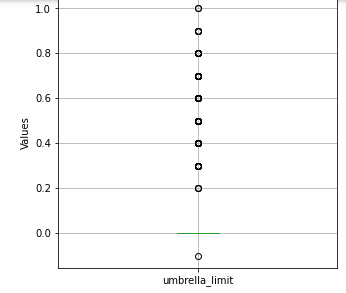
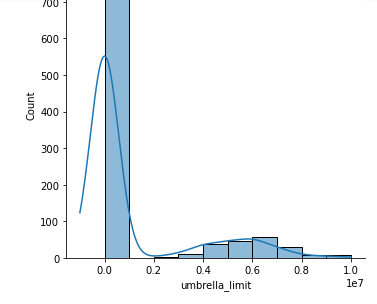
Output:

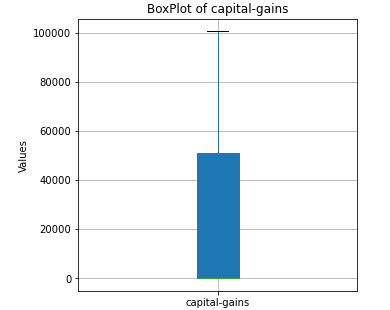
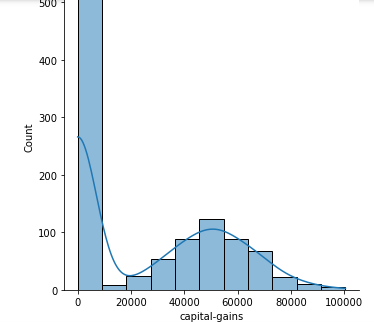
 

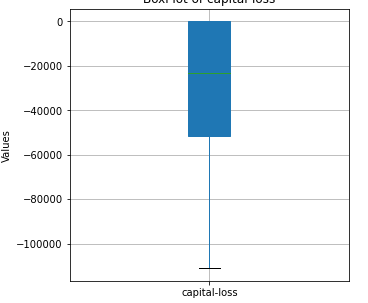
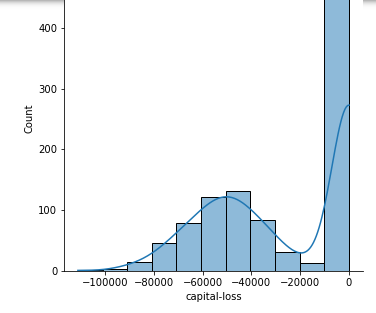


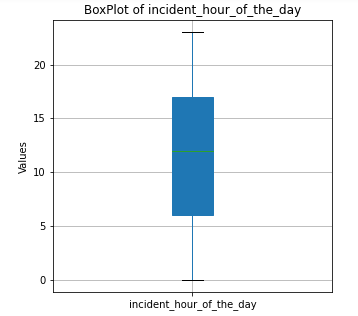
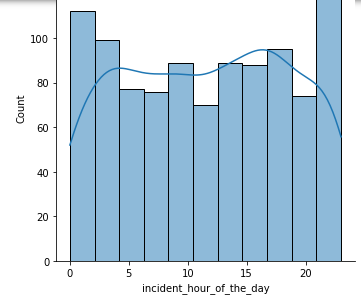


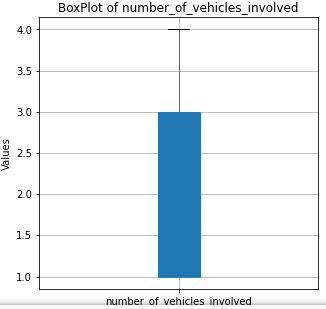
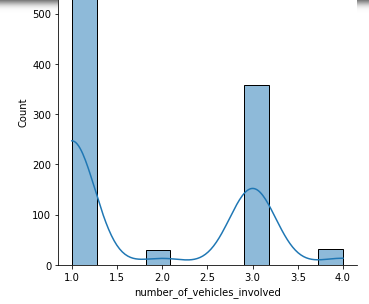


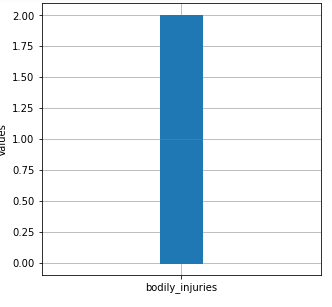
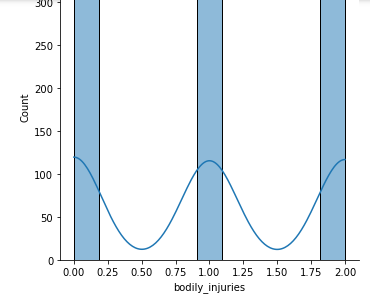


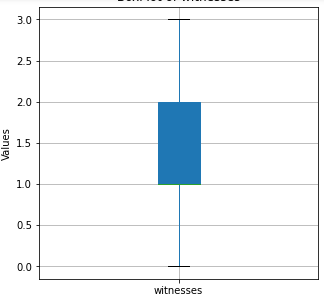
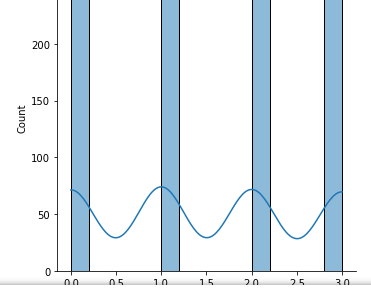


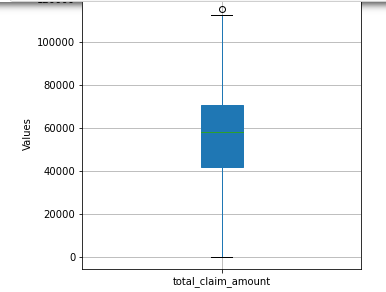
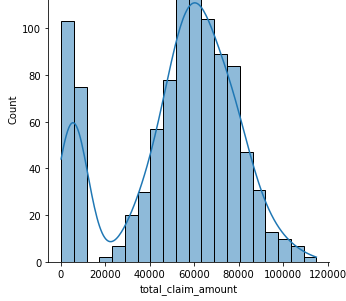


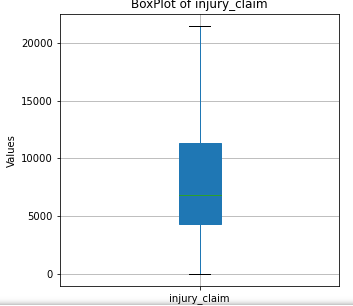
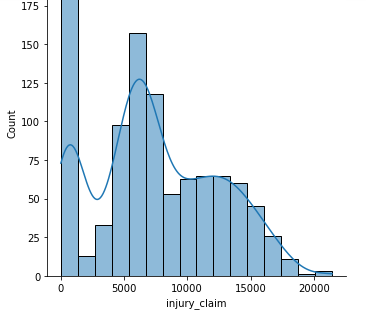


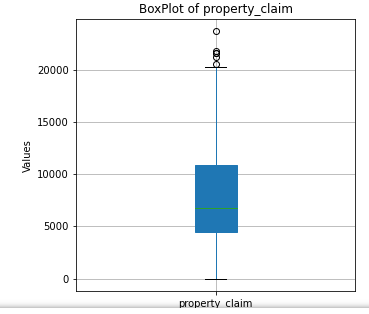
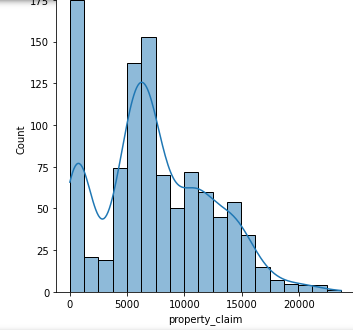


