**INSURANCE CLAIM FRAUD DETECTION PROJECT**

Insurance claim fraud refers to the act of deceitfully filing an insurance claim in order to receive compensation or benefits that are not rightfully owed. This type of fraud can be committed by applicants, policyholders, third-party claimants, or even professionals who provide services to claimants. Insurance fraud can occur across various types of insurance, including but not limited to health, auto, life, property, and workers' compensation.

Types of insurance claim fraud include:

* **Exaggerating Claims**: Inflating the value of a legitimate claim to receive higher compensation.
* **Fictitious Claims**: Submitting claims for damages or losses that never occurred.
* **Staged Incidents**: Deliberately causing an accident or incident (such as a car collision or home fire) to file a claim.
* **False Documentation**: Providing forged or altered documents to support a fraudulent claim.
* **Misrepresentation**: Lying about circumstances related to the claim or failing to disclose relevant information to obtain benefits.
* **Professional Fraud**: When service providers (e.g., doctors, lawyers, auto repair shops) inflate costs, bill for services not rendered, or otherwise conspire with claimants to submit fraudulent claims.

**Prevention and Detection:**

Insurance companies invest in various measures to prevent and detect fraud. These include sophisticated analytics, data mining, and AI algorithms designed to identify suspicious patterns. Additionally, insurers conduct thorough investigations, employ specialized fraud detection units, and work closely with law enforcement. Public awareness campaigns and hotlines for reporting suspected fraud also play a critical role in prevention efforts.

Consumers can help by being vigilant about their insurance transactions, reporting suspected fraud, and understanding that insurance fraud is not a victimless crime—it affects everyone through higher costs and inefficiencies.

1. **Problem Statement:**

Insurance fraud poses a significant challenge within the industry, presenting difficulties in accurately detecting fraudulent claims. Machine Learning offers a promising solution to assist the Auto Insurance sector in overcoming this obstacle.

In this project, you will work with a dataset that encompasses insurance policy details, customer information, and specifics of the incidents leading to the claims.

**Dataset:**

In this project, you 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, 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.

Independent Variables:

months\_as\_customer: Number of months of patronage

age: the length of time a customer has lived or a thing has existed

policy\_number: It is a unique id given to the customer, to track the subscription status and other details of customer

policy\_bind\_date:date which document that is given to customer after we accept your proposal for insurance

policy\_state: This identifies who is the insured, what risks or property are covered, the policy limits, and the policy period

policy\_csl: is basically Combined Single Limit

policy\_deductable: the amount of money that a customer is responsible for paying toward an insured loss

policy\_annual\_premium: This means the amount of Regular Premium payable by the Policyholder in a Policy Year

umbrella\_limit: This means extra insurance that provides protection beyond existing limits and coverages of other policies

insured\_zip: It is the zip code where the insurance was made

insured\_sex: This refres to either of the two main categories (male and female) into which customer are divided on the basis of their reproductive functions

insured\_education\_level: This refers to the Level of education of the customer

insured\_occupation: This refers Occupation of the customer

insured\_hobbies: This refers to an activity done regularly by customer in his/her leisure time for pleasure.

insured\_relationship: This whether customer is: single; or. married; or. in a de facto relationship (that is, living together but not married); or. in a civil partnership

capital-gains: This refers to profit accrued due to insurance premium

capital-loss: This refers to the losses incurred due to insurance claims

incident\_date: This refers to the date which claims where made by customers

incident\_type: This refers to the type of claim/vehicle damage made by customer

collision\_type: This refers to the area of damage on the vehicle

incident\_severity: This refers to the extent/level of damage

authorities\_contacted: This refers to the government agencies that were contacted after damage

incident\_state: This refers to the state at which the accident happened

incident\_city: This refers to the city at which the accident happened

1ncident\_location: This refers to the location at which the accident happened

incident\_hour\_of\_the\_day: The period of the day which accident took place

number\_of\_vehicles\_involved: This refers to number of vehicles involved the accident

property\_damage: This refers to whether property was damaged or not

bodily\_injuries: This refers to injuries sustained

witnesses: This refers to the number of witnesses involved

police\_report\_available: This refers to whether the report on damage was documented or not

total\_claim\_amount: This refers to the financial implications involved in claims

injury\_claim: This refers to physical injuries sustained

property\_claim: This refers to property damages during incident

vehicle\_claim: This refers to property damages during incident

auto\_make: This refers to the make of the vehicle

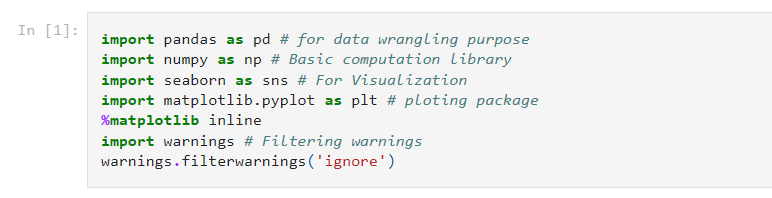
auto\_model: This refers to the model of the vehicle

auto\_year: This refers to the year which the vehicle was manufactured

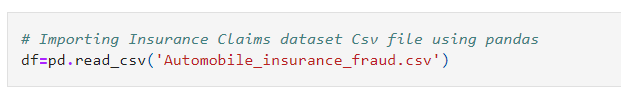
\_c39: fraud\_reported

**2. Data Analysis:**

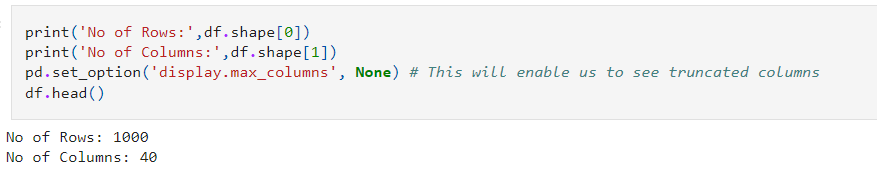
**A) Importing require library for performing EDA, Data Wrangling and data cleaning**



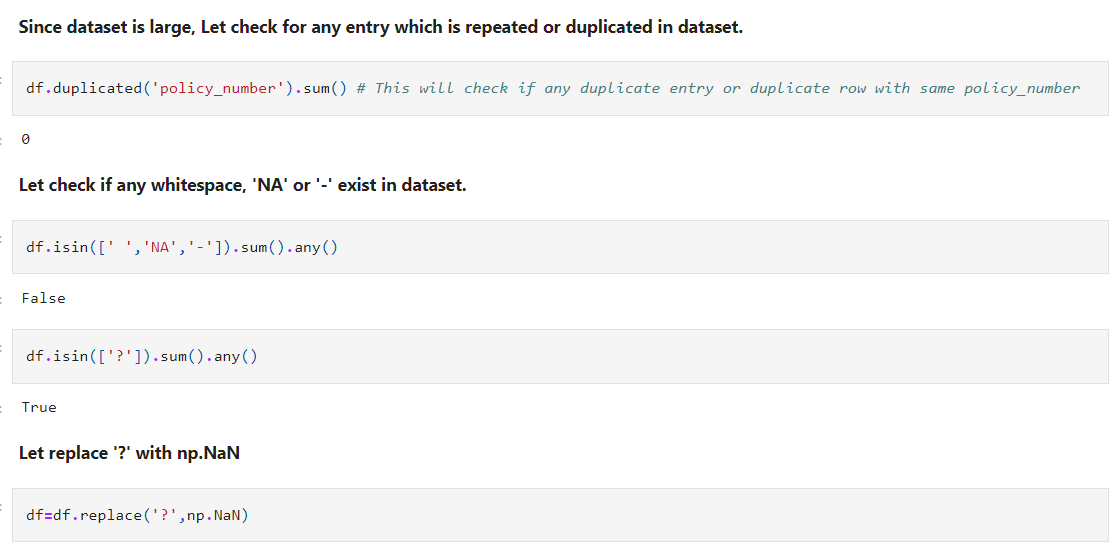
**B) Importing dataset of .csv format**



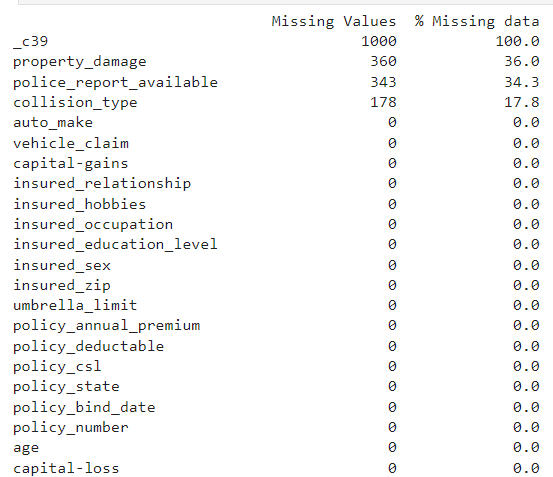
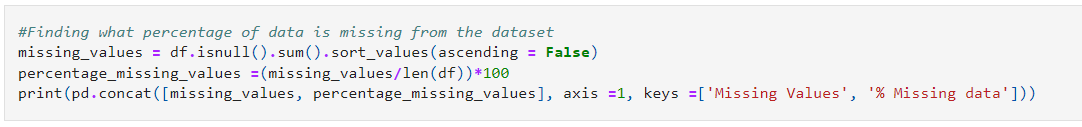
**C) Shape of the dataset to know number of rows and columns of the datasets.**

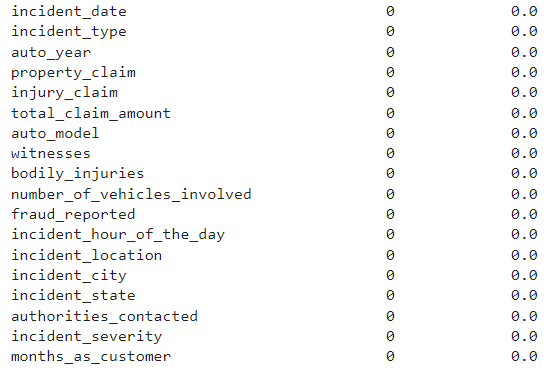


**D) Checked if data has duplicate entries**



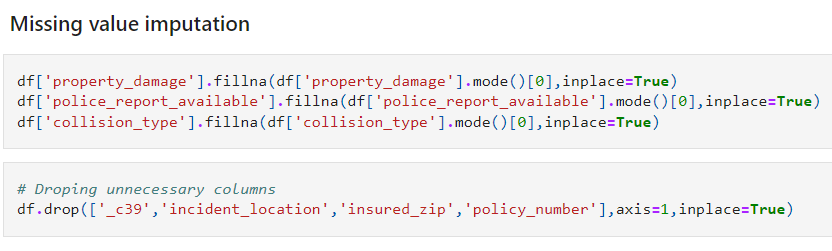
**E) Checking for missing values**





* 'property\_damage','police\_report\_available','collision\_type' contain missing values.
* c39 columns with 100 % null value. We are going drop it.
* Other missing value feature are categorical in nature. We gone impute them with mode of that particular category.

