# A blog by Ashish Kumar Rathor On Insurance Fraud



## **Insurance Claim Fraud Detection Project**

**In association with Data Trained Academy: Batch: 1835**

**Introduction:**

Fraud is one of the largest and most well-known problems that insurers face in the insurance industry. This article focuses on the claim data of automobile insurance.

Insurance fraud is a deliberate deception perpetrated against or by an insurance company or agent for financial gain. Fraud may be committed at different points in the transaction 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.

. Here in this blog, I have created a Machine Learning model to detect if the claim is fraudulent or not. Here various features have been used, insured information, insured persons, personal details, and incident information. In total the dataset has 40 features. So, using all this previously acquired information and analysis done with the data I have achieved a good model that has 92% accuracy. So let’s see what steps are involved to attain this accuracy.

Various visualization techniques have also been used to understand the co-linearity and importance of the features.

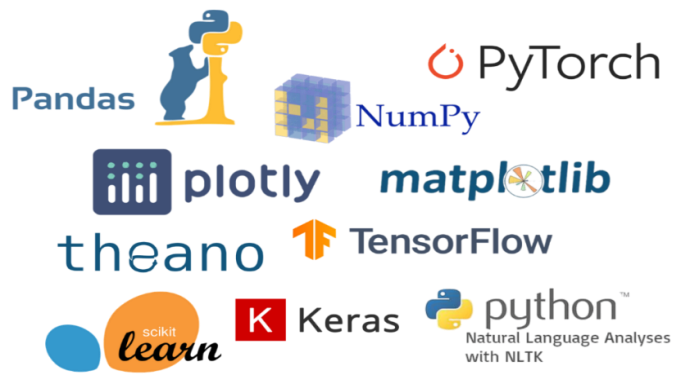
# Hardware & Software Requirements & Tools Used:

### **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
* Pandas
* Matplotlib
* Seaborn
* Date Time
* Scikit Learn

**Problem Definition:**

Insurance fraud is a huge problem in the industry. It's difficult to identify fraud claims. Machine Learning is in a unique position to help the Auto Insurance industry with this problem. In this project; we are provided a dataset that has the details of the insurance policy along with the customer details. It also has the details of the accident based on which the claims have been made.

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

In this problem, we will be looking into the insured person’s details and the incidents and analyzing the sample to understand if the claim is genuine or not.

**Importing the necessary Libraries:**

To analyze the dataset or even to import the dataset, we have imported all the necessary libraries as shown below.

Pandas have been used to import the dataset and also in creating data frames.

Seaborn and Matplotlib have been used for visualization

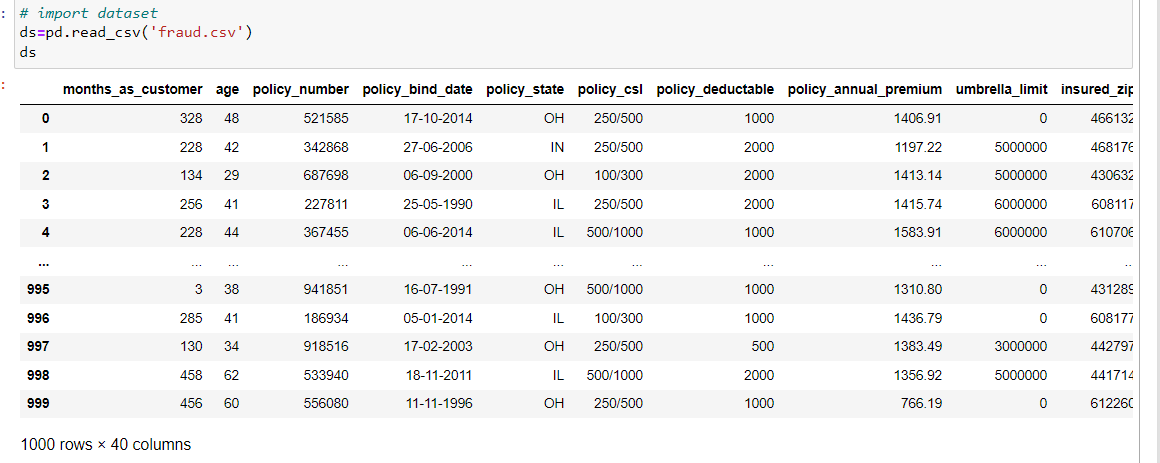
Date Time has been used to extract day/month/date separately

Sklearn has been used in the model building



**Importing the Dataset**

Let’s import the dataset first.



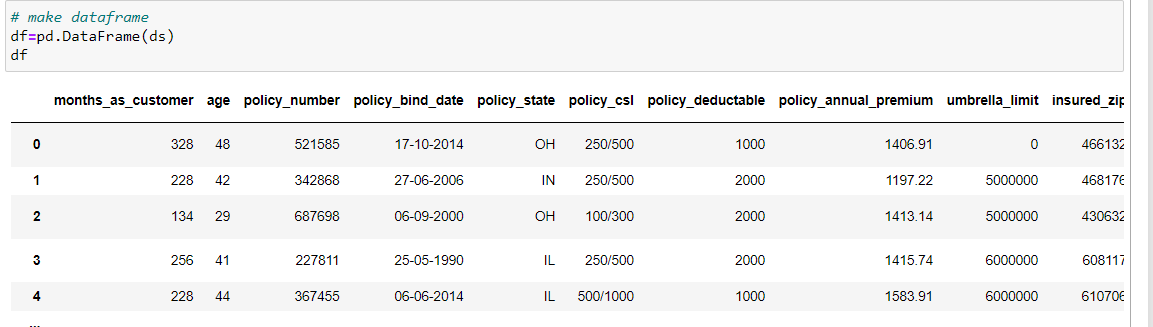
I have imported the dataset which was in “CSV” format as “ds”. Below is how the dataset looks.

By observing the dataset, we could make out that the dataset contains both categorical and numerical columns. Here "fraud reported" is our target column since it has two categories so it is termed to be a "Classification Problem" where we need to predict if an insurance claim is fraudulent or not. As it is a classification problem hence, we will be using all the classification algorithms while building the model that we will see as the blog proceeds.

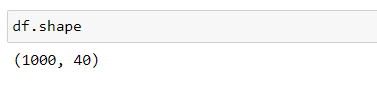
I do not see all the columns so I import ‘max columns’ and then see all the columns of the dataset.



Then I change its type from series to data frame and put it into the frame.



Also, by doing a simple code ‘df. shape’ we also figured out how many rows and columns we have. We have got the result that we have 1000 rows and 40 columns. PCA can be done, however, I decided not to lose any data at this time as the dataset is comparatively small and the first lesson of a data scientist is ‘Data is Crucial’ hence proceeded will all the data.

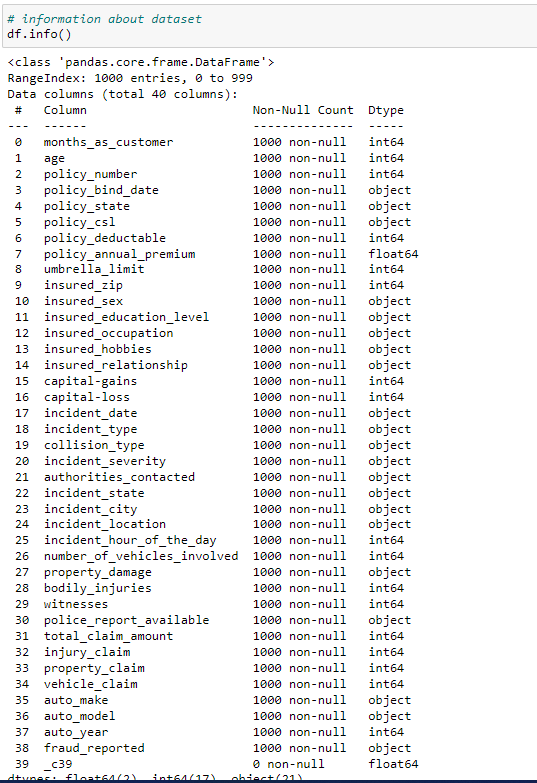


**Exploratory Data Analysis and Data Preparation:**

In this part, we will first be exploring the data with some basic steps and then further proceed with some crucial analysis, like feature extraction, imputing, and encoding.

Let’s start with checking shape, unique values, value counts, info, etc….

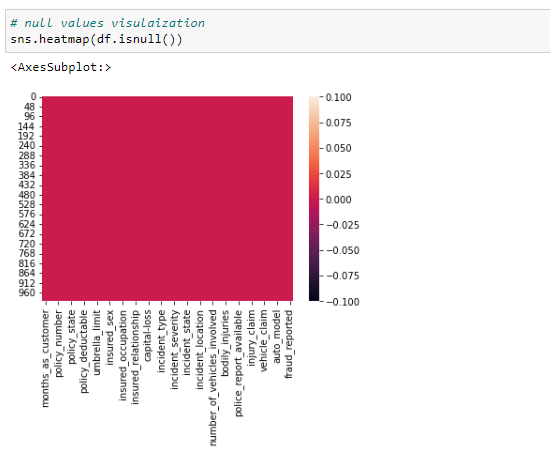
After analyzing if we find any unnecessary columns in the dataset, we can drop those columns.



In our dataset, 2 out of 40 columns are float-type columns and 17 out of 40 columns are int-type and 21 out of 40 columns are object-type datasets. Here in column no. 39 all values are null so we remove /drop it.