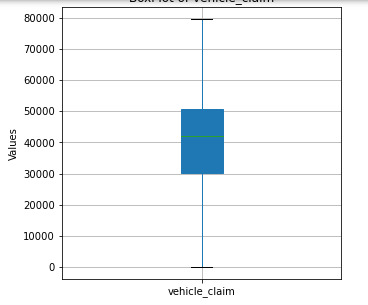
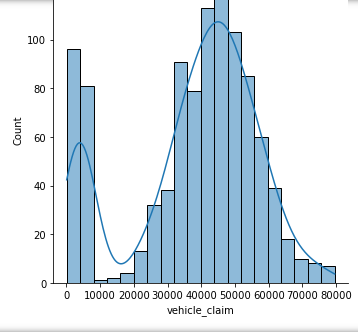


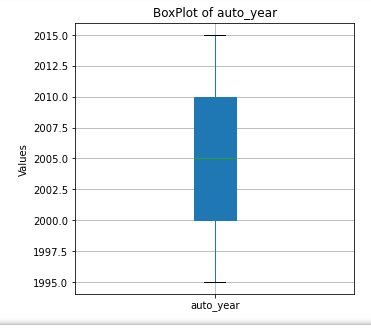
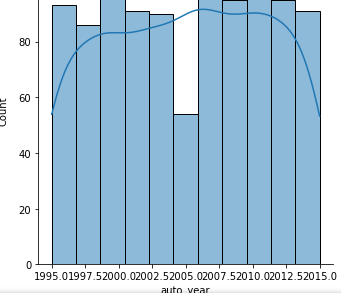


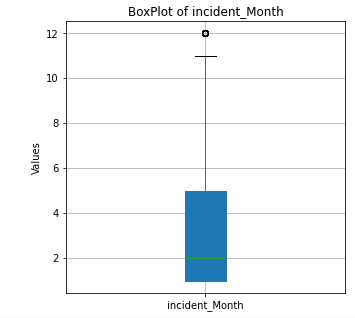
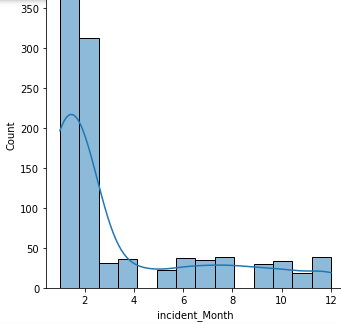






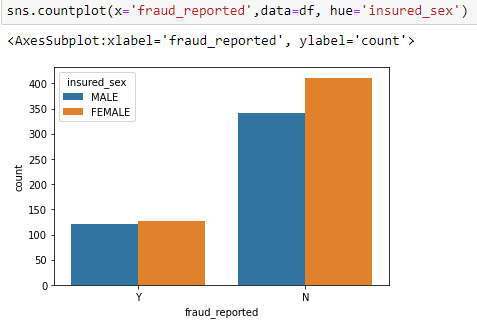




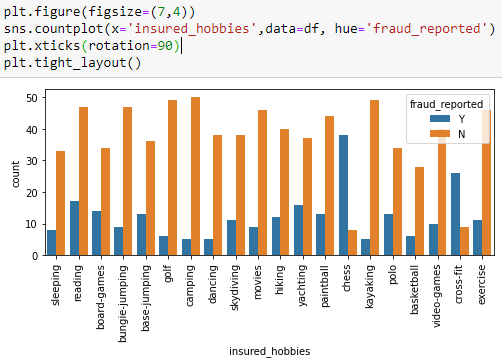


Thus we can see Outliers are present in policy\_ annual\_ premium, umbrella limit, total\_claim\_amount, property\_claim, incident\_month. Also most of the columns are not normally distributed.

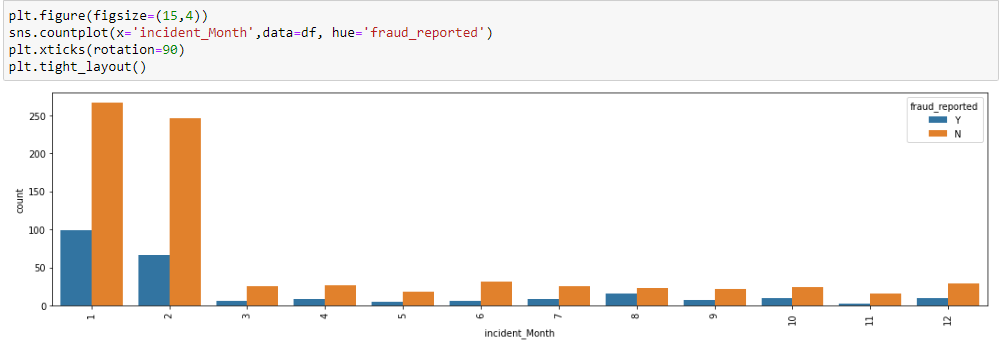
Now analyzing with reference to fraud\_reported as x column and hue denoted as different columns category:



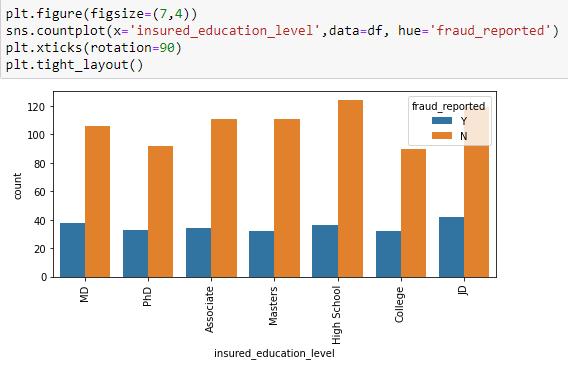
Thus we see that majority of the fraud claims are not reported in case of female



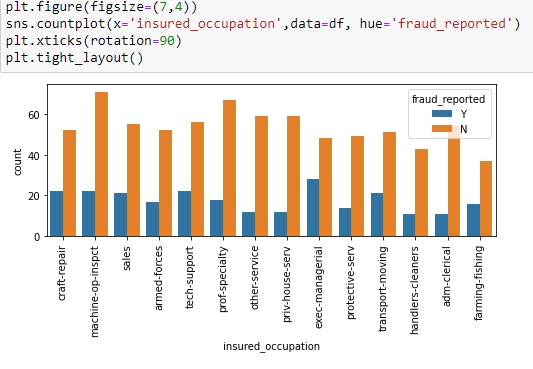
Those who are playing chess and cross-fit are having most number of frauds claim cases comparing to others.



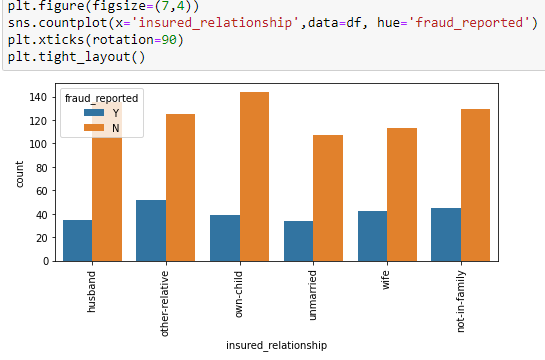
. Thus we see that fraud case was reported highest on the first and second month.



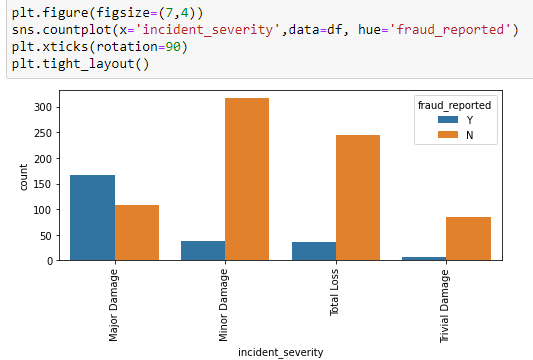
Thus we see that educational level is not impacted by fraud claim cases. It is almost equally distributed in fraud claim cases.