**F) Missing value imputation and dropping the unnecessary column**

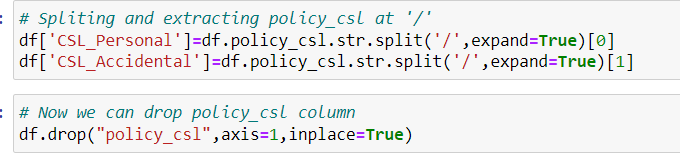


**3. EDA Concluding Remarks:**

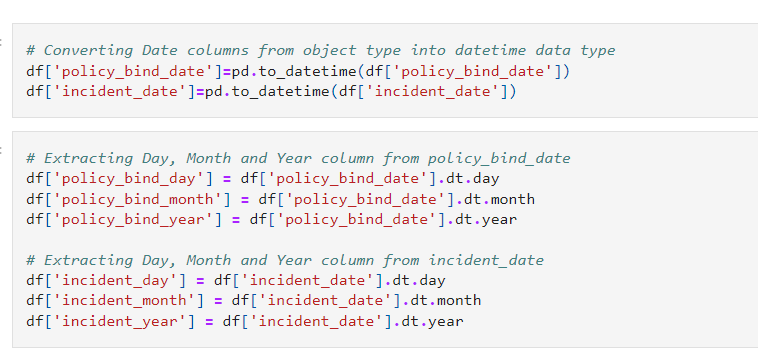
**A) Univariant Analysis**

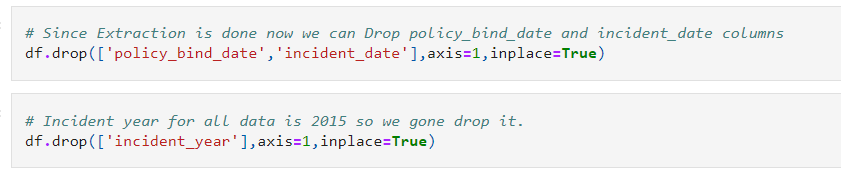
**Policy\_csl column :**

* Combined single limit (CSL):CSL is a single number that describes the predetermined limit for the combined total of the Bodily Injury Liability coverage and Property Damage Liability coverage per occurrence or accident.
* In this dataset Policy\_csl columns have numerical data separated by '/', resulting into object datatype.
* We will split this column into two CSL columns for person and accident.



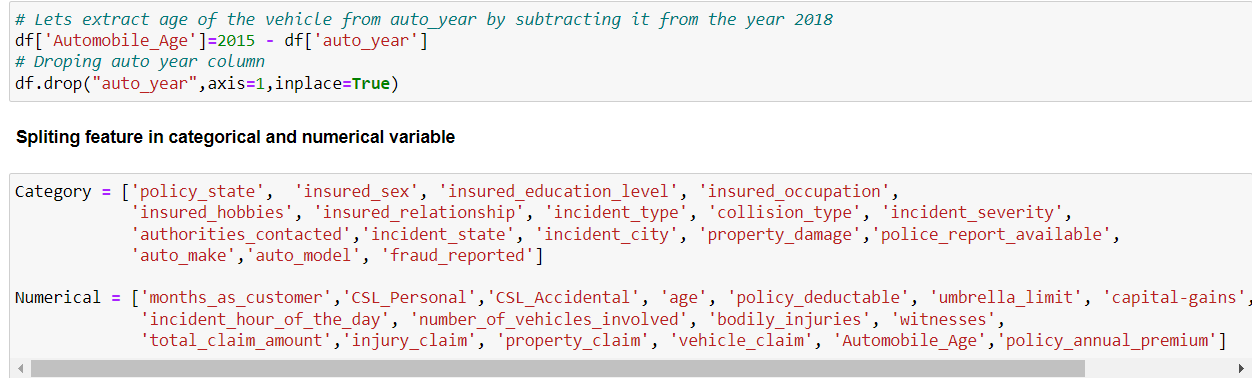
We have two features here with datetime datatypes. We are going to split them in terms of date, month and year.

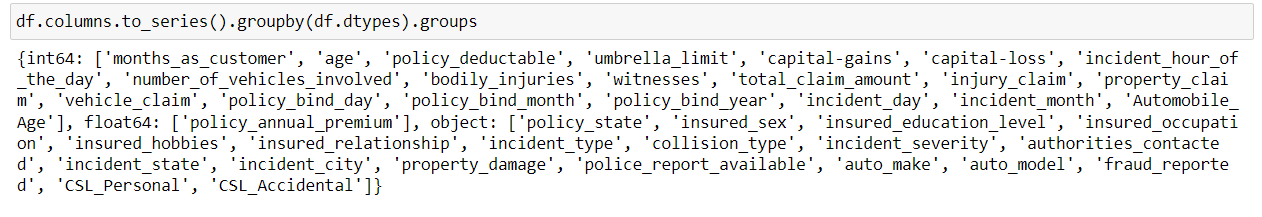




We have one column name as 'auto\_year' which depict year from which Automobile in operational on road. In simple word Automobile age.

Here we will do some simple feature engineering to create new columns with automobile age and drop eariler.As incident year is 2015, we will use 2015 as base year for new column creation.



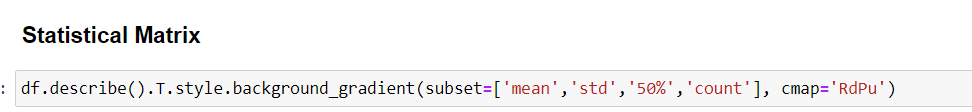
In above code groups the columns of a DataFrame (df) by their data types. Here's a breakdown of how it works:

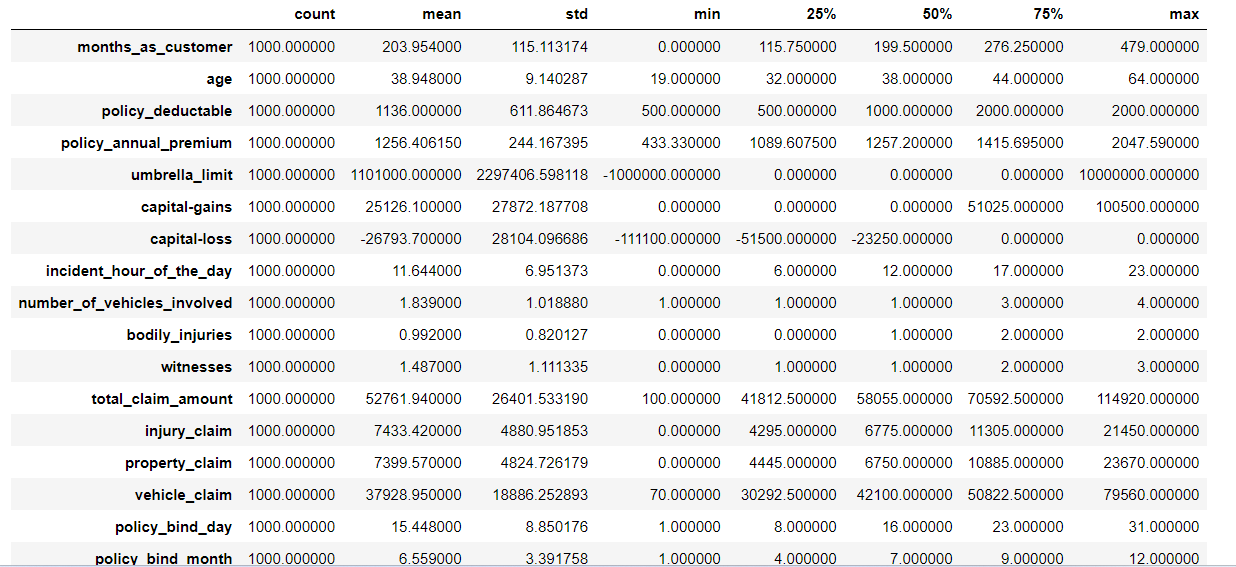
* df.columns: This retrieves the column names of the DataFrame df.
* .to\_series(): Converts the index object containing the column names into a pandas Series, where each column name becomes a value in the Series. The Series index in this case is equivalent to the original column names.
* .groupby(df.dtypes): This groups the Series by the data types of the columns in df. The df.dtypes returns a Series with index as column names and values as their respective data types. By grouping by df.dtypes, you're effectively grouping the column names based on their data types.
* .groups: This attribute of the groupby object gives you the groups as a dictionary. The keys of this dictionary are the data types, and the values are lists (actually, Index objects) of column names that correspond to each data type.

As a result, this code snippet outputs a dictionary where each key is a data type present in the DataFrame, and the associated value is a list of column names that have that data type.