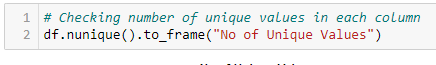


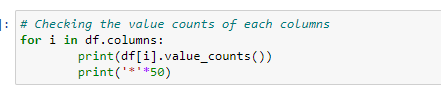
**NULL VALUES:**

Then I check no.of unique values in ****

No null values are present in our dataset.

Then I check no.of unique values in our dataset, I found that in the 'policy\_number', 'insured\_zip', and 'incident\_location’ columns more than 98% of unique values are present so we drop it, It raises unnecessary problems during modeling



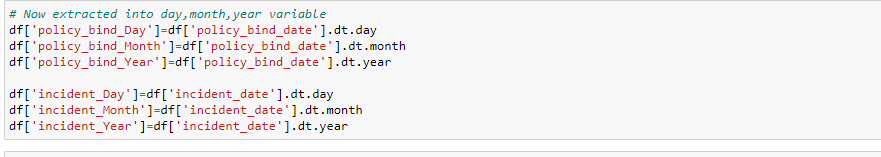
* Then I went for checking all counts values in each column, Then I found more than 80% of zeros are present in the ‘umbrella limit’ then I decided to remove this feature. umbrella limit, capital gains, and capital loss contain more zero values around 79.8%, 50.8%, and 47.5%. I am keeping the zero values in the capital gains and capital loss columns as it is. Since the umbrella limit column has more than 70% zero.

**Proceeding to Feature Extraction:**

The policy\_bind\_date and incident date have object data types that should be in a Date Time data type which means python is not able to understand the type of this column and give the default data type. We will convert this object data type to the Date Time data type and we will extract the values from these columns.



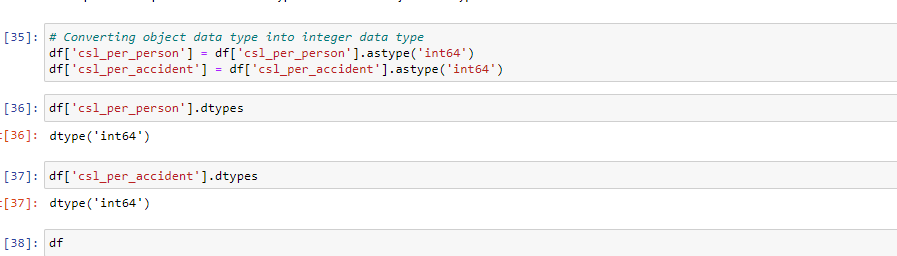
Now that we have converted the object data type into the Date Time data type. Now let's extract Day, Month, and Year from both columns



After we have extracted Day, Month, and Year columns, from both the policy\_bind\_date and incident date columns. So we can drop these columns.



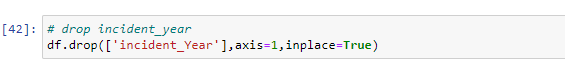
Again, from the features we can see that the policy\_csl column is showing as an object data type but it contains numerical data, maybe it is because of the presence of "/" in that column. So first we will extract two columns csl\_per\_person and csl\_per\_accident from policy\_csl columns and then will convert their object data type into an integer data type.





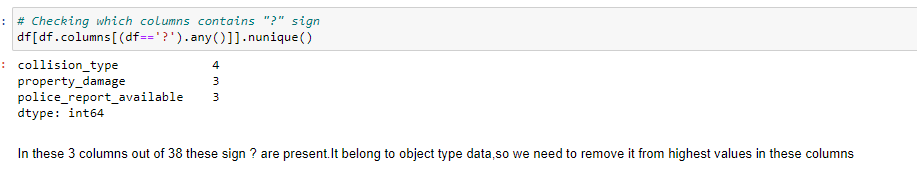
After extracting we have dropped the policy\_csl feature.

Also, we have observed that the feature ‘incident-year’ has one unique value throughout the column also it is not important for our prediction so we can drop this column.

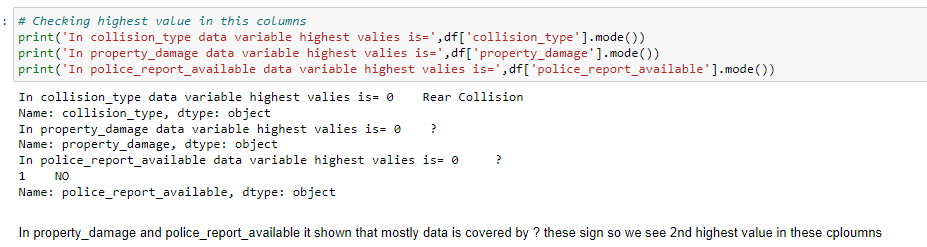


**Moving on to Imputation:**

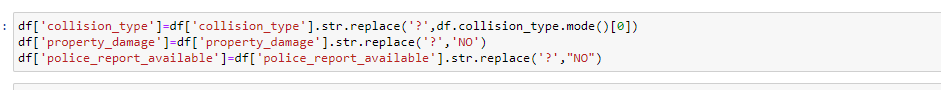
Imputation is a technique to fill null values in the dataset using mean, median, or mode. YES…. I know you might be thinking that we did not get any null values while checking for the null values, however from the value counts of the columns we have observed that some columns have "?" values, they are not NAN values but we need to fill them.



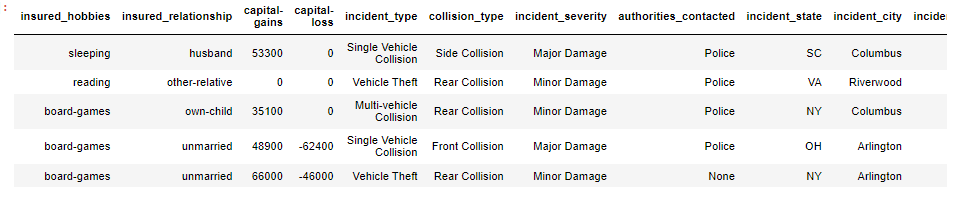
These are the columns that contain the "?" sign. Since this column seems to be categorical so we will replace "?" values with the most frequently occurring values of the respective columns that is their mode values.



The mode of property damage and police\_report\_available is "?", which means the data is almost covered by the "?" sign. So we will fill them by the second highest count of the respective column.



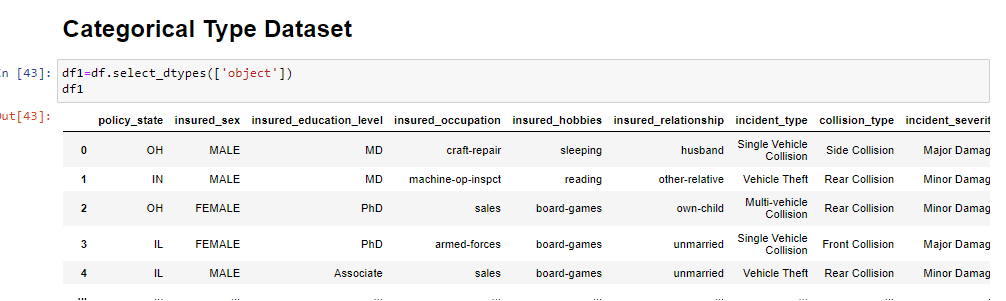
Now after all the data cleaning until now, the dataset looks like this……



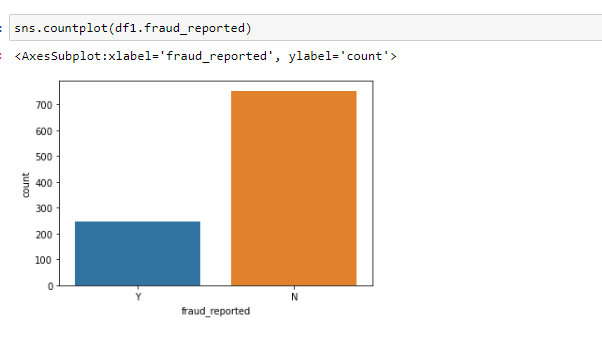
Now we separate both numerical and categorical data into 2 different datasets like df1, and df2 for a better understanding of the data.