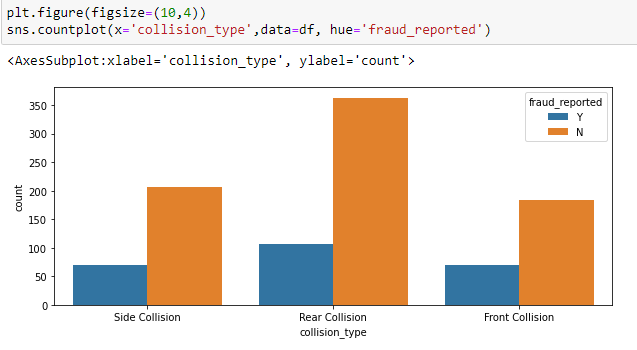
Thus we see those who are exec-managerial were having higher fraud claim cases compare to other occupation.



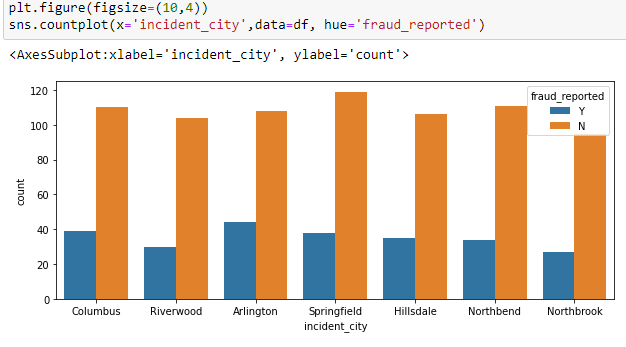
In relationship, those who are other\_relative,wife and not in family seems to have higher fraud claim cases compare to other relationship.



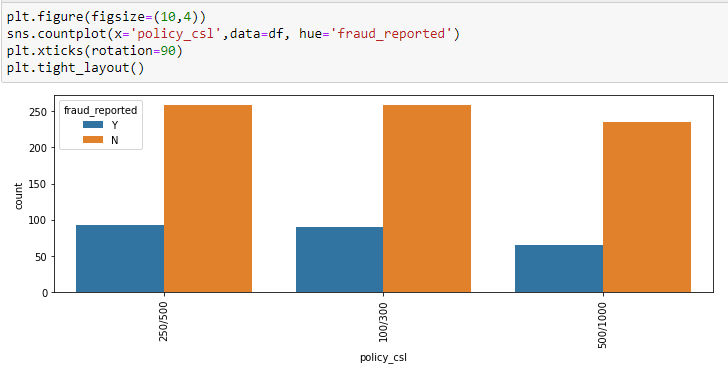
Thus we see that major damage reported to have highest fraud claim cases than non-fraud claims.



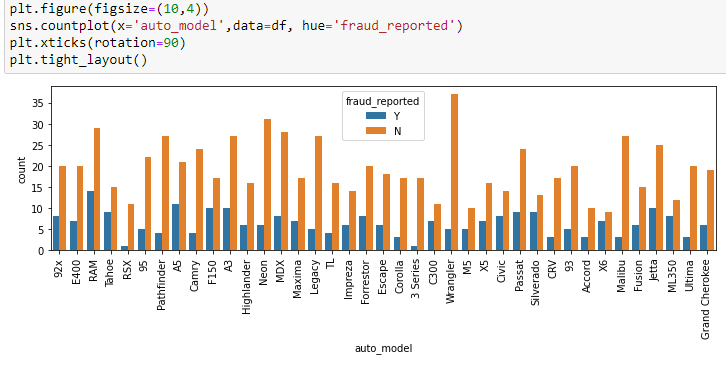
There seems to be slightly higher fraud claim cases in rear collision followed by side and front collision.



Thus we can see that almost same count of fraud claim cases reported in each city ,still Arlington and Columbus are slightly higher.



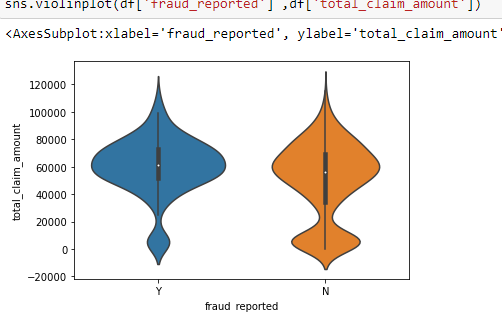
Thus we see that policy 250/500 and 100/300 are having slightly higher fraud claim cases and least is 500/1000.



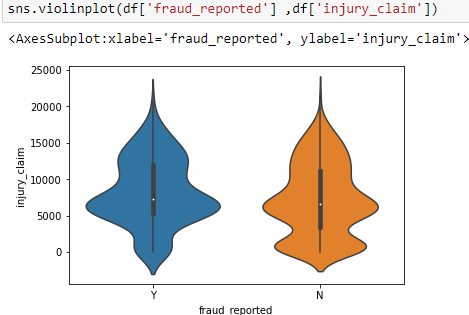
The model of the auto where the fraud claim case reported is higher in RAM,A5,F150,A3,Jetta,Silverado.

**Bivariate/Multivariate Analysis**

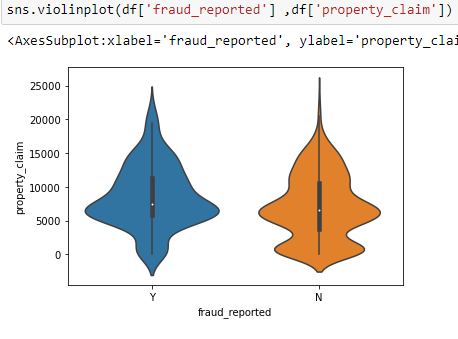
A Bivariate analysis is will measure the correlations between the two variables. Multivariate analysis is a more complex form of statistical analysis technique and used when there are more than two variables in the data set.



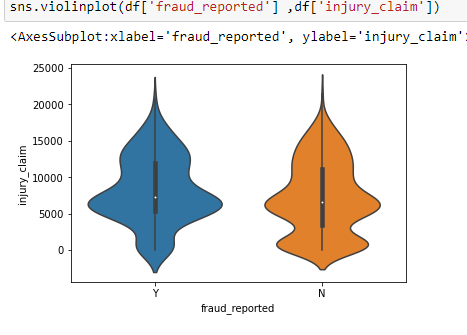
We obsreve that the number of frauds are more than non-fraud claims when the total claims were between 40k to 80k.



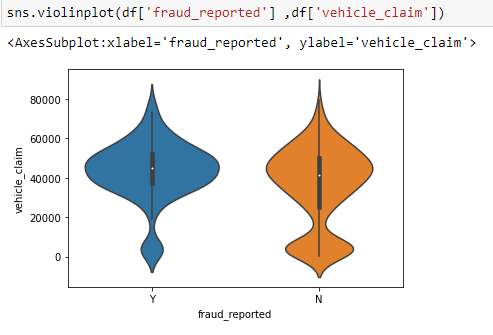
Thus we observe that there seem to be more frauds than non-fraud claims when the insurance claim is between 5k to 15k.



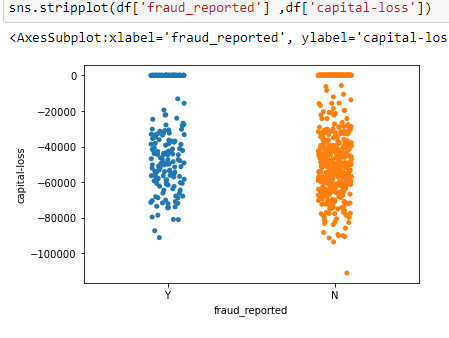
we observe that there seem to be more frauds than non-fraud claims when the property claim is between 5k to 15k.