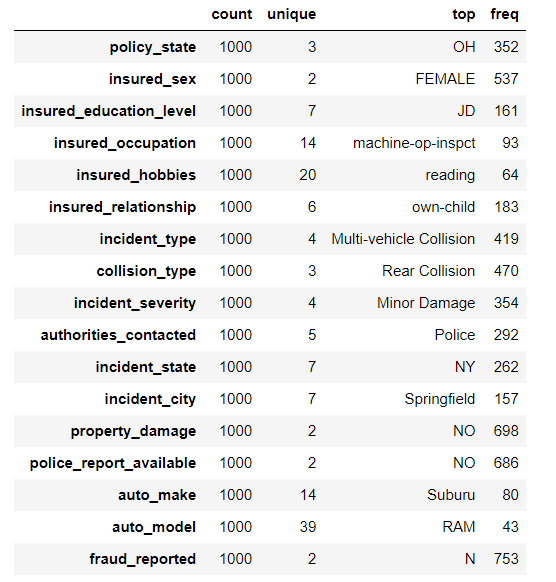
This can be useful for quickly identifying which columns are numerical, categorical, datetime, or any other data type, facilitating tasks such as data preprocessing or feature selection based on data type criteria.

**B) Bivariant and multivariant Analysis:**

It gives summary of the dataset.



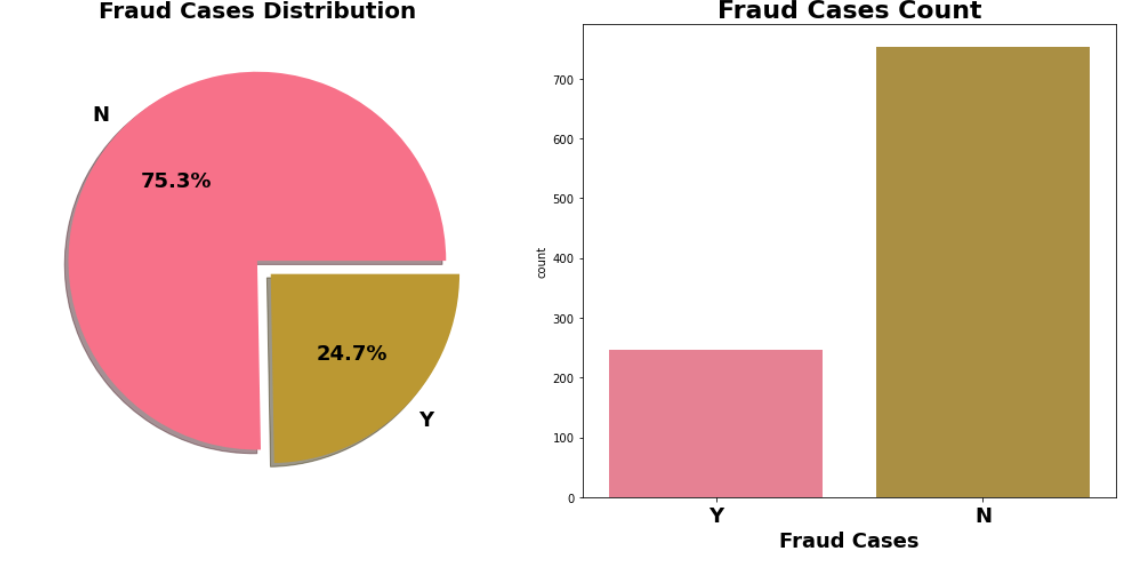




* This aimed at providing a descriptive statistical summary of a DataFrame column or columns grouped by a certain 'Category', with an added visual touch using a color gradient. Here's a detailed breakdown:
* df[Category]: This part of the code is supposed to select a specific column or set of columns from the DataFrame df. However, for it to work correctly, 'Category' should be replaced with the actual name of the column you're interested in, or it should be a list of column names. If 'Category' is intended to filter or group data, you would typically need a different approach, such as groupby.
* .describe(): This function generates descriptive statistics that summarize the central tendency, dispersion, and shape of the dataset’s distribution, excluding NaN values. It's particularly useful for getting a quick overview of numerical data but can be applied to objects for summaries of categorical data as well.
* .T: The transpose of the DataFrame. .describe() returns a DataFrame where the rows correspond to statistical measures (like mean, std, etc.) and the columns are your data's features. Transposing (.T) switches the rows and columns, which can make the table easier to read, especially if you have many features.
* .style.background\_gradient(cmap='summer\_r'): This part applies a stylistic element to the DataFrame output. It uses a background gradient coloring, making it easier to visually analyze the data. The coloring is based on the values in each column. The cmap='summer\_r' parameter specifies the colormap (in this case, 'summer\_r', which is a reversed version of the 'summer' colormap, meaning it goes from green to yellow).

**Start Exploring the target variable.**

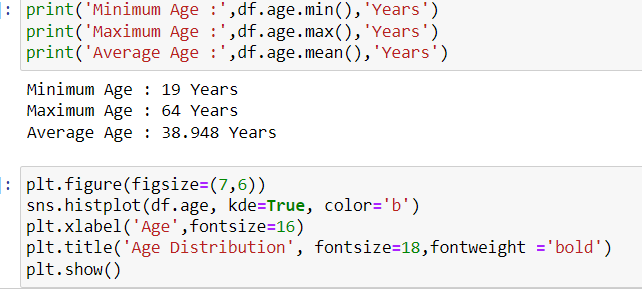


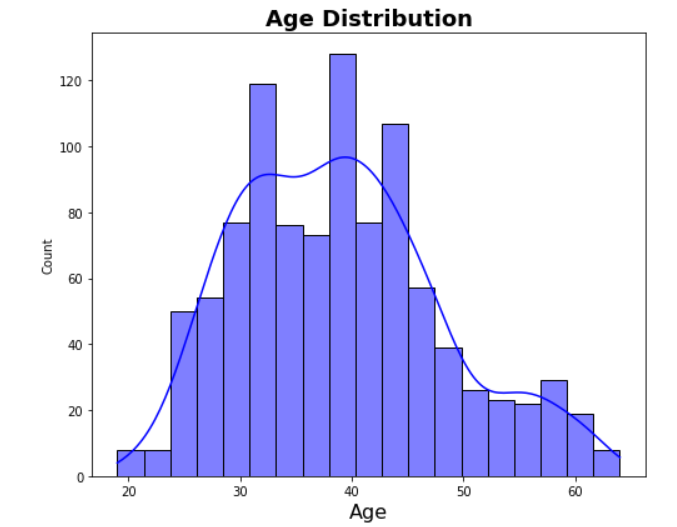


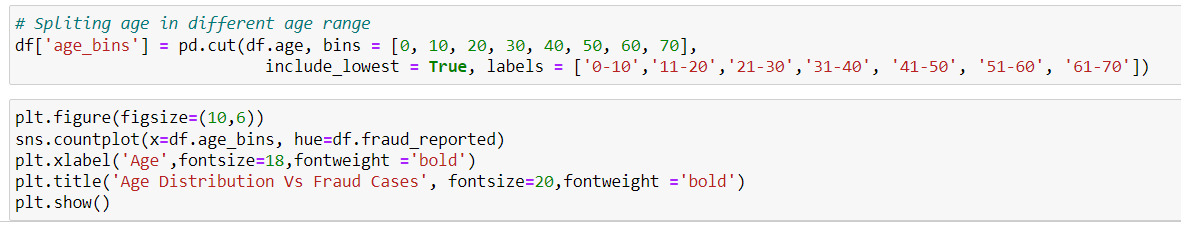
* Out of all cases around 24.7 % cases are Fraud.
* 'fraud\_reported' is our target variable to be predicted. From count plot we can say dataset is imbalanced in nature.

**Start Exploring target variable against independent features to gain more insight.**

1. **Analysing Age vs Fraud**



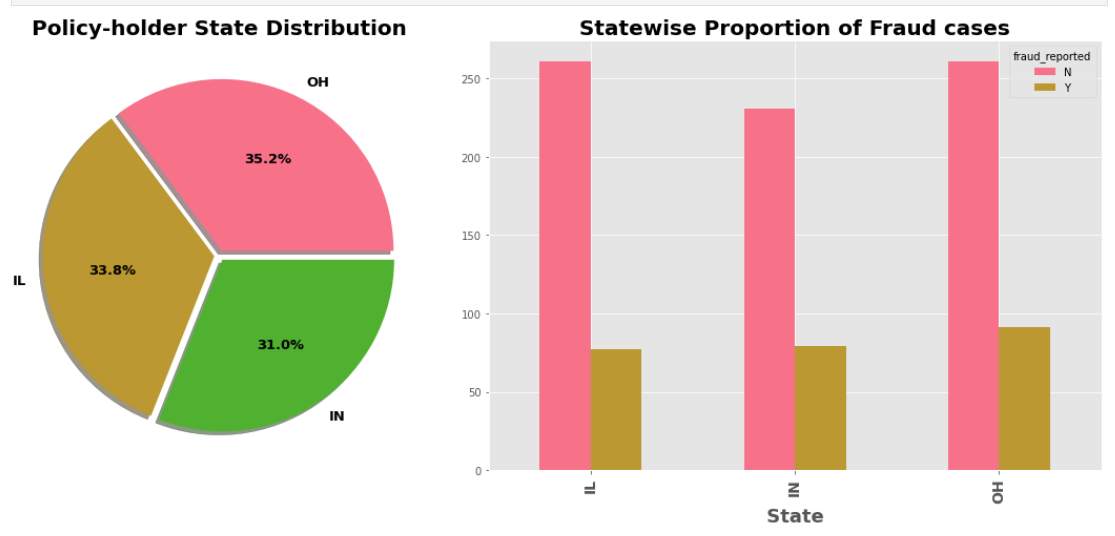




* Maximum fraud cases comes from people with age group of 31-50 year.
* Very few cases in 60+ year old peoples.