**VISUALIZATION:**

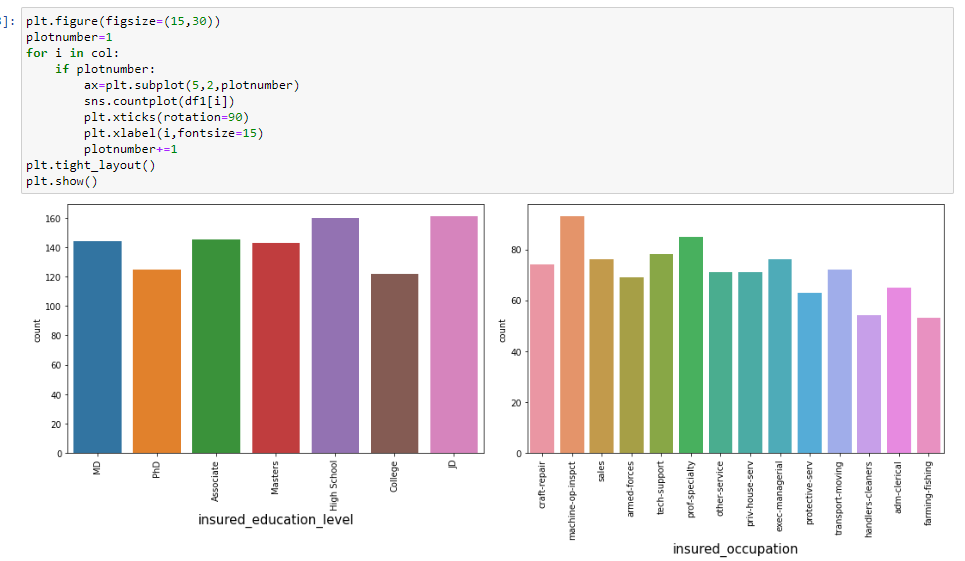
1. **Categorical Dataset**

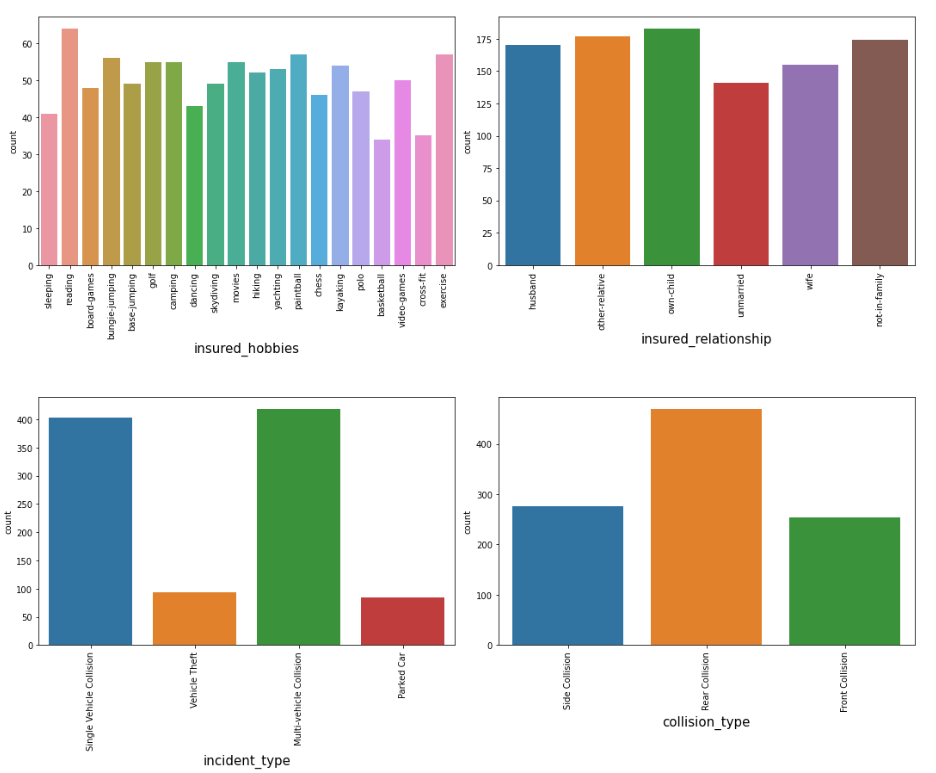


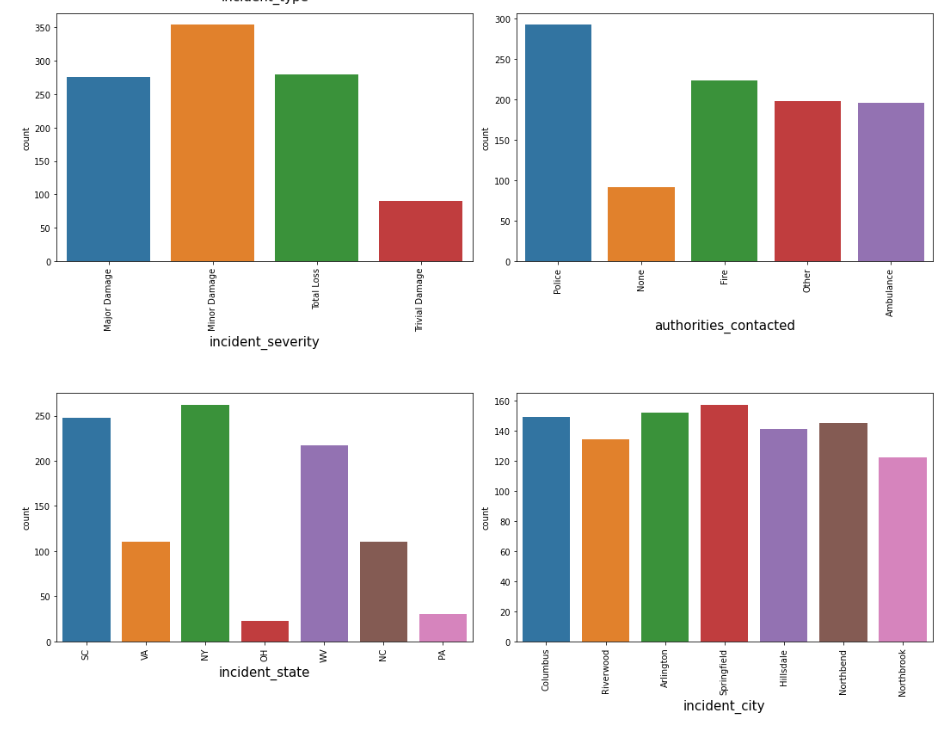
UNI-VARIANT ANALYSIS



By looking into the plot we can observe that the count of "N" is high compared to "N". This means here we can assume that "Y" stands for "Yes" that the insurance is fraudulent and "N" stands for "No" which means the insurance claim is not fraudulent. Here most of the insurance claims have not been reported as fraudulent. Since it is our target column, it indicates the class imbalance issue. We will balance the data using the oversampling method in the later part.



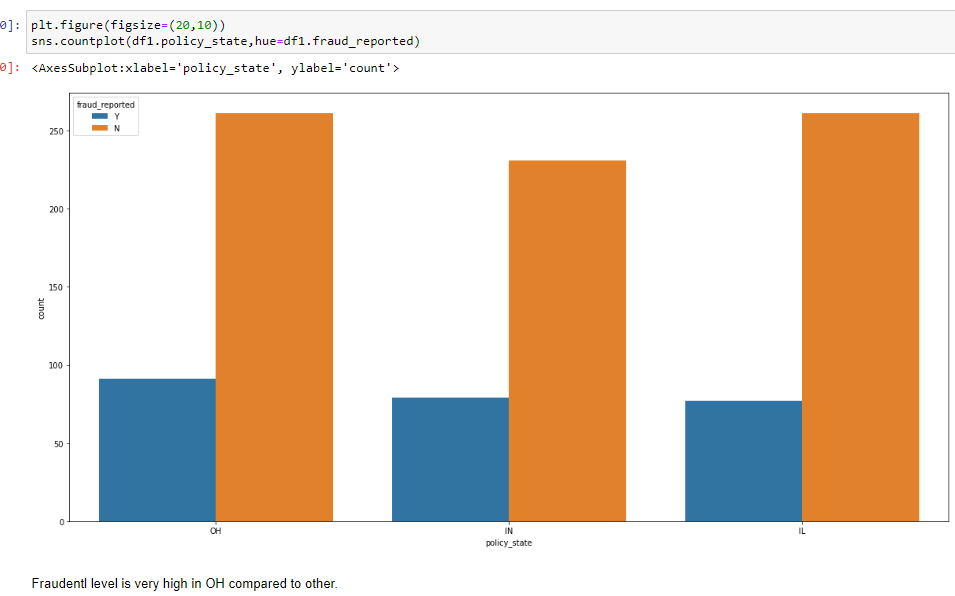


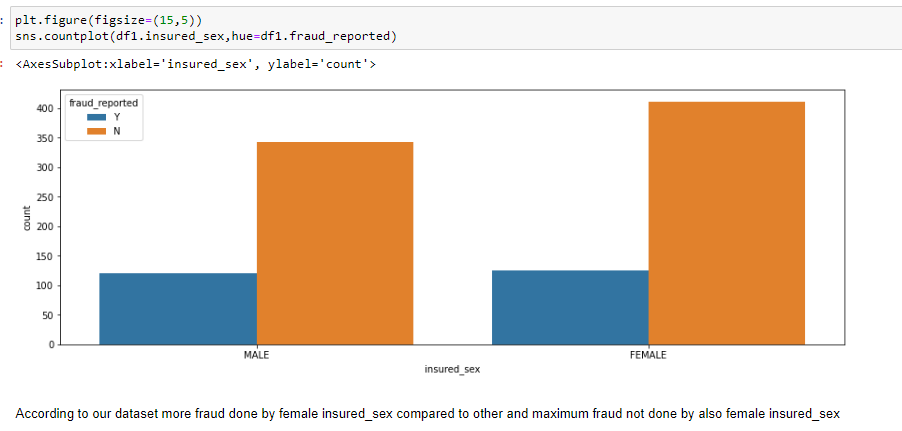


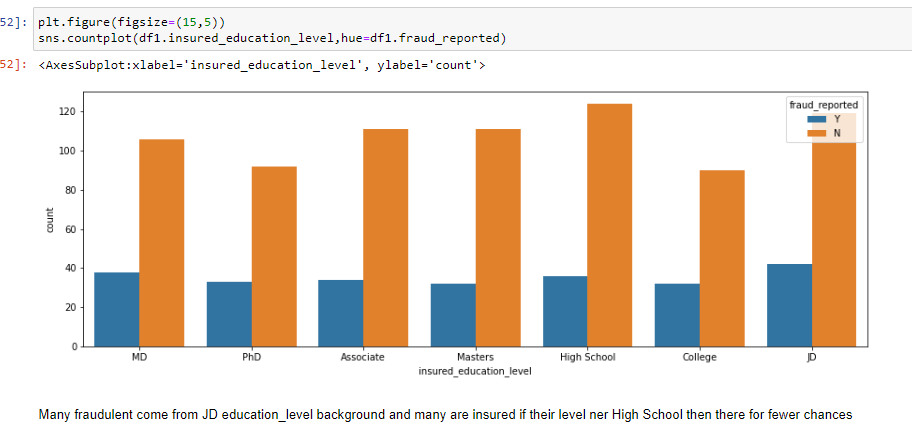
By looking into the count plots we can observe the following things:

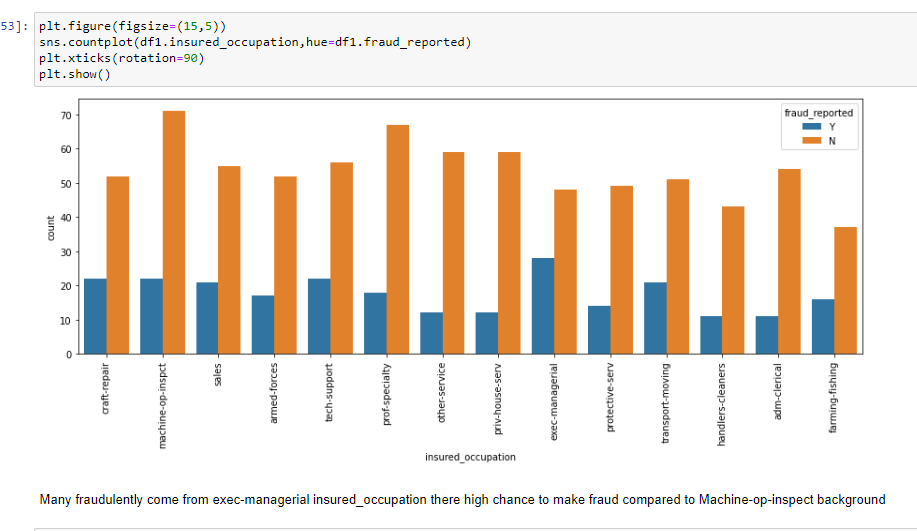
* In the insured occupation we can observe most of the data is covered by a machine operation inspector followed by a professional specialty.
* Concerningto insured hobbies, we can notice reading covered the highest data followed by exercise. And other categories have the average counts.
* The incident severity count is high for Minor damages and trivial damage data has a very less count compared to others.
* When accidents occur most of the authorities contacts the police, here the category police cover the highest data, and Firehasg is the second highest count. But Ambulance and Others have almost the same counts and the count is very less for none compared to all.
* Concerning the incident state, New York, South Carolina, and West Virginia states have the highest counts. In the incident city, almost all the columns have equal counts.
* When we look at the vehicle manufactured companies, the categories Saab, Suburu, Dodge, Nissan, and Volkswagen have the highest counts.
* When we take a look at the vehicle models the RAM and Wrangler automobile models have the highest counts and also RSX and Accord have very counts.