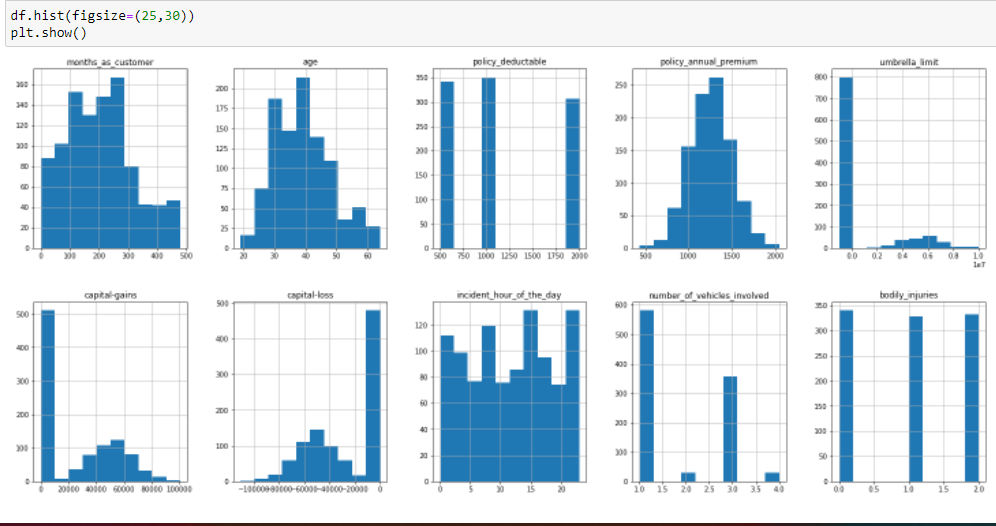
we observe that there seem to be more frauds than non-fraud claims when the injury claim is between 4.5k to 15k.

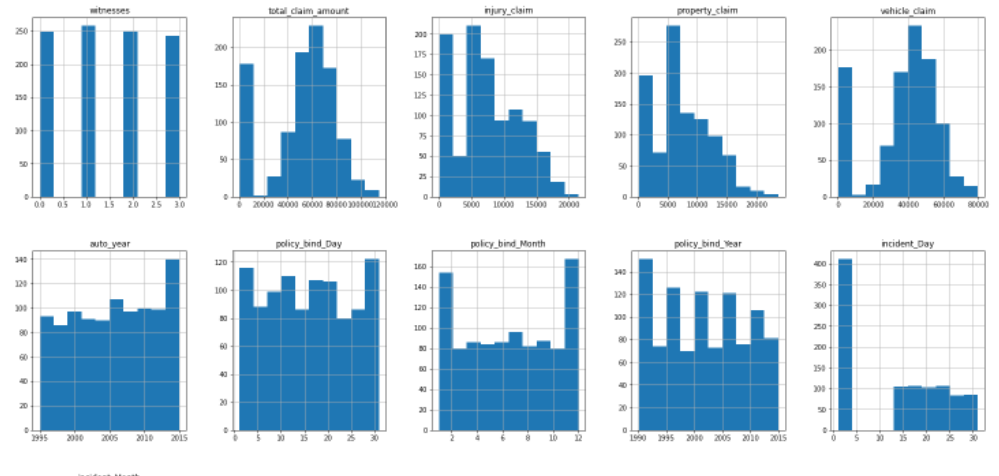


We observe that there seems to be more fraud cases than non fraud cases when the vehicle claim amount is between approximately 20k to 60k.



We saw that when there is fraud claim reported, then capital loss to the company is more as the company has to settle the claim amount demanded by the customer than the premium paid by the customer.Thus we need to be vigilant and examine properly the insurance claim cases.

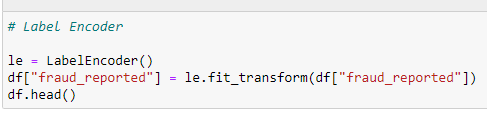


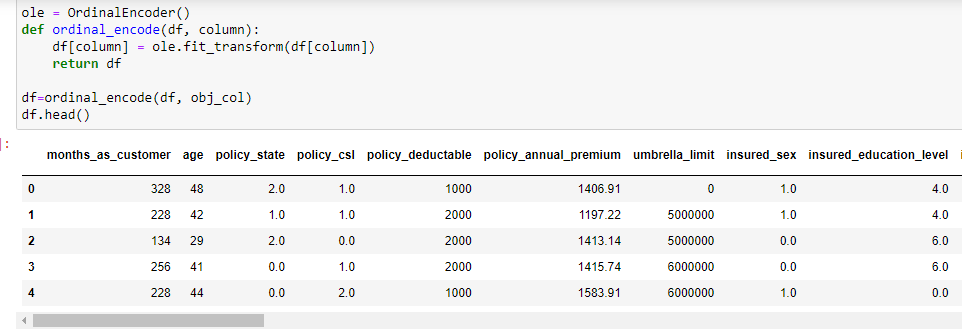


Thus we see that some of the columns are normally distributed like age, policy\_annual\_premium and rest are not normally distributed.

**5.Pre-processing Data**

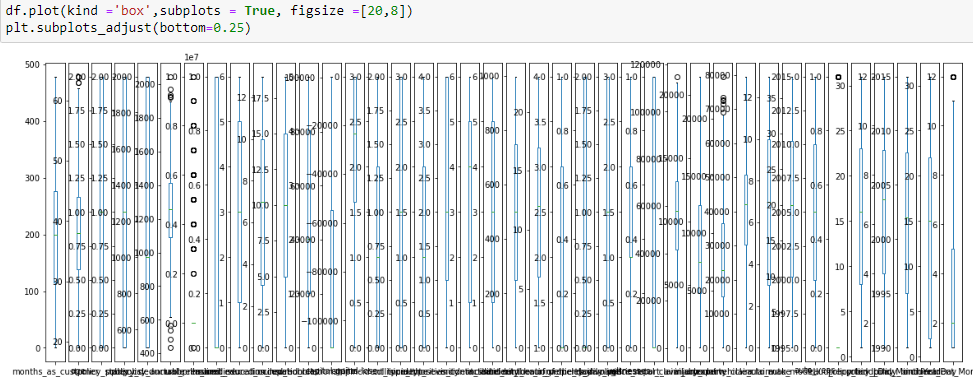
Data preprocessing is a very important step in machine learning which can yield accurate and insightful results. As machine learning can only understands 0s and 1s.So encoding is the very important required pre-processing step of machine learning which converts the categorical data to numerical data.Remember,better encoding always results in better model. Here I have used Label encoder for my target column as my target column fraud reported is in binary form Y(Yes) and N(No) which will easily convert into 1 for Y(Yes) and 0 for N(No). And for categorical column,I have used ordinal encoder in which each unique category value is assigned an integer value.

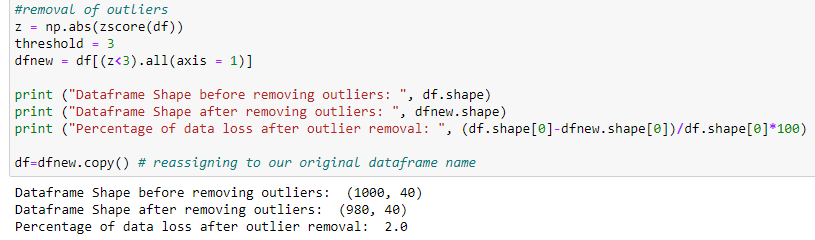




# **Checking Outliers(Outliers are the abnormal values)**

# We have already seen that there are outliers present in some of the features while ploting box plot.Thus as subplots ,we can see all the columns at single place .





Thus we see percentage loss in the data is only 2% which is acceptable

Pairrplot shows how each column is having relationship with each other. Thus short half view as below.



# **Correlation Analysis**

Correlation Analysis is statistical method that is used to discover if there is a relationship between two variables/datasets, and how strong that relationship may be.It is used to identify whether there is any significant connections, patterns, or trends between the two.