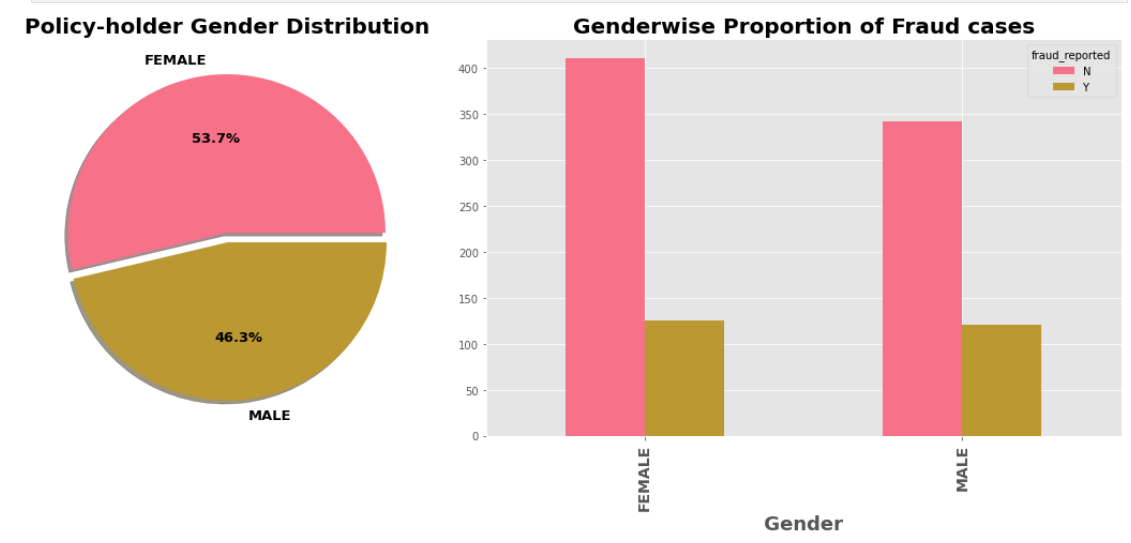
1. **Policy State Vs Fraud case**





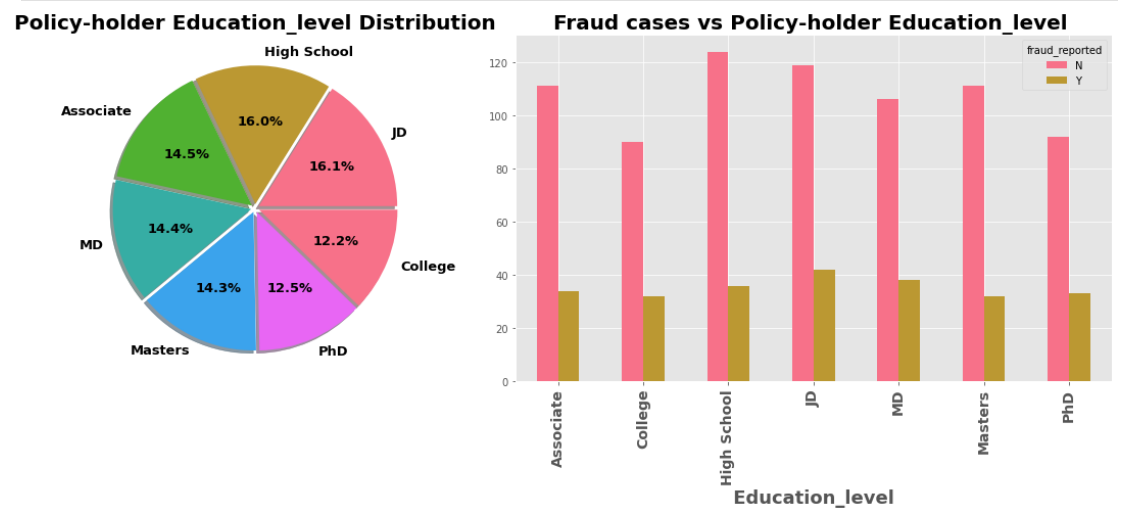
* Almost same amout of cases come from each state.
* Maximum fraud cases come from state of Ohio.

1. **Insured Gender VS Fraud cases**

* 
* Number of claims come from female is higher than which reported by male insured.
* Almost same amount of fraud cases comes from same gender.

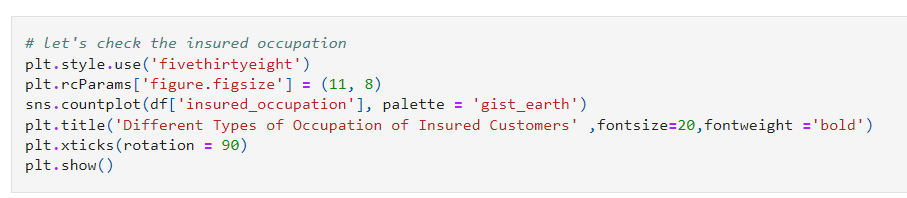
1. **Education\_level vs Fraud cases**

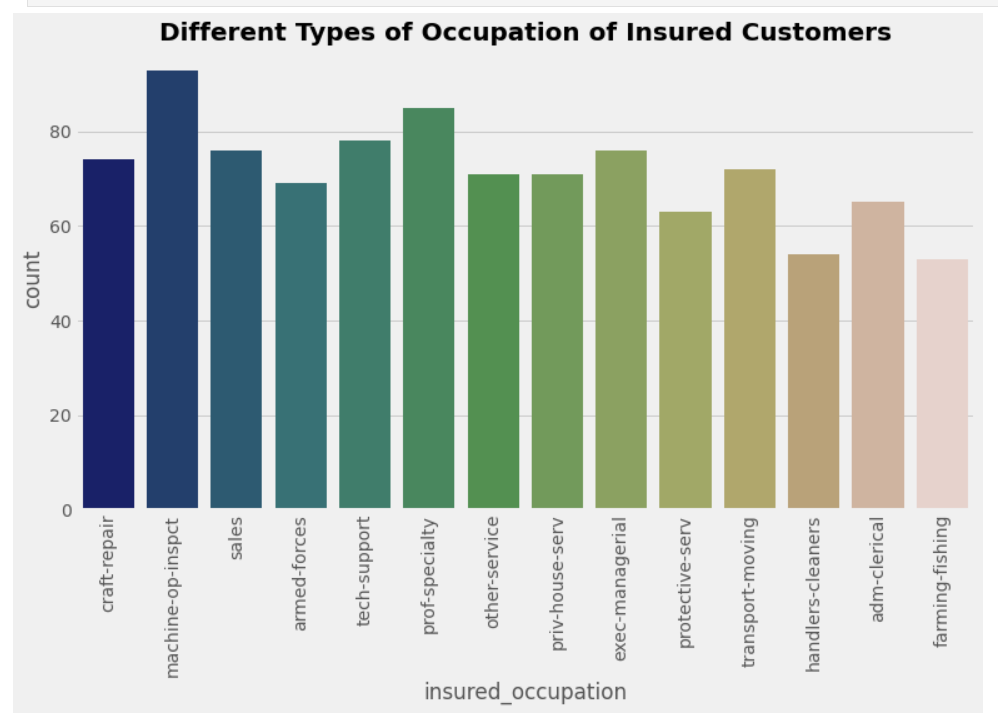




* **We can see a tendency to make fraud claims across every education background, even in Masters, PhD.**
* **Education Level is not much important variable for us.**