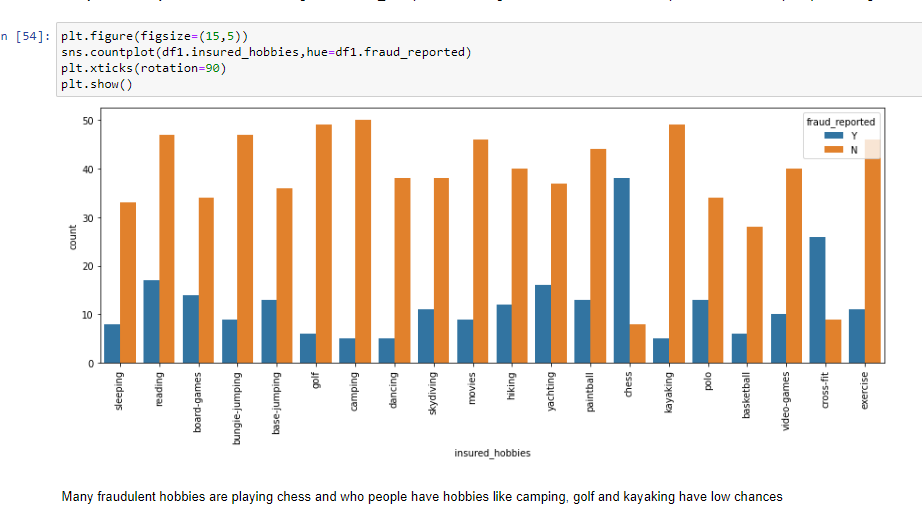
BI-VARIANT ANALYSIS

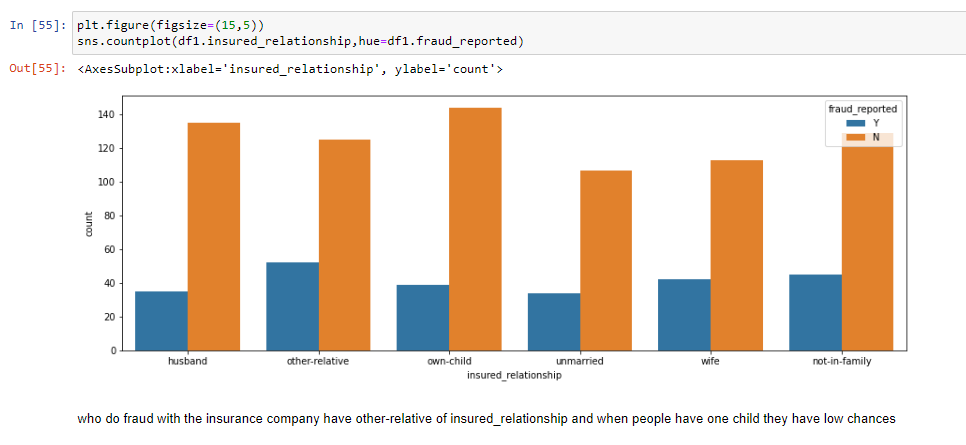


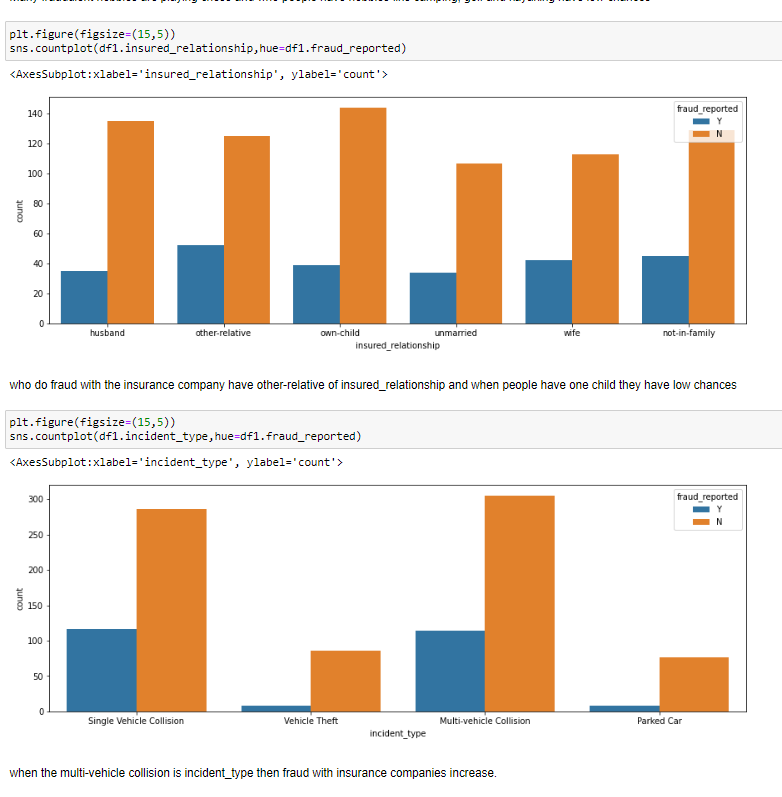


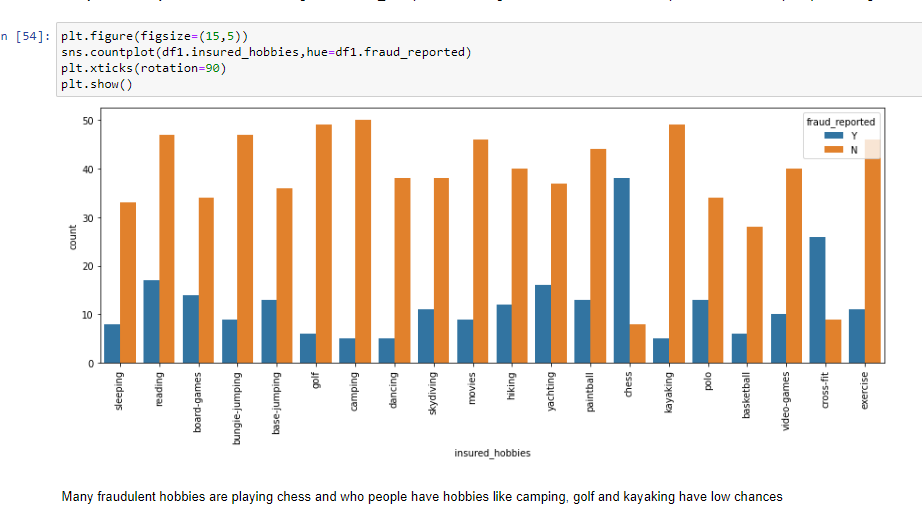


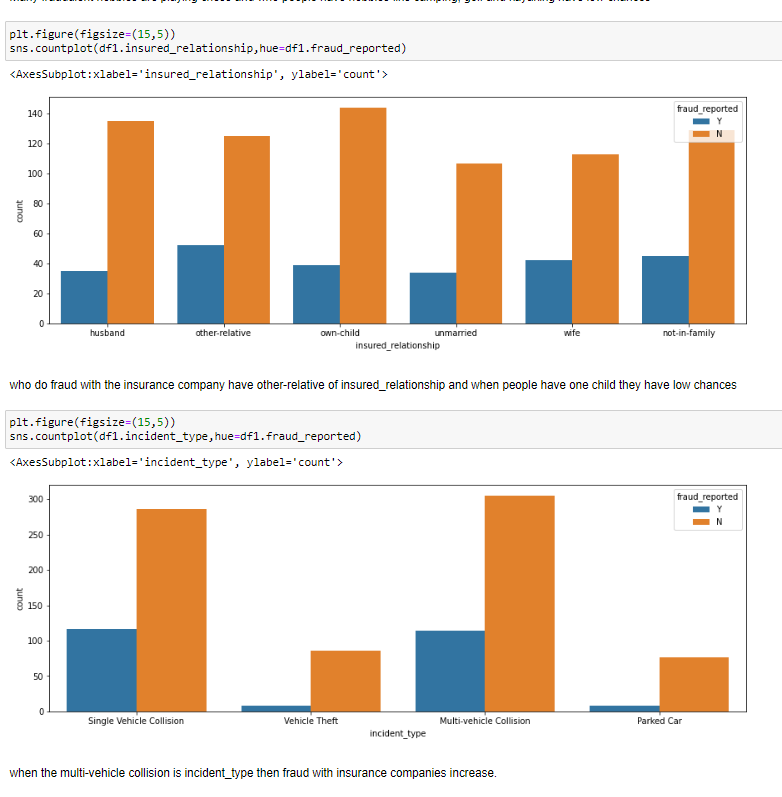


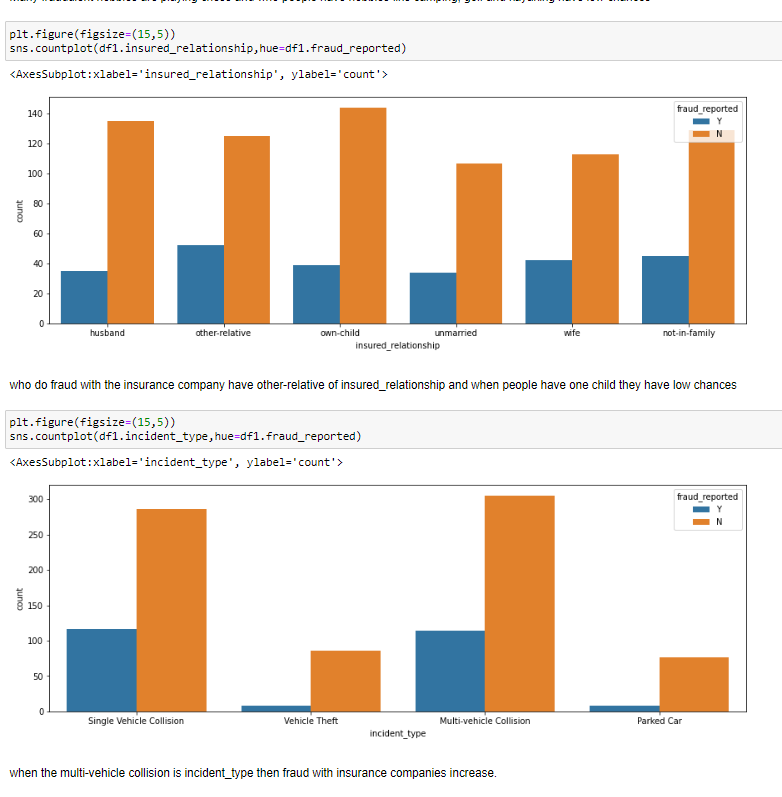


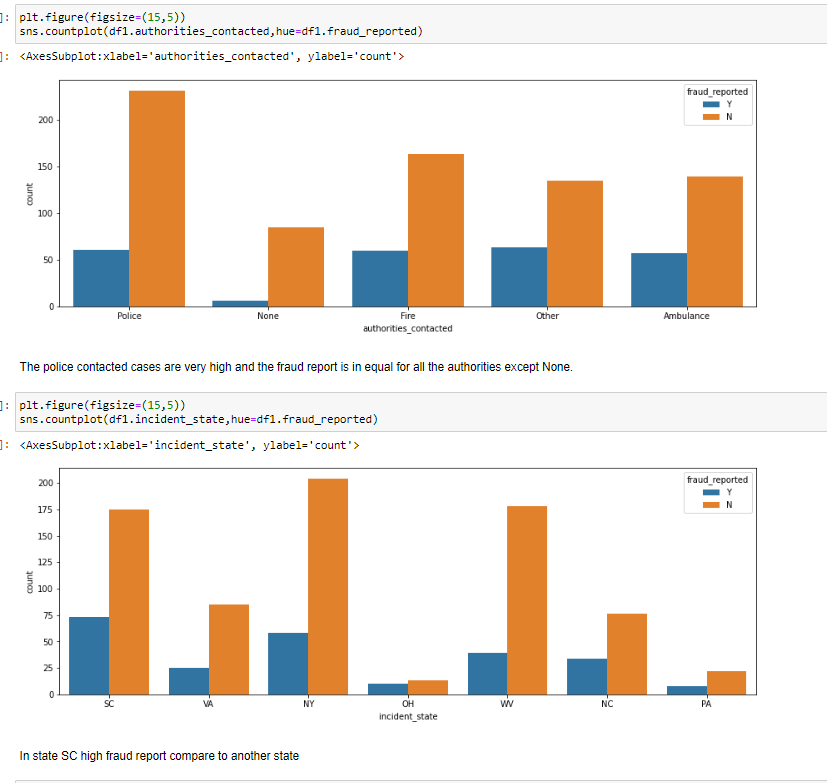


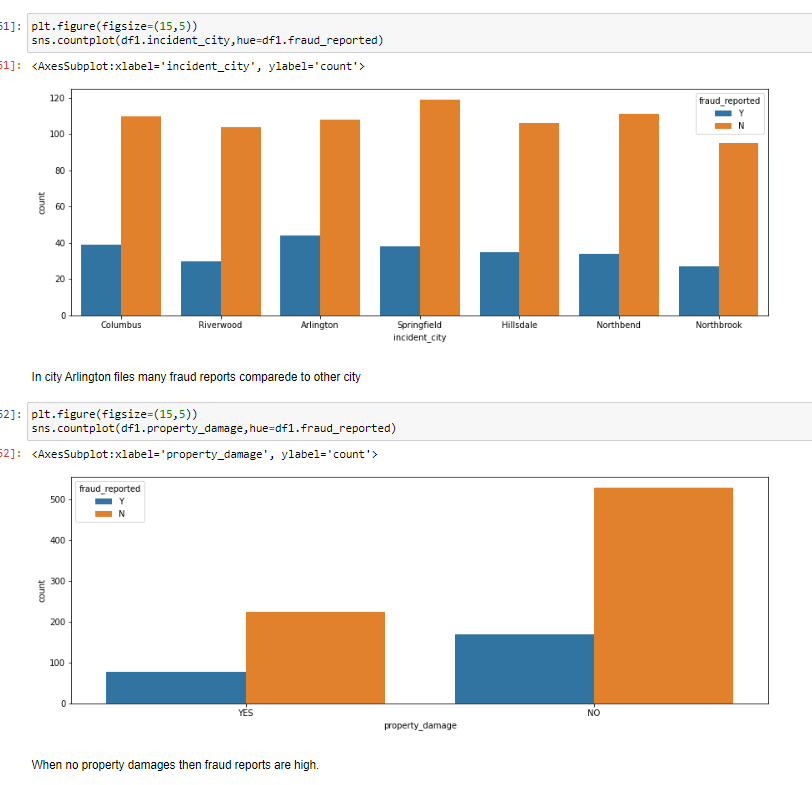




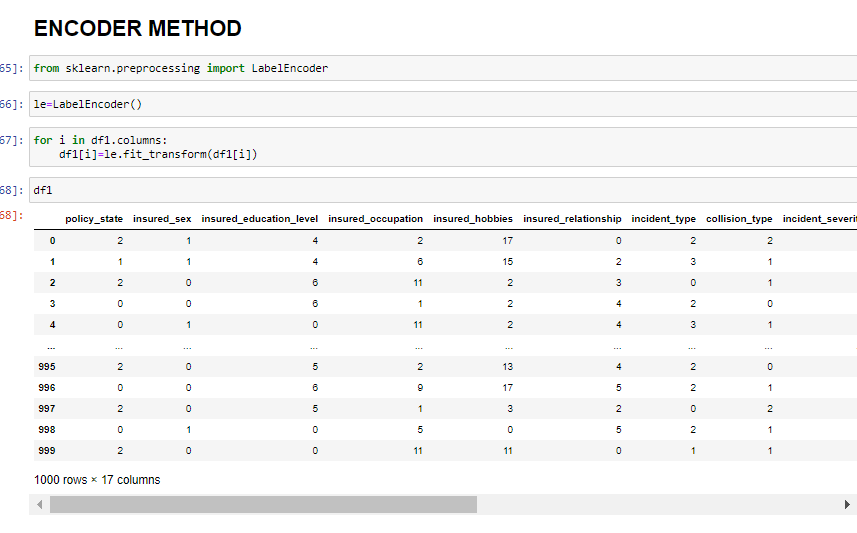




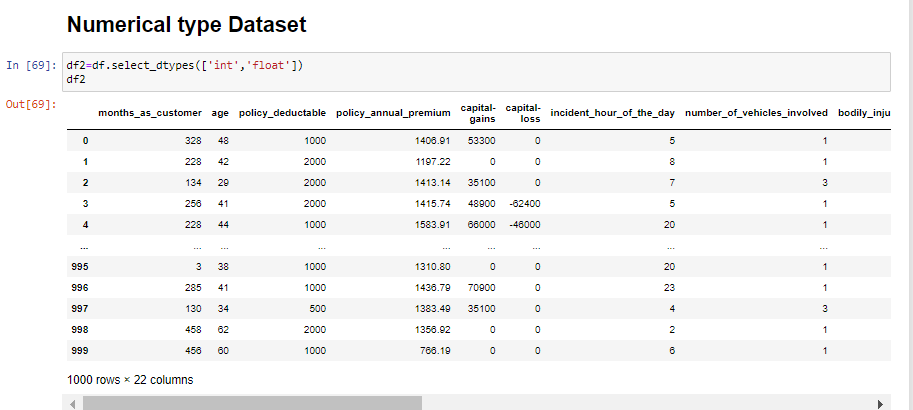




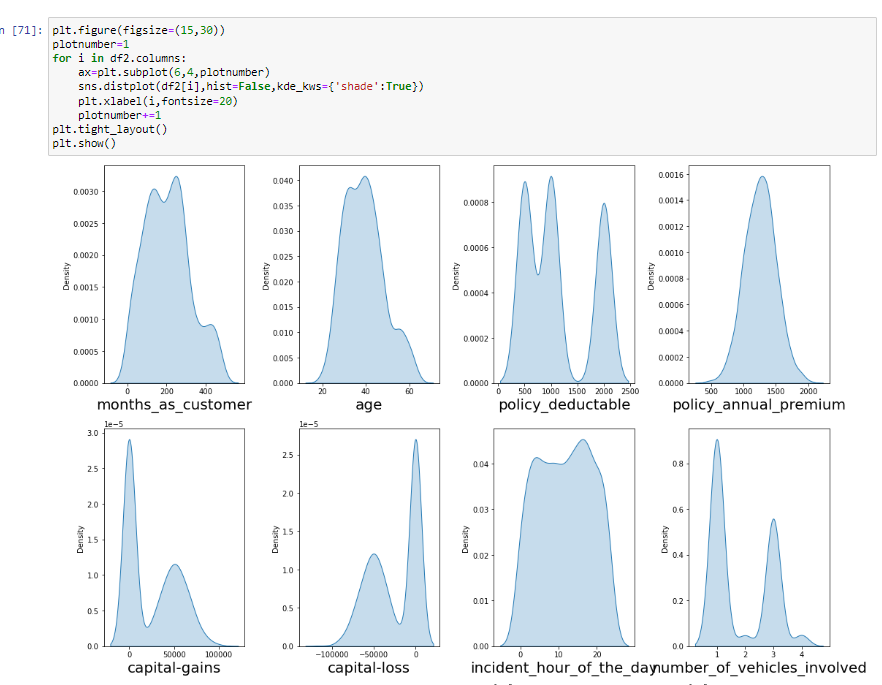
**ENCODER METHOD**🡪CHANGE OBJECT TYPE TO INT TYPE FOR NO PROBLEM RAISE DURING TRANSFORMATION METHOD AND MODELING.

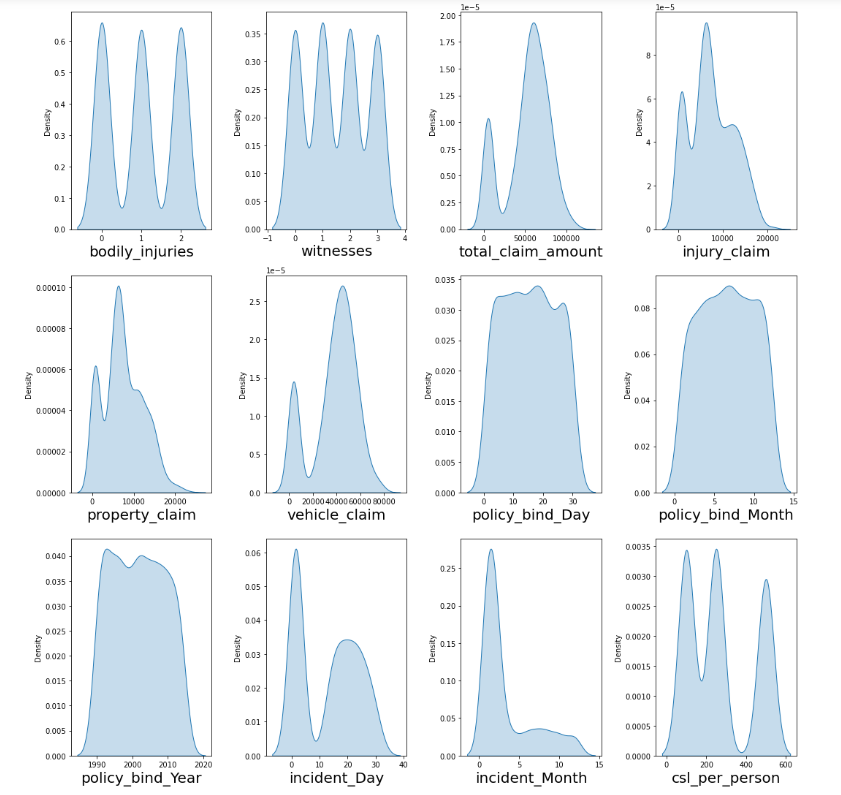