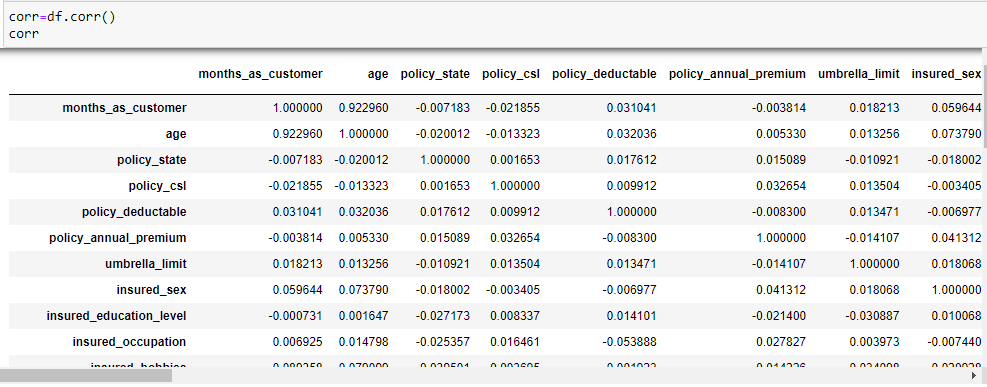
A positive correlation result means that both variables increase in relation to each other, while a negative correlation means that as one variable decreases, the other increases.

Positive Correlation Any score from +0.5 to +1 indicates a very strong positive correlation, which means that they both increase at the same time.

Negative Correlation Any score from -0.5 to -1 indicate a strong negative correlation, which means that as one variable increases, the other decreases proportionally.

No Correlation A score of 0 indicates that there is no correlation, or relationship, between the two variables.

Code and output:



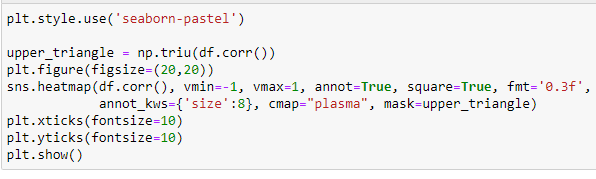
This shows correlation between independent and dependent variables.

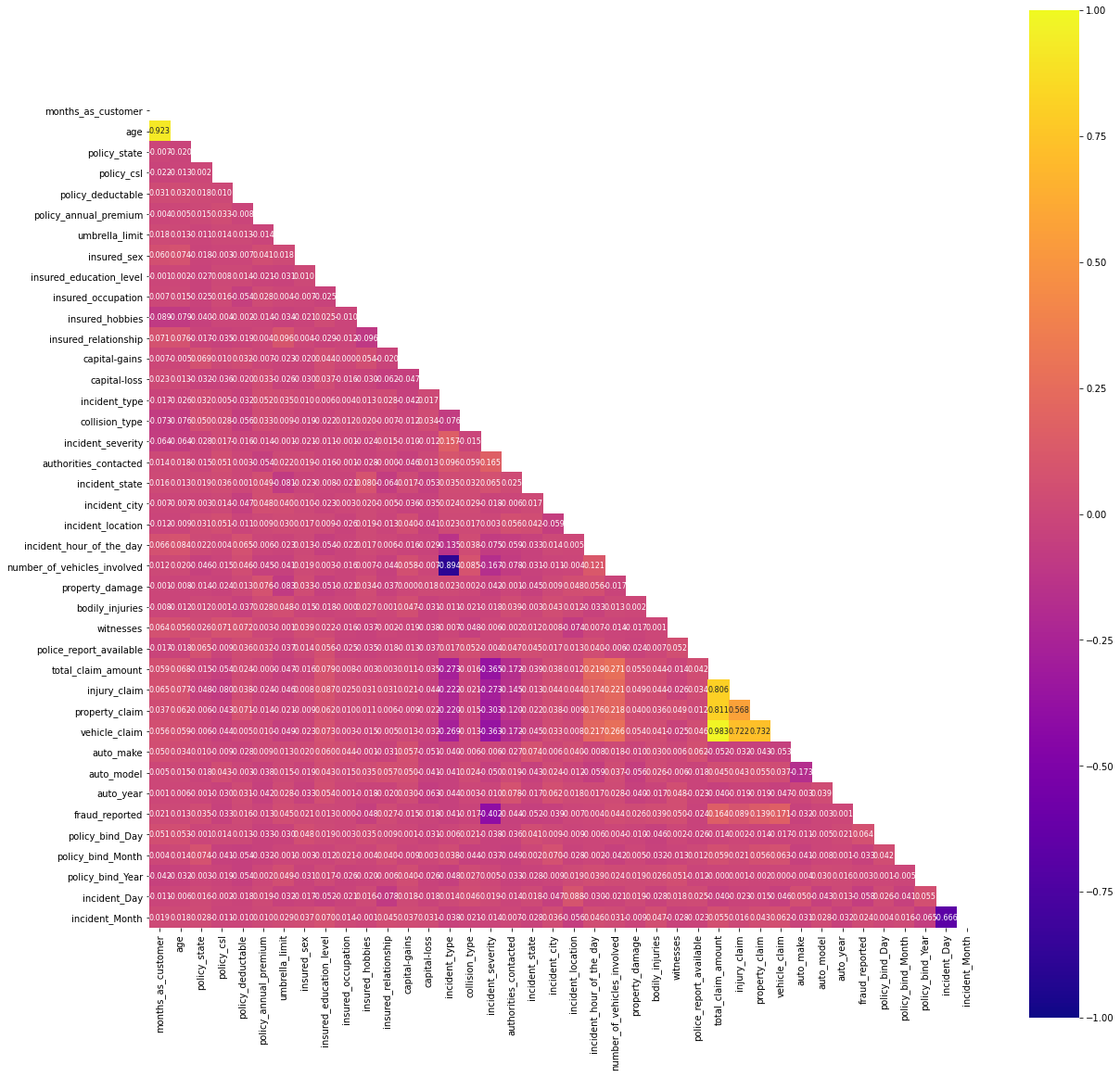
**Correlation using a Heatmap**

A heat map (or heatmap) is a data visualization technique that shows magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity, giving obvious visual cues to the reader about how the features are clustered or varies over space.

Correlation heatmaps are a type of plot that visualize the strength of relationships between numerical variables. Correlation plots are used to understand which variables are related to each other and the strength of this relationship. The rows represent the relationship between each pair of variables. The values in the cells indicate the strength of the relationship, with positive values indicating a positive relationship and negative values indicating a negative relationship.

Code:

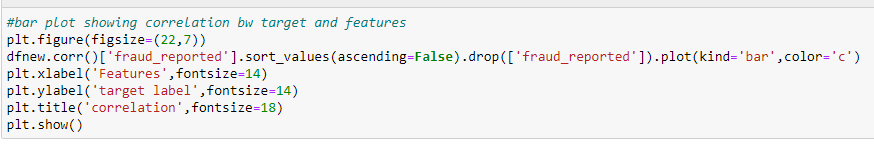


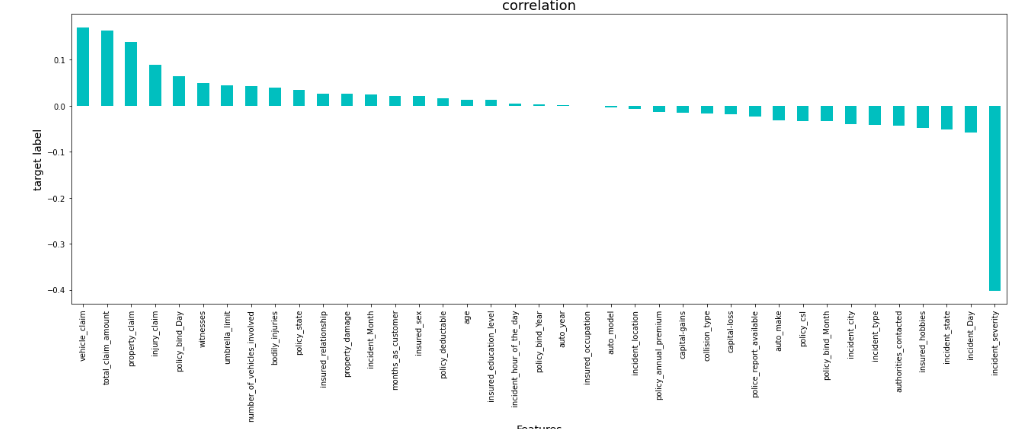


Thus we are not able to find out easily ,but can see few columns shows multicollinearity like all the claims type,incident\_severity etc ,we will be retaining them as they are significant for dependent variable.