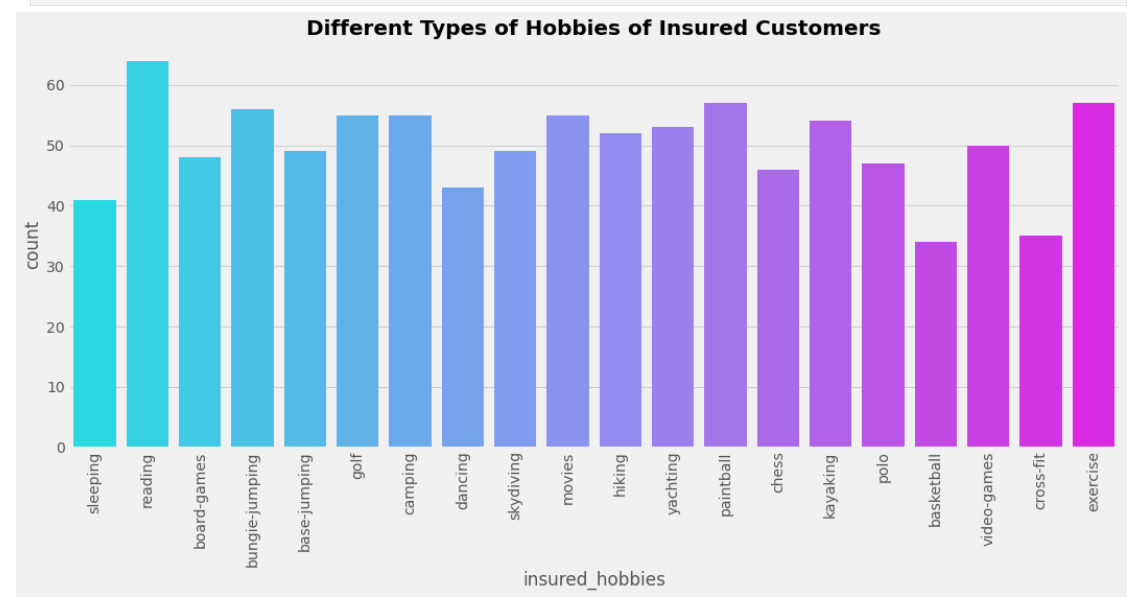
1. **Occupation of Insured Customers**



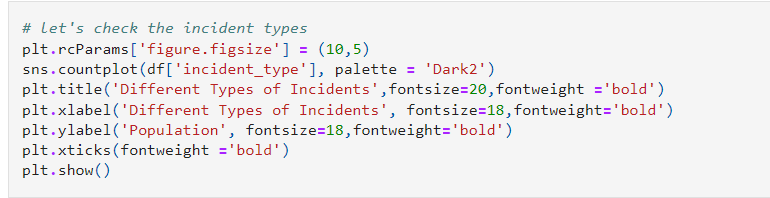


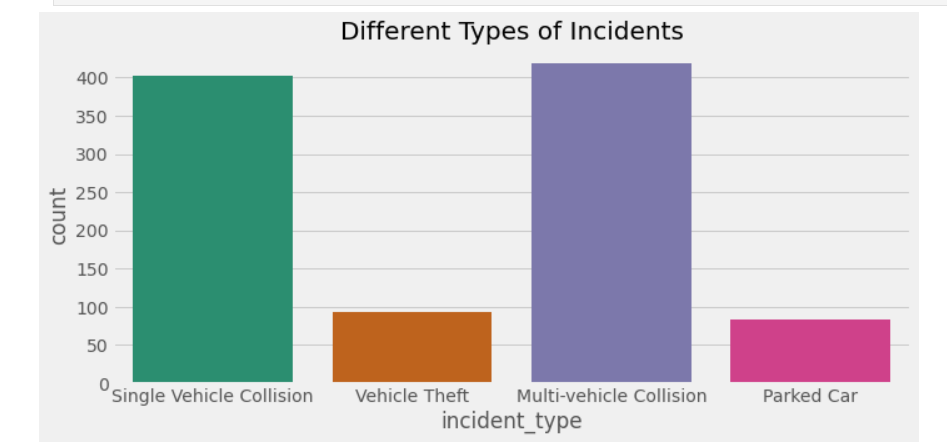
1. Hobbies of Insured Customers



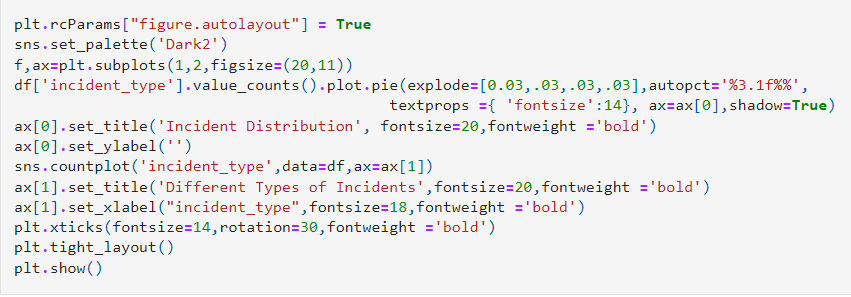


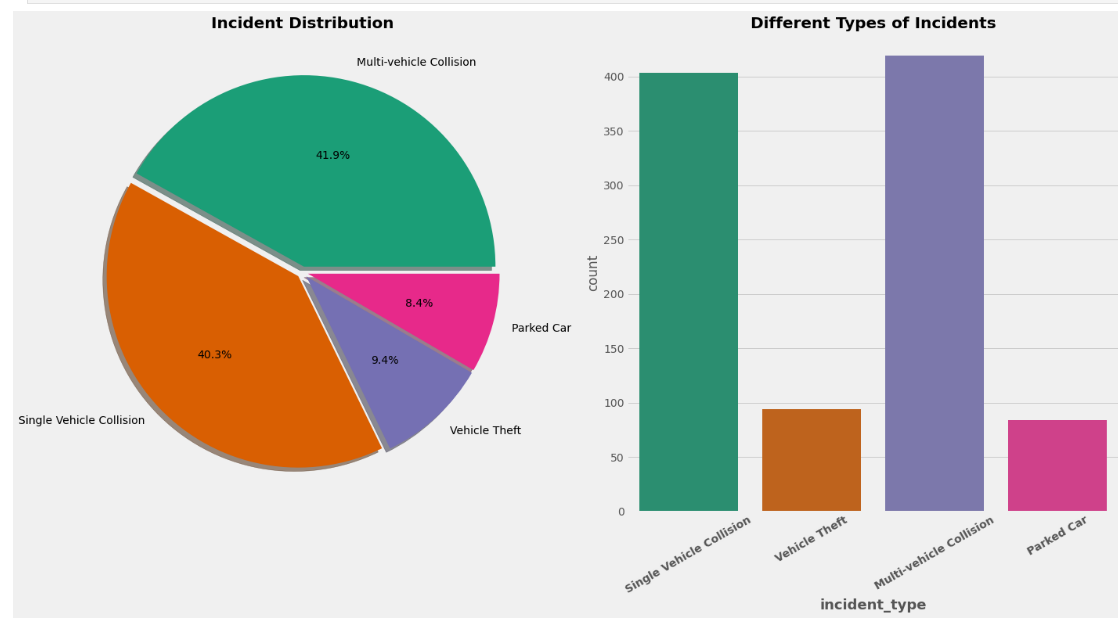
1. Different Types of Incidents Vs Fraud cases





Single vehicle collision and multi-vehicle collision incidents happened more than vehicle theft incidents and parked car

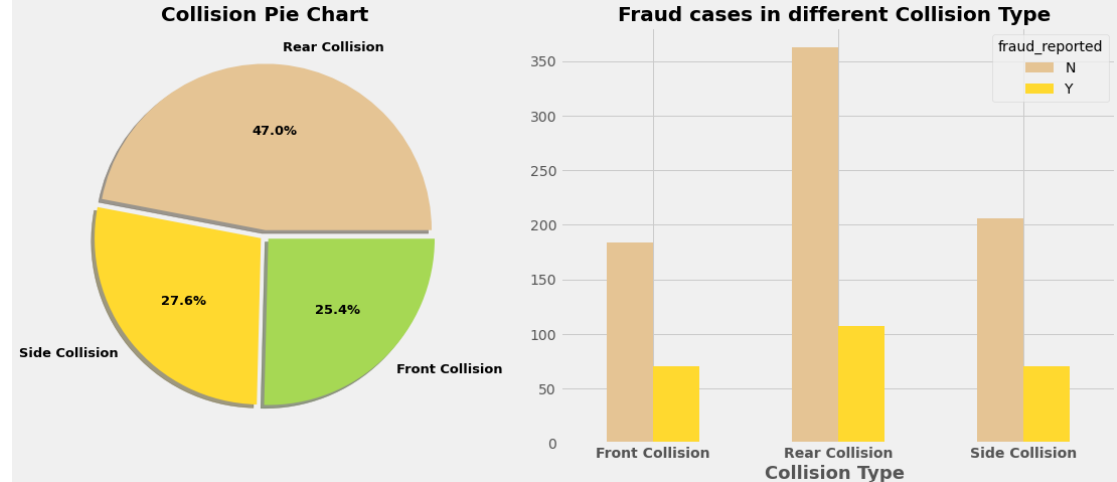


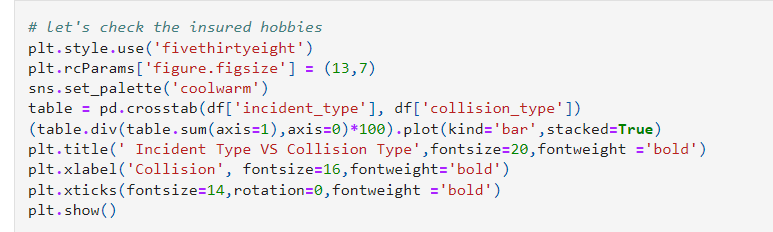


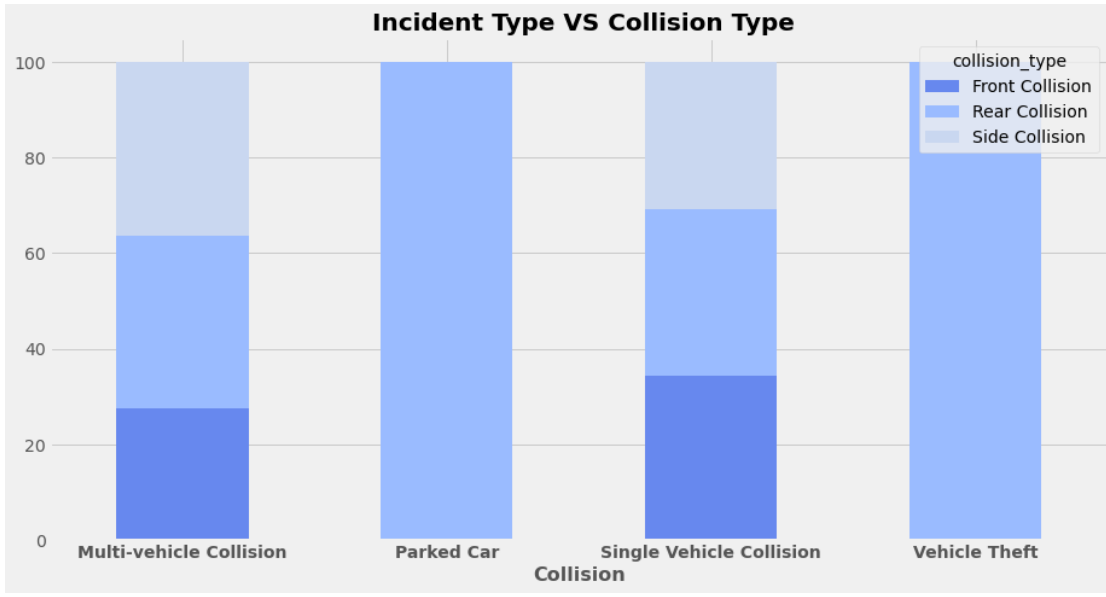
* Most of case comes from Multi-vehicle and single vehicle collision.
* Some claims are due to automobile robbery.
* **One claim out of three claim is fraud in multi or single vehicle collision incident.**

***It will be interesting to figure out collision type and severity for different incident and corresponding fraud claim.***