**2.NUMERICAL DATASET**

****

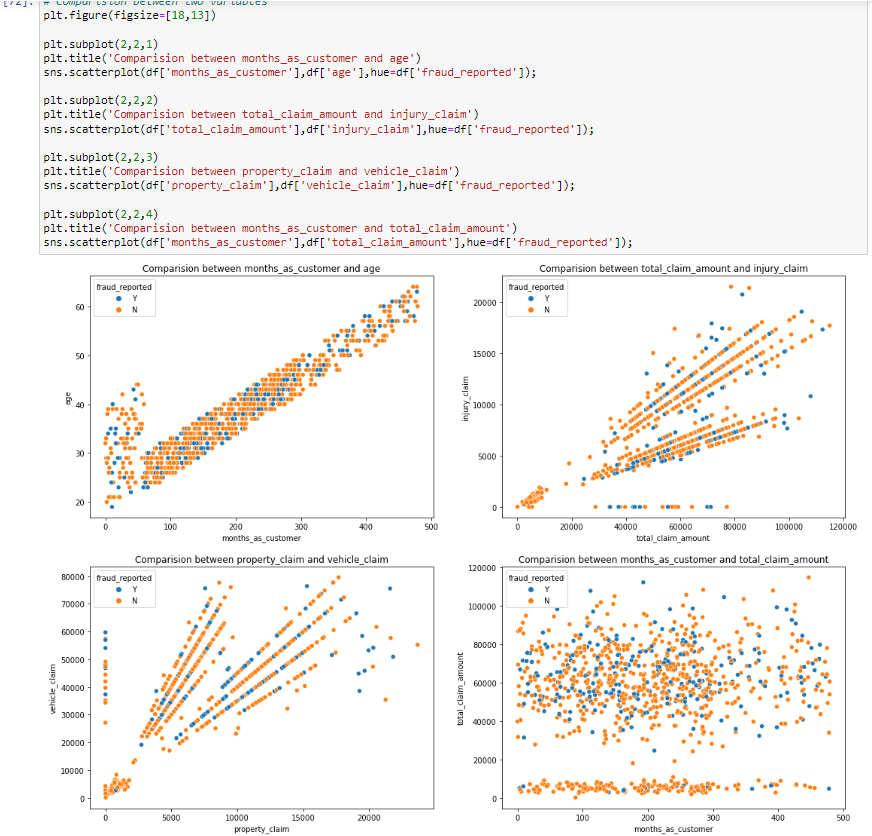
**UNI-VARIANT ANALYSIS**

****

****

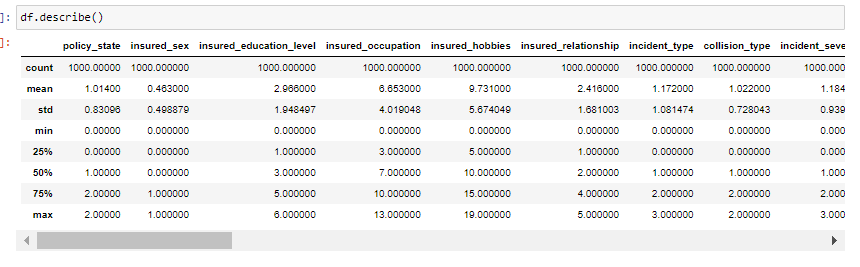
The data is normally distributed in most of the columns. Some of the columns like capital gains and incident months have a mean value greater than the median, hence they are skewed to right. The data in the column capital loss is skewed to the left since the median is greater than the mean. We will remove the skewness using appropriate methods in the later part.