**Correlation of features with our label:**

Code and Output:

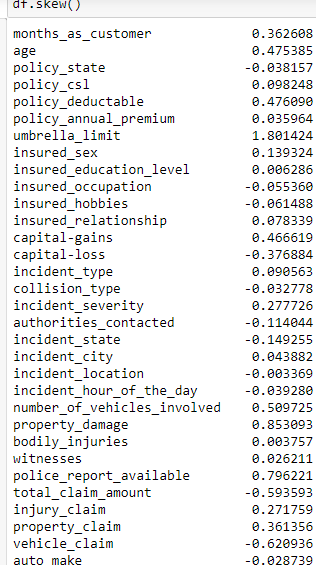




As above the heatmap was not able to give us a clearer picture on positive and negative correlation columns. So we used bar plot which clearly showing the positive and negative correlation with the target column. Thus we see that dependent variable fraud reported is positively correlated with vehicle\_claim,total\_claim\_amount,property\_claim, collision\_type,inury\_claim,policy\_bind\_day,witnesses,umbrella\_limit,policy\_state and negatively correlated with incident severity and incident\_day.So we will retain all the columns.

**Skewness Analysis**

Code and output:



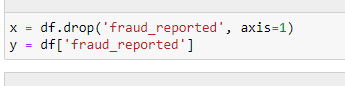
As a general rule of thumb,If skewness is less than -1 or greater than 1, the distribution is highly skewed.

If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed.

If skewness is between -0.5 and 0.5, the distribution is approximately symmetric.

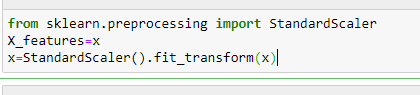
Considering the threshold as -0.5/+0.5,we can see some of the columns are skewed and skeweness will be removed later.Due to screen clipping issue,not able to clip full view of the output.

Now I will be splitting the feature and target column into variable into x and y where x stores the features and y stores the target column. The code for the same as below:



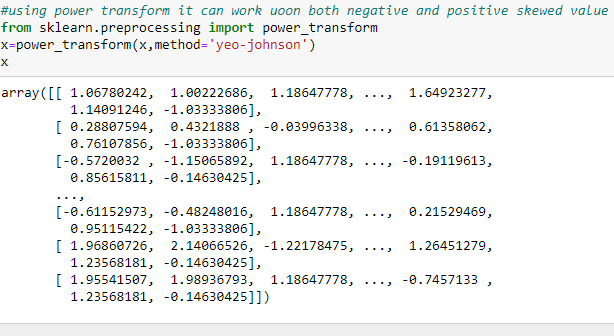
**Feature Scaling**

Then I applied standard scaler to my features columns so that their values share a similar scale and bias towards any column.The code is as below:



**Removing Skewness using PowerTransform**

I applied power transform method to remove skewness . Power transform :Power transforms are a family of parametric, monotonic transformations that are applied to make data more Gaussian-like. The method i have used here is yeo-johnson as it works with both positive and negative values.

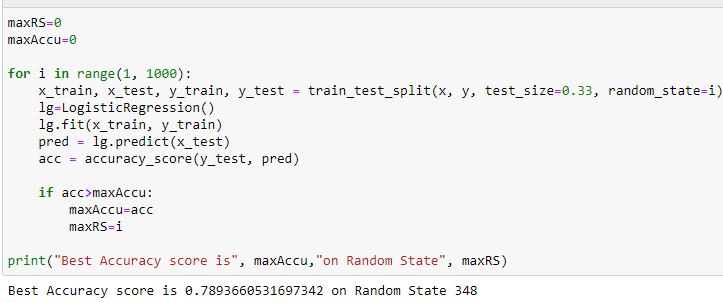


Now I will be fixing class imbalance in my label column by below code:



**6.Building Machine Learning Models**

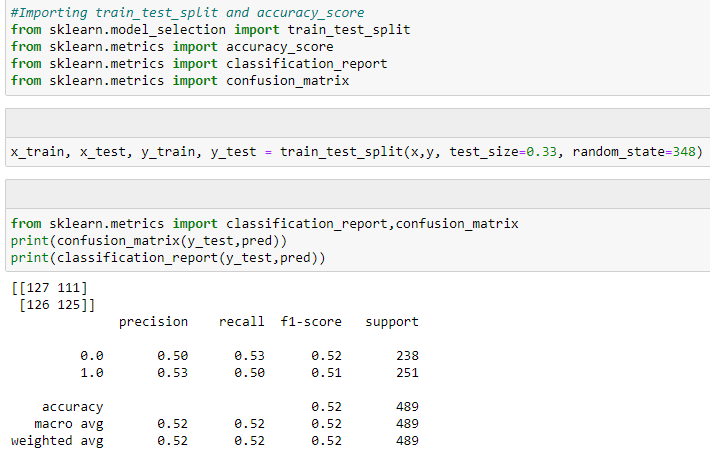
First we need to split the dataset into training data set and testing data set by giving test\_size=0.33(value is userinput) which means 67% goes for training data phase and remaining 33% goes to testing data phase. Before that we can either manually enter the random state or find the best random state as below code and output.

****

Thus we see the best random state is 348 where the accuracy score is 78.93%. Setting random\_state a fixed value will guarantee that same sequence of random numbers are generated each time you run the code. And unless there is some other randomness present in the process, the results produced will be same as always. This helps in verifying the output.Now we will take this random state and give it into the base model as logistic regression.

**Base Model:Logistic Regression;**

So first we need to load the required libraries and perform the train\_test\_split and find the confusion and classification report as below:Code and Output shared below:



Thus we see base model has accuracy score of only 52% and confusion matrix explaination as below:

Confusion matrix:It compares the actual target values with those predicted by the machine learning model and what kinds of errors it is making.Thus we see here in 2\*2 matrix above,different values of the confusion matrix as explained below:

True Positive(TP)=127,It states that 127 positive class data points were correctly classified by the model.

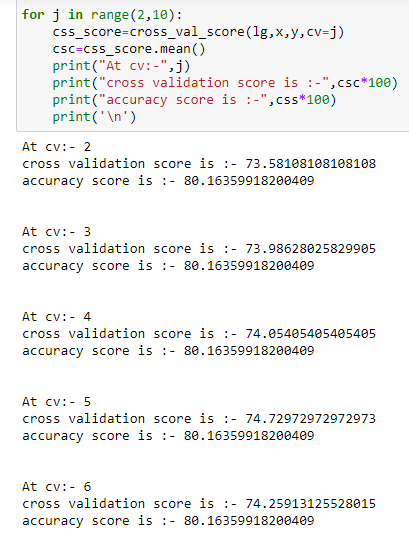
True Negative(TN)=125,It states that 125 negative class data points were correctly classified by the model.

False Positive(FP)=111,It states that 111 negative class data points were incorrectly classified as belonging to the positive class by the model.

False Negative(FN)=126 ,It states that 126 positive class data points were incorrectly classified as belonging to the negative class by the model.

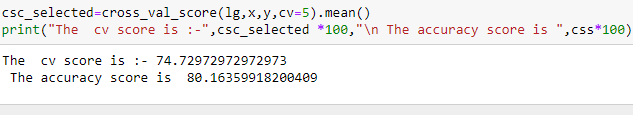
Thus we see that its not a good classifier as from our dataset as there is large number of false positive and false negative values.

now we will perform cross validation on logistic regression.



Thus we see that difference between cross validation score and accuracy score is almost same in all cases,so we will be taking default value that is cv=5 in this case.

Code and output:



We will take cv score as the final accuracy score which is 74.72% as it has reduced the overfitting problem.Thus we see that after cross validation,the accuracy score has improved from 52% to 74.72% which is good.