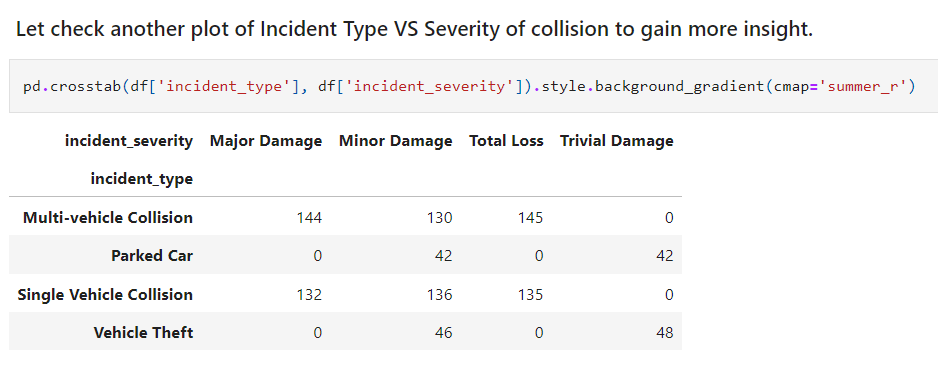
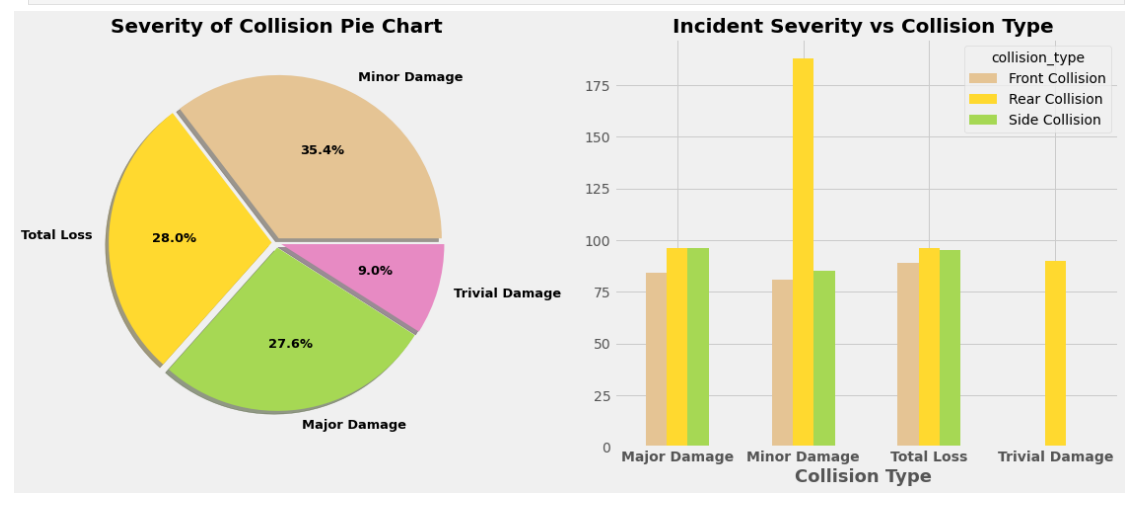
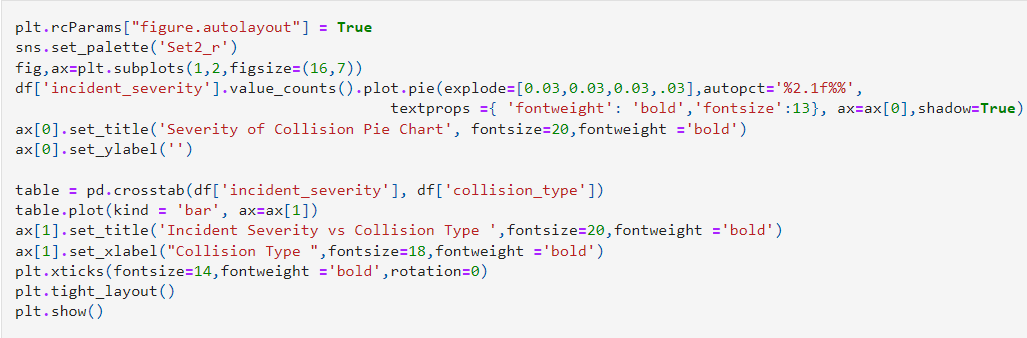
1. Exploration of different Collision

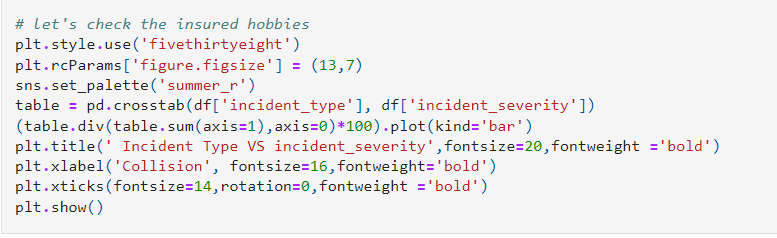






1. Collision VS Incident Severity





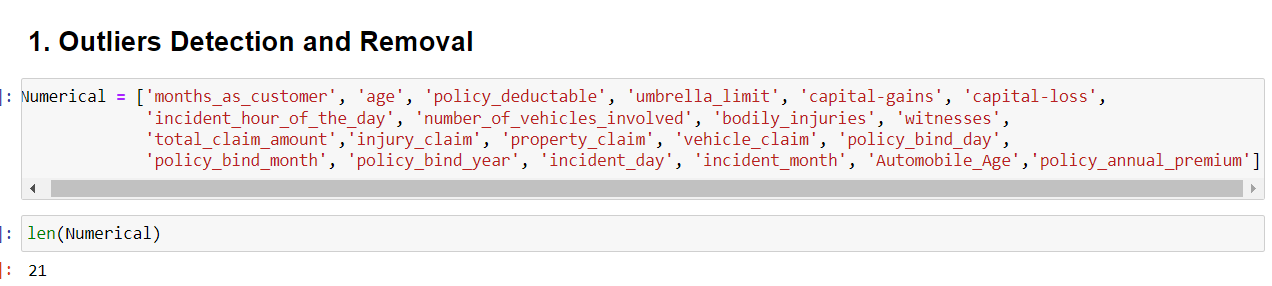
### 

### It seems like incident Type is an important variable for us.

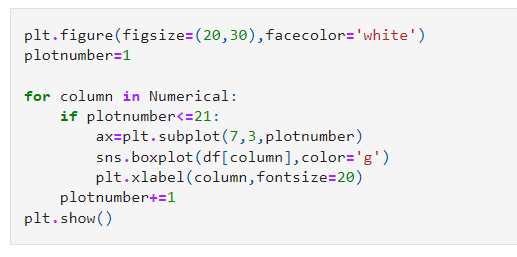
### Now is time to dive deep to get more insight on incident type by visualize incident type with Numerical features.

**4. Pre-processing Pipeline:**

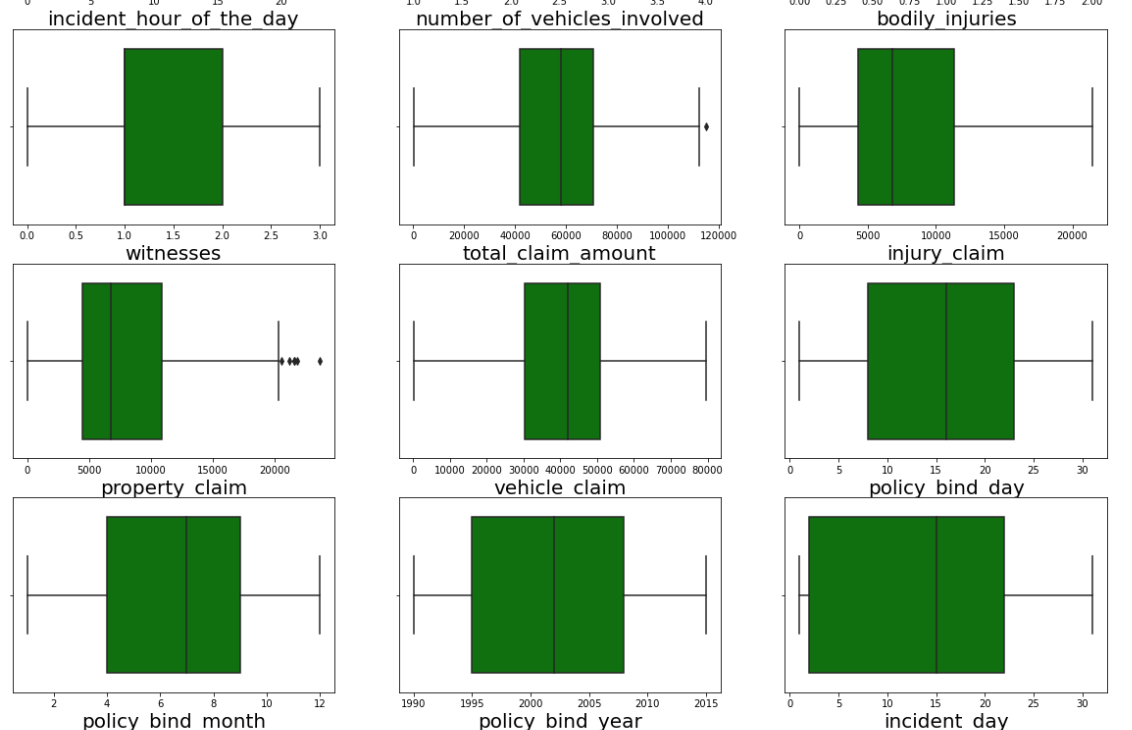
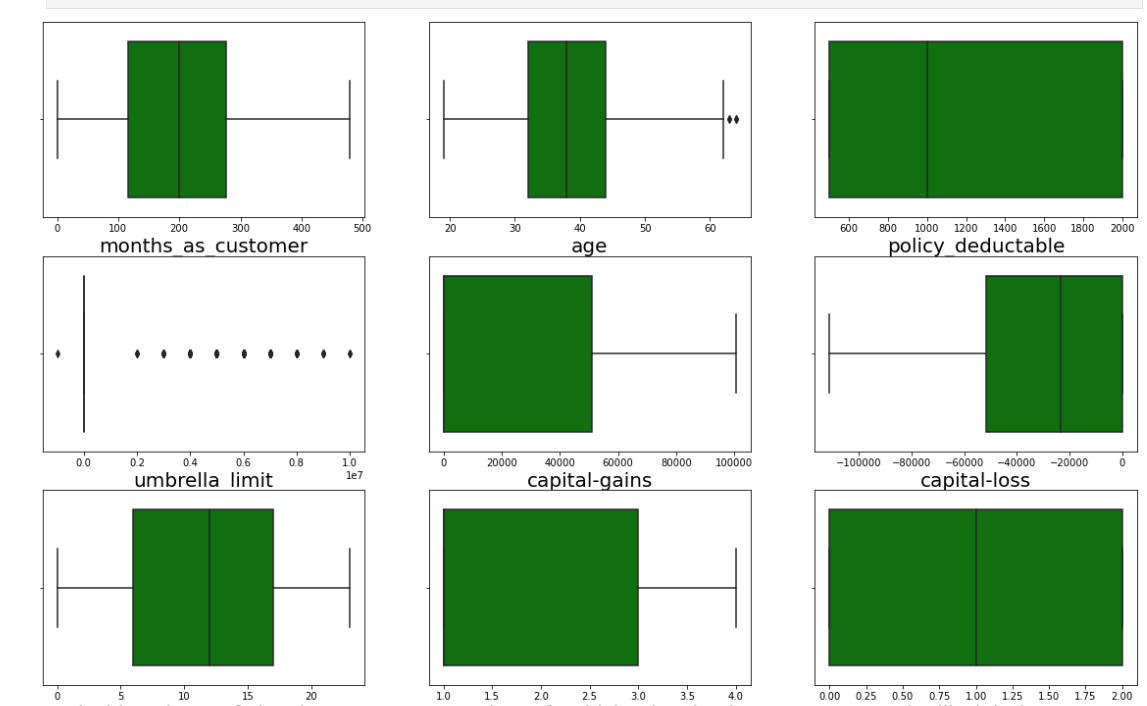
# **Feature selection and Engineering**