**BI-VARIANT ANALYSIS**



* There is a positive linear relationship between the age and month\_as\_customer column. As age increases the month\_as customers also increases, also the fraud reported is very less in this case.
* In the second graph we can observe the positive linear relation, as the total claim amount increases, an injury claim also increases.
* The Third plot is also the same as the second one that is as the property claim increases, the vehicle claim also increases.
* In the fourth plot we can observe the data is scattered and there is no such relation between the features.

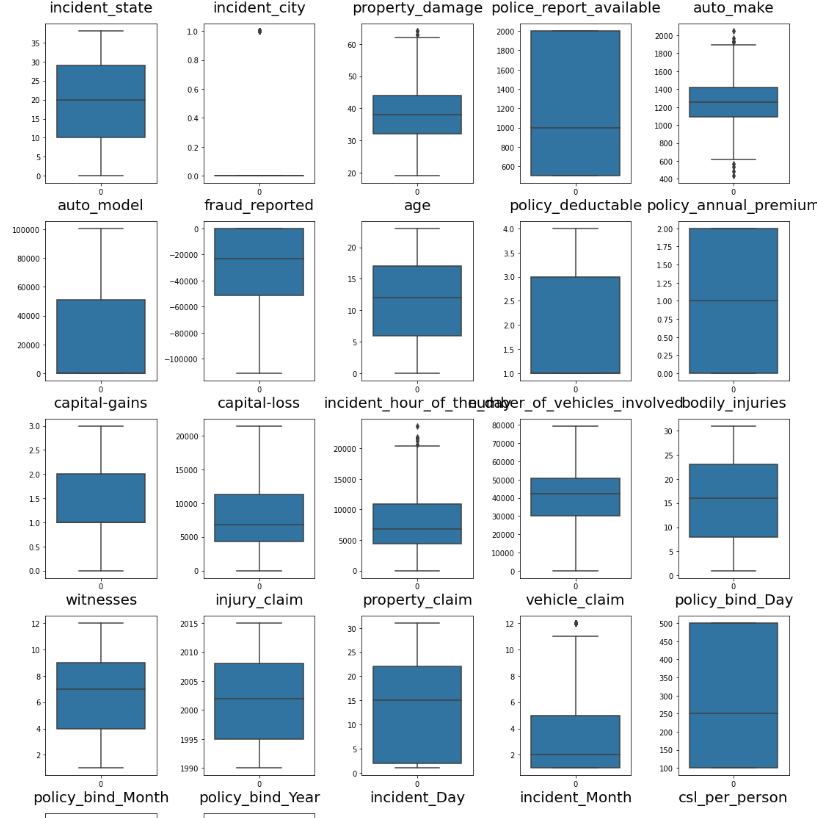
**DESCRIBE THE DATASET—STATISTIC:**

****

With the help of the statistic of the dataset we analysis that where outliers are present and where is which kind of skewness means right-side skew and left-side skew is present.

**CHECK OUTLIERS**

****

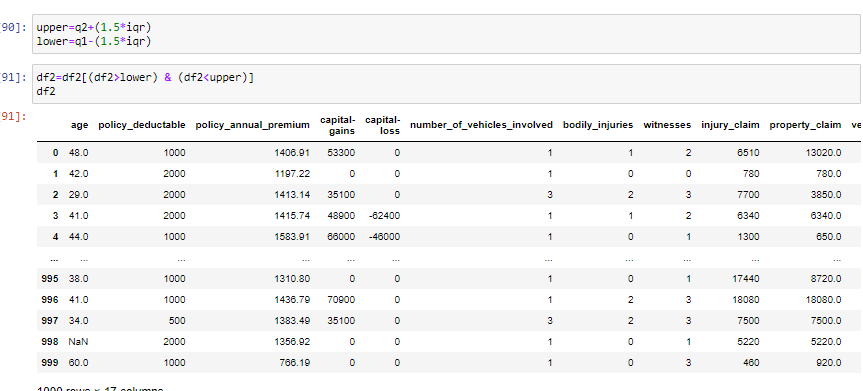
****As we can see, I have used a box plot to identify the outliers and we can find the outliers in the following columns:

* Age
* policy\_annual\_premium
* total\_claim\_amount
* property\_claim
* incident\_month

We don’t apply any transformation method on categorical and target data. So we do all analysis on only the Numerical dataset

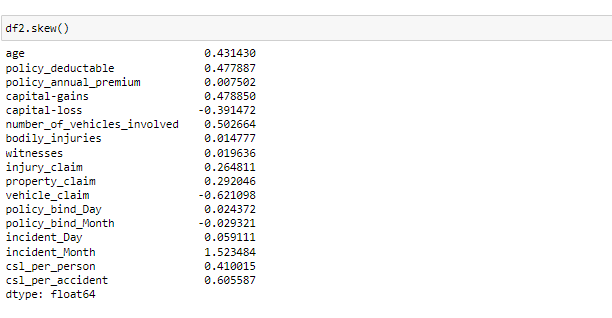
These are the numerical columns that contain outliers hence removing the outliers in these columns using the IQR Method. We don’t use the Zscore method bcoz in this some data is lost so we don’t match with the categorical and numerical datasets.

**IQR METHOD FOR OUTLIERS**



After applying the IQR method outliers are removed from my dataset. Now we go for Check skewness after removing outliers

**SKEWNESS:**

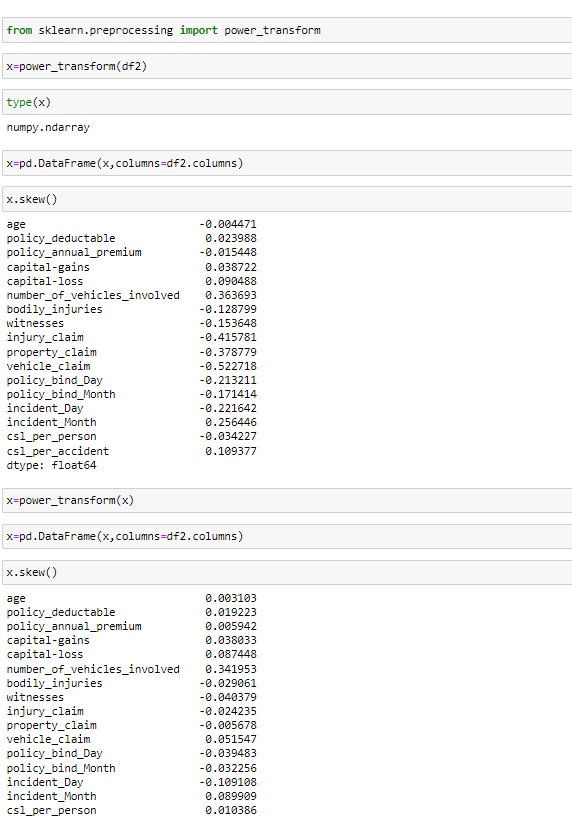
****

# If take threshold value for skewness is +/-0.5 then skewness is present in the dataset-->

1. number\_of\_vehicles\_involved
2. vehicle claim
3. incident Month
4. csl\_per\_accident

To remove skewness we go with the power transformation method

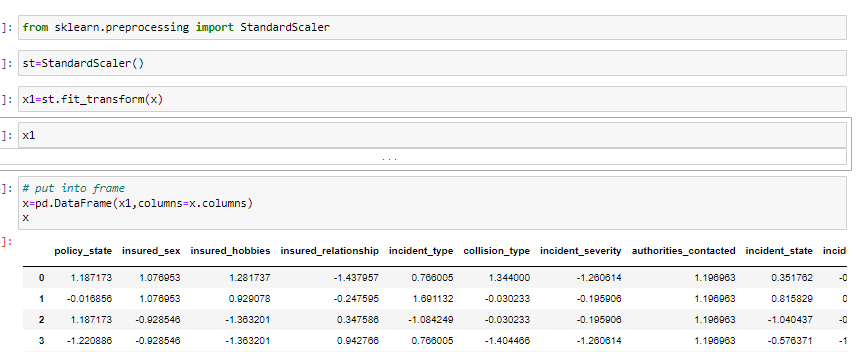
**POWER TRANSFORMATION:**

****

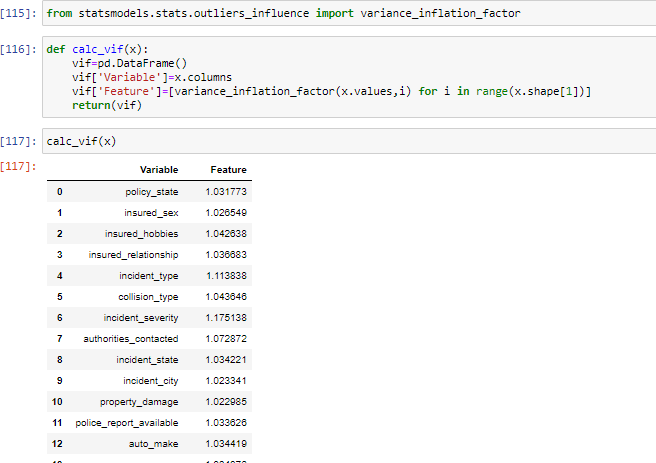
**Here in the dataset skewness** not remove once time so I apply it a second time .now skewness is present in number\_of\_vehicle\_involved and incident\_day present. I drop it elsewhere it raises an error during the prediction model.