# **Modelling-Other models**

# It is recommended to take at least 5 machine learning models and choose the best out of it and thereby perform hyperparameter tunning so that it can perform more better with more accuracy. I have taken Decision tree classifier, Random Forest Classifier,Ada Boost Classifier,KNeighbors,Support Vector Classifier as my 5 machine learning models .Evaluation metrics as confusion matrix,classification report,f1,recall,precision,accuracy.

# **Random Forest Classifier**:It is an ensemble technique which works on the principle of bagging.It builds decision tree on different samples and final output decided upon majority voting for classification and averaging for regression.It takes care of overfitting issues.

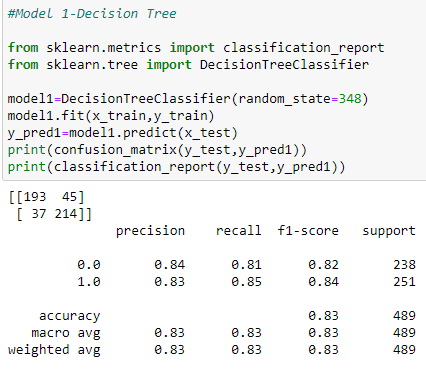
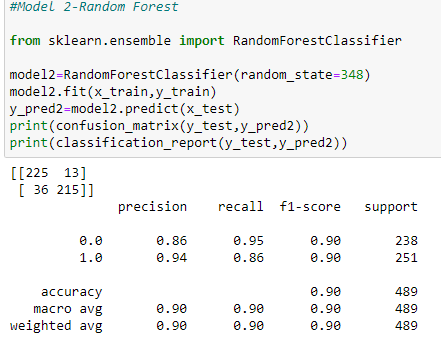
# **Decision Tree Classifier**:It creates the classification model by building a decision tree.It starts with a single point called node,from node it branches out into two or more different pathways specifying outcomes or decisions until final outcome is achieved. It suffer from the problem of overfitting if it’s allowed to grow without any control.

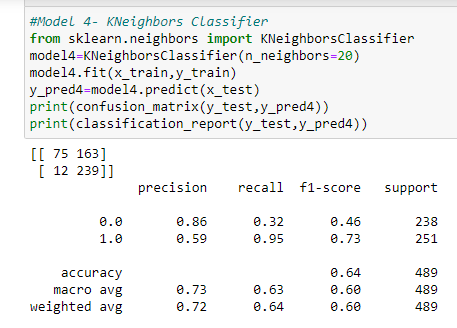
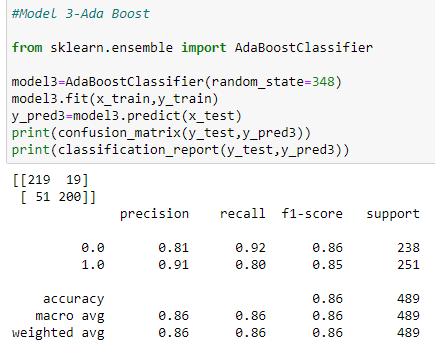
# **Ada Boost Classifier**: An AdaBoost [1] classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.It is one of ensemble boosting classifier .It converts the weak leaners into strong leaners.

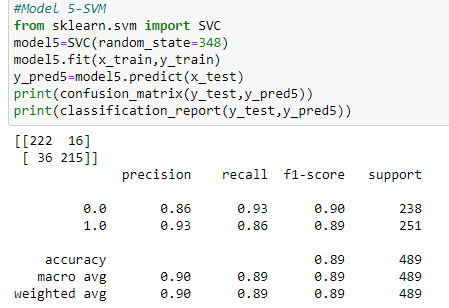
# **Kneighbors Classifier**: It is based on the k nearest neighbors of a sample, which has to be classified. The number 'k' is an integer value specified by the user.

# **Support Vector Classifier**: They are a set of supervised learning methods used for [classification](https://scikit-learn.org/stable/modules/svm.html#svm-classification), [regression](https://scikit-learn.org/stable/modules/svm.html#svm-regression) and [outliers detection](https://scikit-learn.org/stable/modules/svm.html#svm-outlier-detection). The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine

# Code and output of five models used are given below

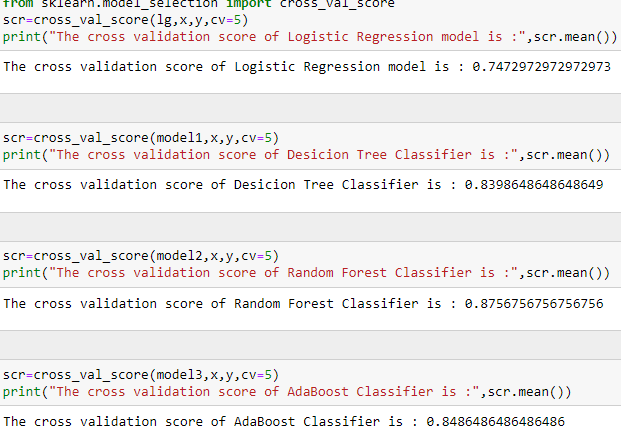


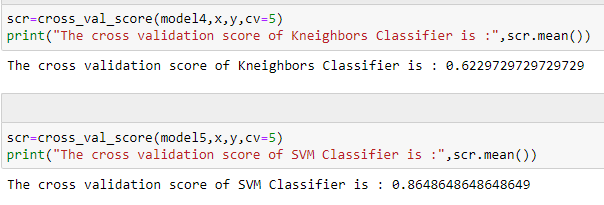


|  |  |
| --- | --- |
| Thus we see, |  |
| **MODEL** | **Accuracy Score** |
| Decision tree classifier | 83% |
| Random Forest Classifier | 90% |
| Kneighbors Classifier | 64% |
| Ada Boosting Classifier | 86% |
| Support vector machine (SVM) | 89% |

Conclusion: We see that almost all the models are showing good accuracy rate except Kneighbors,Now we will check cross validation for all the models for overfitting.

**Cross validation for all the models: code and output**:





Thus we see that after doing cross validation ,we can see the scores that Cross validation score of all the models have been reduced which means that cross validation is not decreasing the accuracy,it is rather giving us a better approximation for that accuracy, including less overfitting.

|  |  |
| --- | --- |
|  |  |
| **Models** | **Cross validation score** |
| Desicion Tree | 83.98% |
| RandomForest: | 87.56% |
| AdaBoost: | 84.86% |
| Kneighbors: | 62.29% |
| SVM: | 86.48% |

So I am taking Random Forest Classifier and SVM as their accuracy is close to each other for hyperparameter tuning under GridSearch compare to other models.

**Hyper parameter Tunning**:

Hyper parameter optimisation in machine learning intends to find the hyper parameters of a given machine learning algorithm that deliver the best performance as measured on a validation set.

Using Scikit-Learn’s GridSearchCV method, we can explicitly specify every combination of settings to try. We do this with GridSearchCV, a method that, instead of sampling randomly from a distribution, evaluates all combinations we define.