**Scaling the DataSet**

## **Feature Scaling using Standard Scalarization**



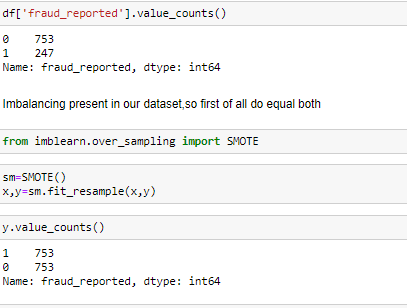
**Checking Multi-colinearity using VIF**

****

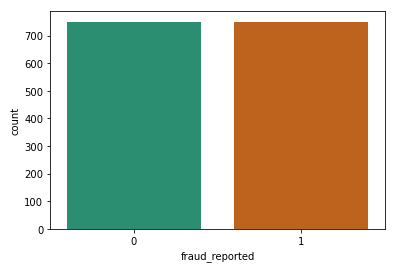
It is observed that some columns have VIF above 10 which mean they are causing multicollinearity problem. Let's drop the feature having a high VIF value amongst all the columns.

I have dropped the csl\_per\_accident features with a colinearity of more than 10, and now we have removed the problem.

We had earlier identified another problem of imbalanced data in the target variable, let’s treat it.



As we have treated the oversampling issue using SMOTE, now the data looks good.

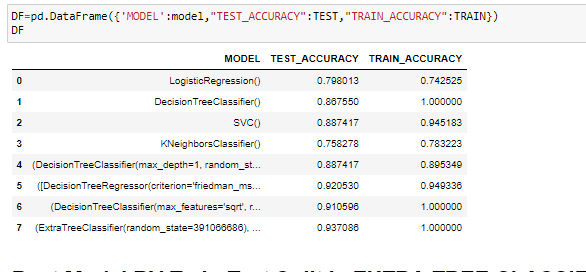


Finally, we have got into the position where we will start building the model.

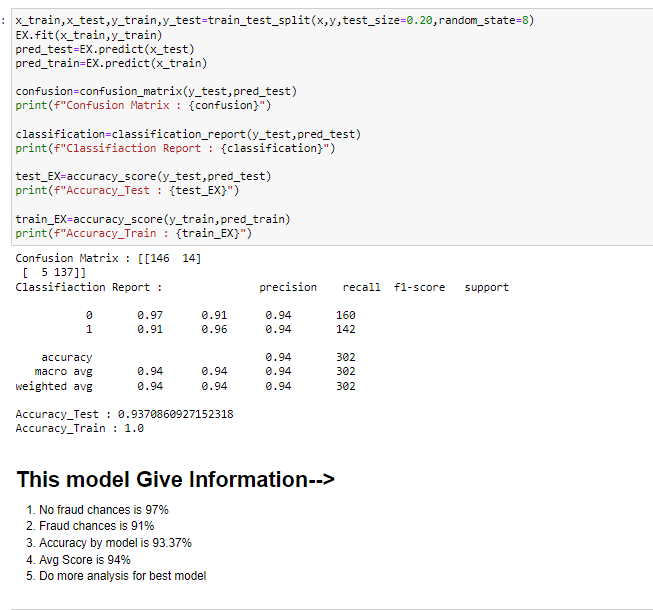
First, let’s find the best random state in which we can build the model*.*

**BEST MODEL:**

**We put a table here for finding the best model—**

****

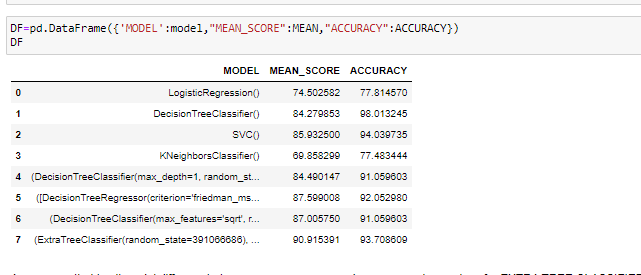
**We try many models but get the best model extra tree classifier.**

****

We get the highest accuracy and highest auc\_score 98%

Now go to 2nd method

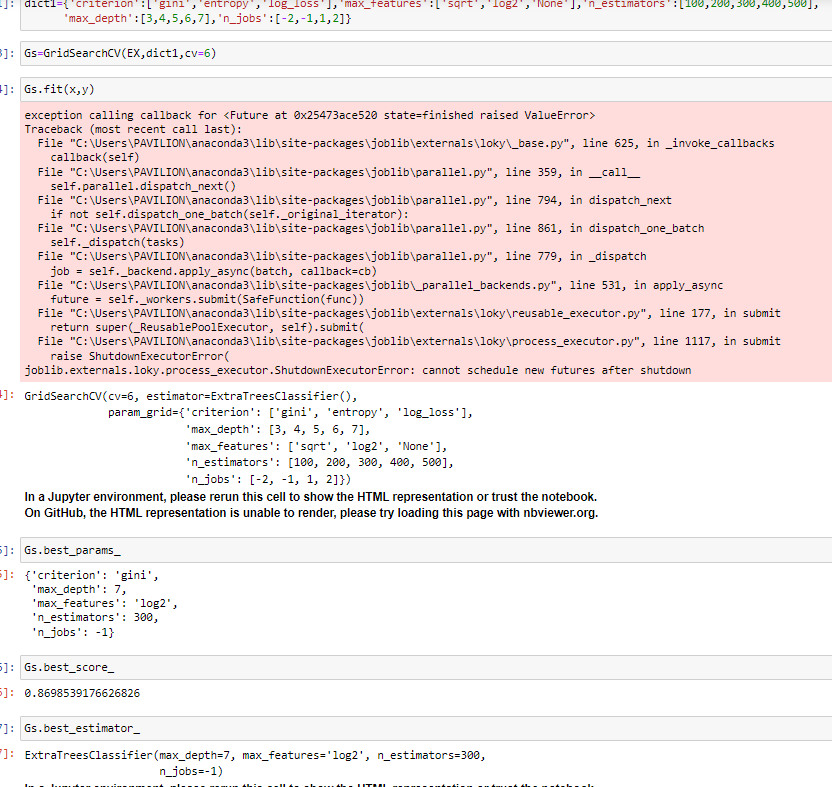
**CROSS VAL SCORE**

****

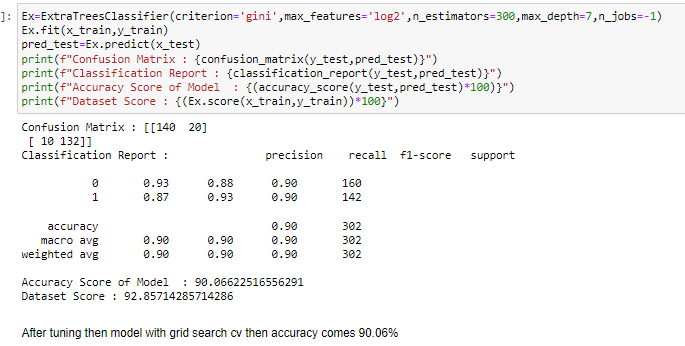
By this table, we get that the difference between the mean and accuracy score is less for the Extra tree classifier model.

After the cross-validation, we can see that ExtraTreesClassification is the best-fit model.

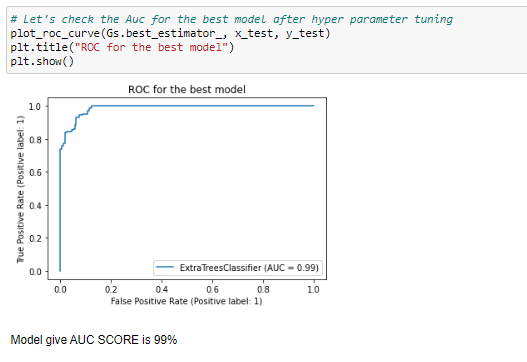
Now, that we have found the best fit model, let’s perform some HyperParameterTuning to improve the performance of the model.



Here we have got the best parameters, and we will build our final model using these parameters.

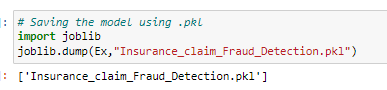


Plotting and AUCROC curve for the final model.

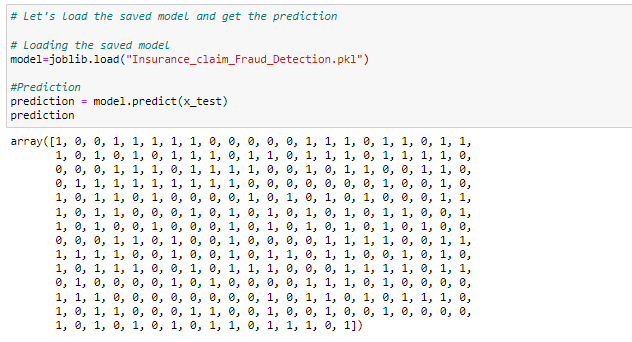


Auc-score is 99% for the model extra tree classifier.

**Saving the model**

****

**Predicting the model**

****

**CONCLUSION: At**

At the beginning of the blog have discussed the lifecycle of a Machine Learning Model, you can see how we touched base on each point and finally reached up to the model building and made the model ready for deployment.

This industry area needs a good vision of data, and in model-building, ding problem Data Analysis and Feature Engineering is the most crucial part.

You can see how we have handled numerical and categorical data and also how we build different machine-learning models on the same dataset.

Using hyperparameter tuning we can improve our model accuracy, for instance in this model the accuracy remained to need same.

Using this machine Learning Model we people can easily predwhether ict the insurance claim is fraudulent or not and we could be that applicationtion that will be considered as a fraud claim.

**THANKS 😊**