**RANDOM FOREST HYPERTUNNING**

Random Forest Classifier-Hypertunning parameters: I have used n\_estimators = number of trees in the foreset max\_features = max number of features considered for splitting a node max\_depth = max number of levels in each decision tree criterion =The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain. Code and output as below:

****

**Support Vector Classifier-Hyper tunning**

Support Vector Classifier-Hypertunning parameters: I have used c = *float, default=1.0,*Regularization parameter, **kernel=***{‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’} or callable, default=’rbf’* Specifies the kernel type to be used in the algorithm. If none is given, ‘rbf’ will be used. Degree :*int, default=3,*Degree of the polynomial kernel function (‘poly’). gamma*{‘scale’, ‘auto’} or float, default=’scale’,*Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.Code and output as below:



**Model Analysis**

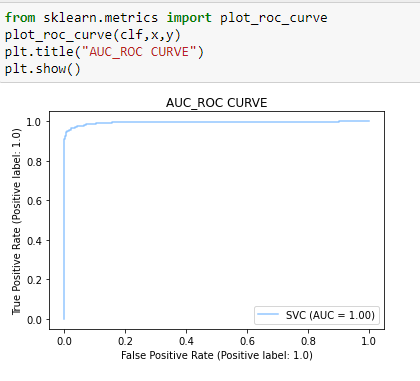
Comparing both the models that is random forest classifier and Support Vector classifier considering the hyperparameter tuning Model Accuracy Score Cross validation score.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Models-Final | Accuracy Score | Cross Validation Score |
| Random Forest Classifier | 89.77% | 86.95% |
| Support Vector Classifier | 90.79% | 90.20% |

We find that support vector classifier performs well considering that cross validation score after hypertuning has improved from 86.95 to 90.20% and accuracy score has improved from 86.48% to 90.79% which has reduce overfitting issues. Also the gap between accuracy score and cross validation score is very less .Thus we take this our best fit model.

**AUC\_ROC PLOT**

It is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s.

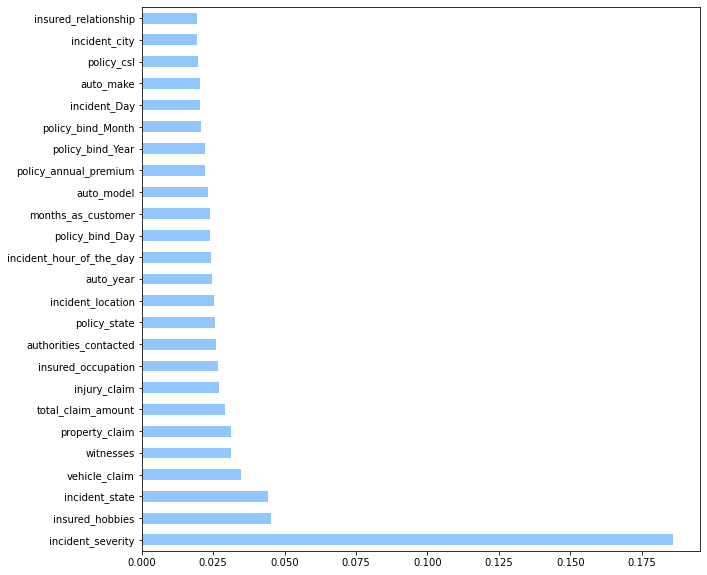
****

OBSERVATION: So final accuracy is 90% as our cross validation score accuracy and auc\_roc score is 100 % which is very good and here we can see our AUC (c-value) is far greater than 0.5 ,it is 1.00 which can be considered very good which means that the classifier is able to distinguish between 0s and 1s. So it can distinguish the whether auto fraud claim is genuine or not. We can definitely employ this model.

**Feature Importance**

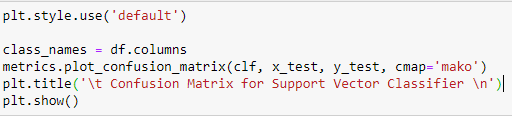
Feature Importance refers to techniques that calculate a score for all the input features for a given model — the scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.

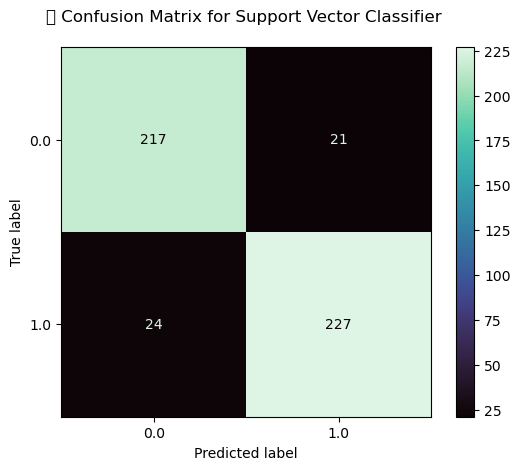




Thus we see that incident\_severity is the most important feature for insurance claim fruad detection followed by insured\_hobbies.

**Confusion Matrix:**





Thus we see confusion matrix after hypertuning gives us result as below: Confusion matrix:It compares the actual target values with those predicted by the machine learning model and what kinds of errors it is making.Thus we see here in 2\*2 matrix above,different values of the confusion matrix as explained below:

True Positive(TP)=217,It states that 217 positive class data points were correctly classified by the model.

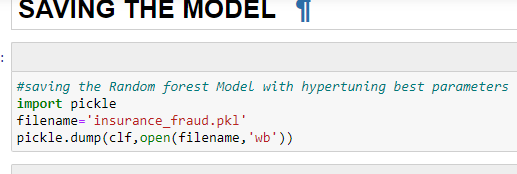
True Negative(TN)=227,It states that 227 negative class data points were correctly classified by the model.

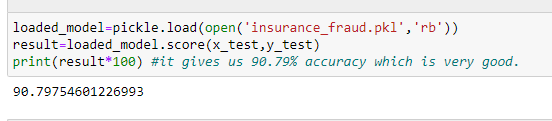
False Positive(FP)=21,It states that 21 negative class data points were incorrectly classified as belonging to the positive class by the model.

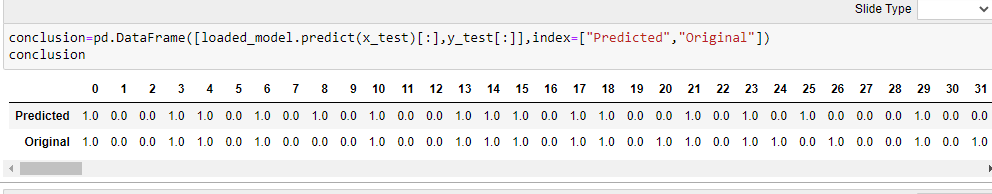
False Negative(FN)=24 ,It states that 24 positive class data points were incorrectly classified as belonging to the negative class by the model.

Thus we see that its a good classifier as from our dataset it can disntinguisn larger number of true positive and true negative values and errors are less

Finally after getting the best fit model, now I will be saving the final model either using joblib or pickle. I have used the pickle method to save and then load my model from the same saved filename and then I have done the prediction test to see my predicted result.

****

****



This test shows that the model is able to predict the correct accuracy of 90.79%.Thus we see model is best fit model.

**CONCLUDING REMARKS:**The objective of the business case was to create a predictive model that predicts if an insurance claim is fraudulent or not. So first I started with loading the dataset and carry out data analysis and then did the EDA process with visualization patterns using pie-plot, box plot, distribution plot ,countplot, violinplot and learnt about different relationship between the features.

After that I did pre-processing techniques like checking outliers, removal of skewness, encoding of categorical column, balancing of imbalance data with smote and scaling of data for normalization.

Then I took base model as logistic regression for training and testing the dataset and find out the accuracy score, confusion matrix and classification report,f1 scrore.Then also perform cross validation to reduce the overfitting problem.

It is advisable to take at least 5 machine learning. So I took 5 below models to check their accuracy.To evaluate the performance of the machine learning algorithms (Logistic Regression, Random Forest, Support Vector Machine(SVM), AdaBoost,Kneighbors,Decision Tree), following metrics were used: Precision, Recall, F1-score, ROC Area, Classification report,confusion matrix. We find that SVM and Random Forest were best fitted model, so I performed hyper tuning through gridserachcv on these models.

I found that SVM model was having the highest precision accuracy for fraud detection problem with machine learning data. Hence by implementation of this model the insurance company can get accurate results in short duration of time and limited manual intervention. I saved the best model using pickle method and loaded the model for prediction test and find 90% accuracy which is very good. Thus this model can be used in further deployment process.

**I want to thank Datatrained institute and internet for the valuable knowledge providing me from time to time to carry out the blog report.**

**GitHubLink:https://github.com/shilpimohanty85/Practise-Projects/blob/main/insurance\_fraud\_project10.ipynb**