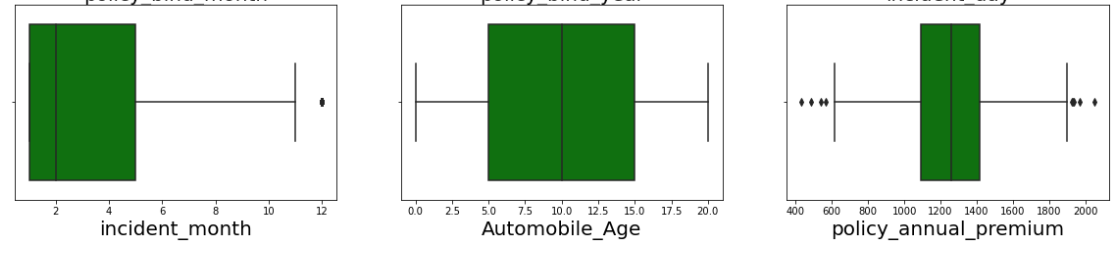


Outlier detection and removal is a crucial step in the preprocessing phase of a data science or machine learning project. Outliers are data points that deviate significantly from the rest of the data, and they can arise due to various reasons such as measurement errors, data entry errors, or genuine variations in the dataset. Not handling outliers appropriately can lead to skewed results and affect the performance of machine learning models.



* A boxplot, also known as a box-and-whisker plot, is a graphical representation used in descriptive statistics that shows the distribution of a dataset. It visualizes five summary statistics (the minimum, first quartile (Q1), median, third quartile (Q3), and maximum), outliers, and the spread of the data. Boxplots are particularly useful for indicating whether a distribution is skewed and whether there are potential unusual observations (outliers) in the data set. it visually and statistically helps identify data points that deviate significantly from the rest of the dataset. This method is based on the interquartile range (IQR), which is the difference between the third quartile (Q3) and the first quartile (Q1) in a dataset. Here's how outliers are detected using a boxplot:
* **Calculate the IQR**: The IQR is calculated by subtracting Q1 from Q3. This range represents the middle 50% of the data.
* **Determine the Whiskers**: The lower whisker is typically set at Q1 - 1.5 \* IQR, and the upper whisker is set at Q3 + 1.5 \* IQR. However, the actual data points closest to these calculated values within the dataset are used as the whiskers' ends. Some variations use a multiplier other than 1.5, depending on how stringent you want to be about defining an outlier.
* **Identify Outliers**: Data points that fall below the lower whisker or above the upper whisker are considered outliers.



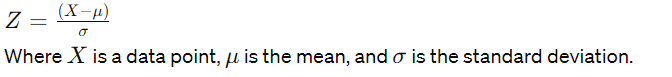
**From Boxplot we can see outliers exist dataset.**

**Outlier Removal using z-score method:**

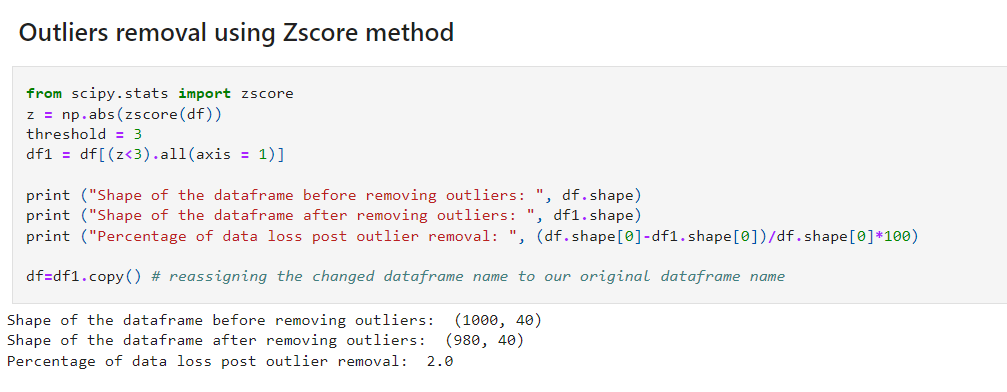
Outlier removal using the Z-score method is based on the Z-score, which measures how many standard deviations an element is from the mean. This approach is especially useful for datasets that follow a Gaussian (normal) distribution, but it can be applied more broadly with caution.

### **Steps for Outlier Removal Using Z-score:**

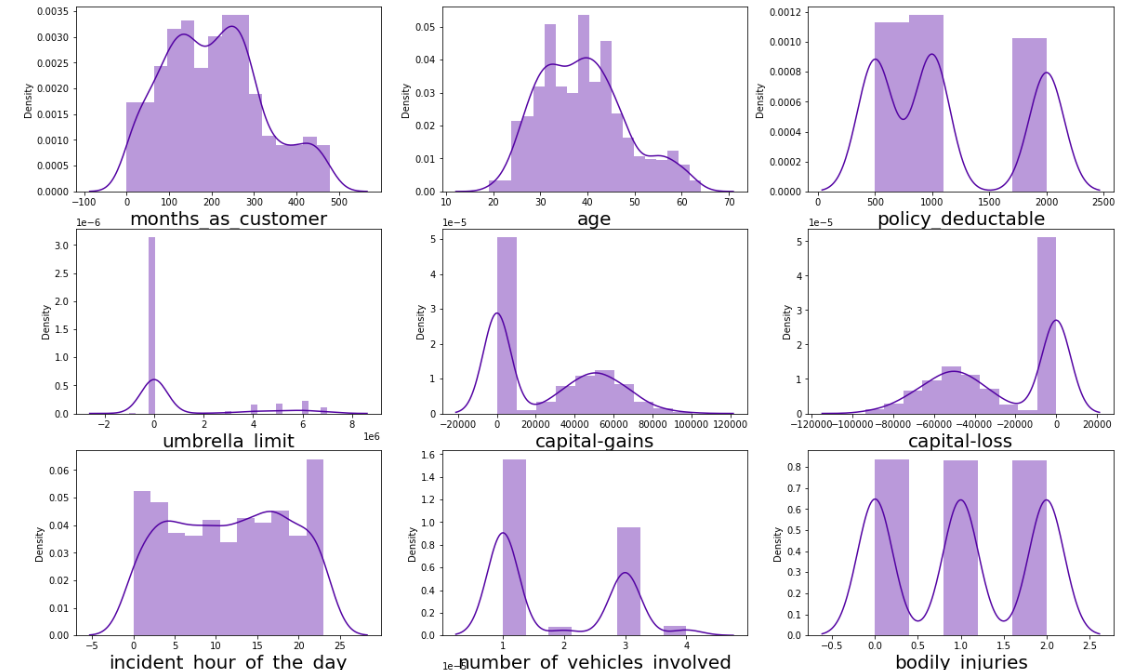
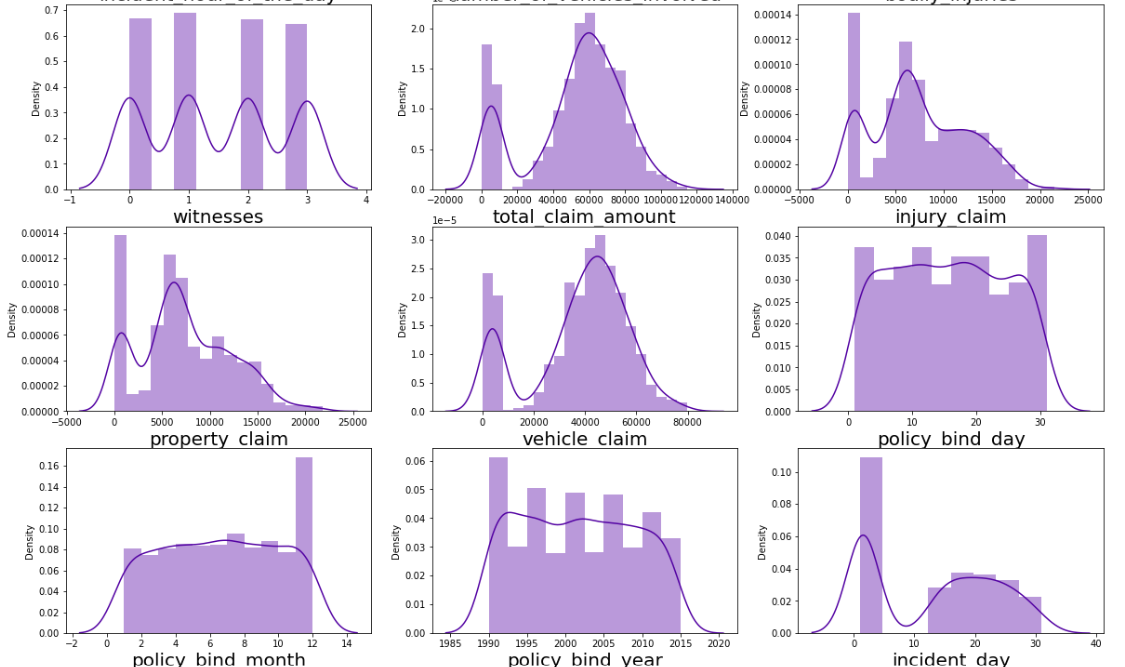
* **Calculate the Mean and Standard Deviation**: First, compute the mean (average) and standard deviation of the dataset. The mean provides a measure of central tendency, while the standard deviation measures the dispersion or variability.
* **Calculate the Z-score for Each Data Point**: The Z-score is calculated using the formula:

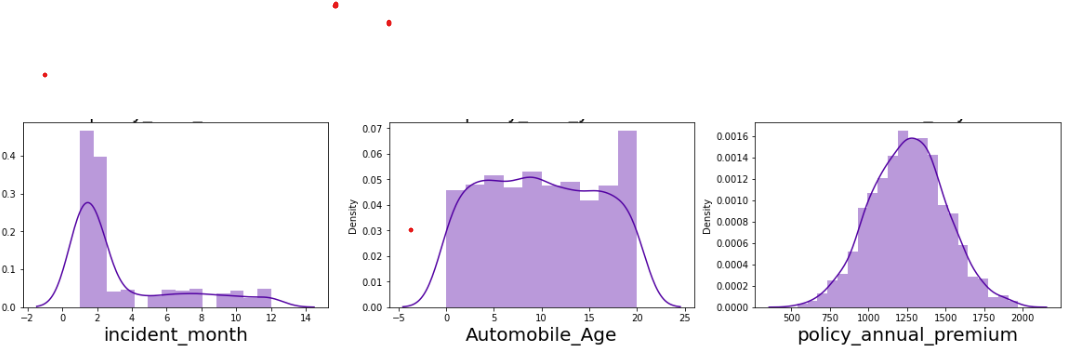


* **Define a Threshold**: A common threshold for identifying an outlier is a Z-score of 3 or -3, meaning any data point more than 3 standard deviations away from the mean is considered an outlier. However, this threshold can be adjusted based on the specific requirements of your analysis.
* **Filter Out Outliers**: Remove data points that have a Z-score beyond the threshold.

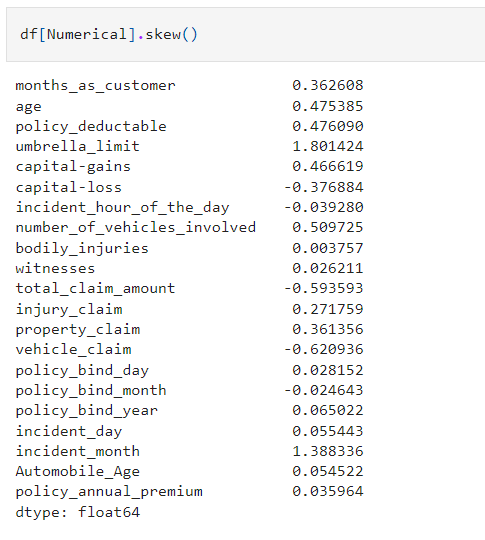




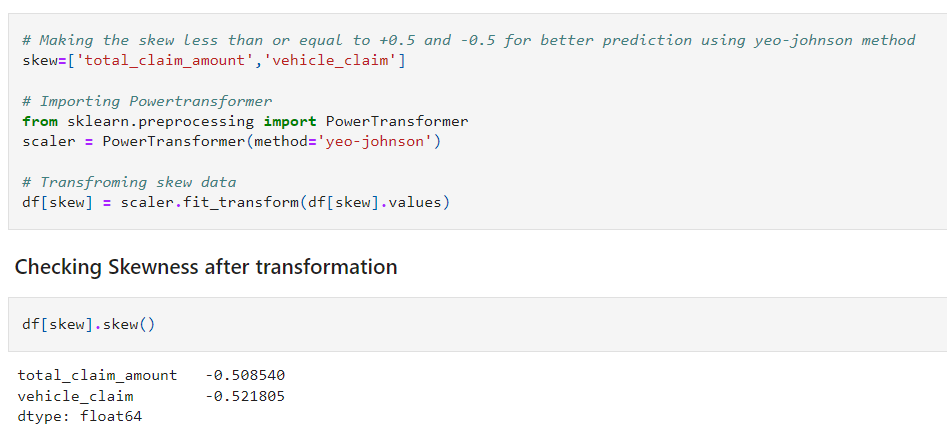


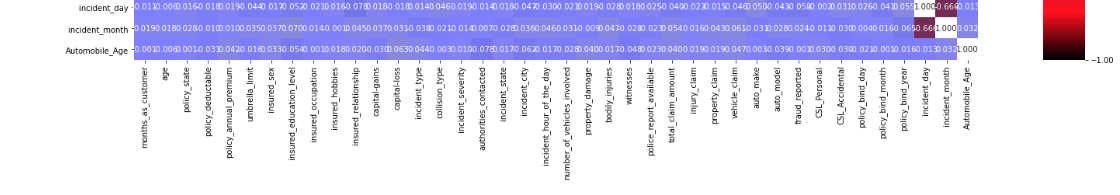
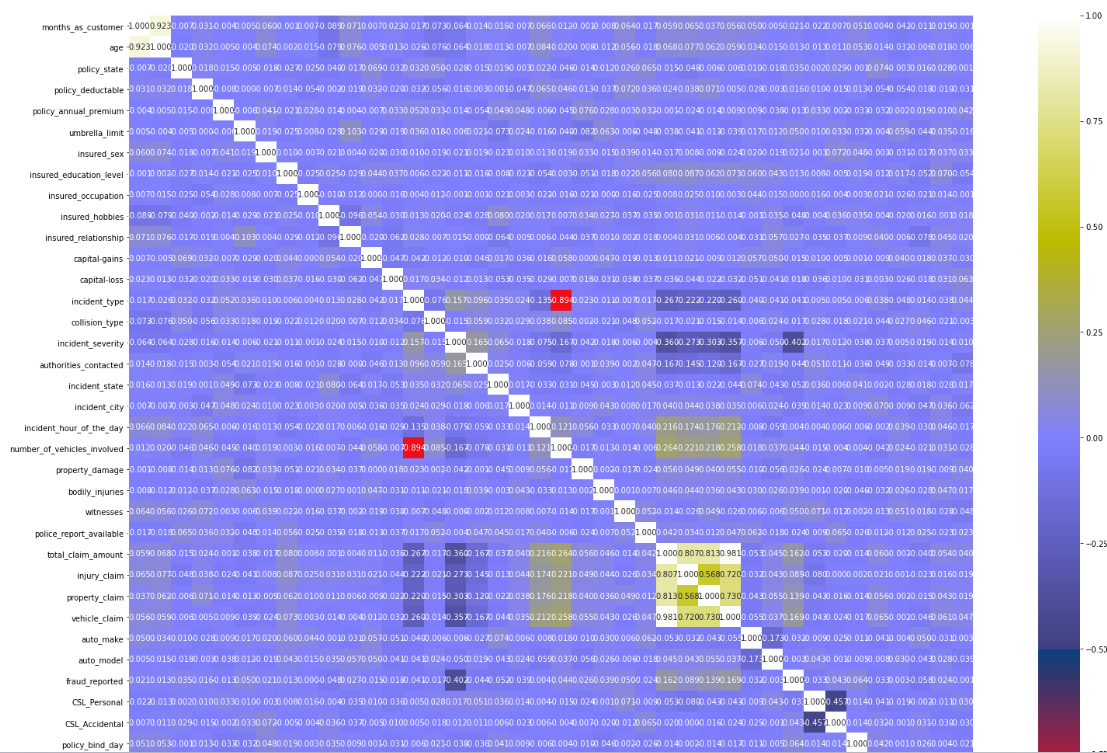
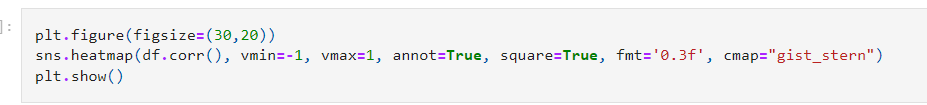
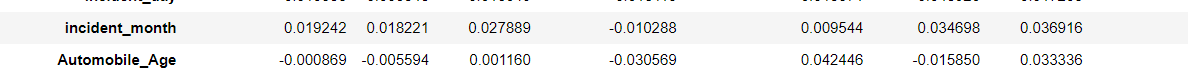
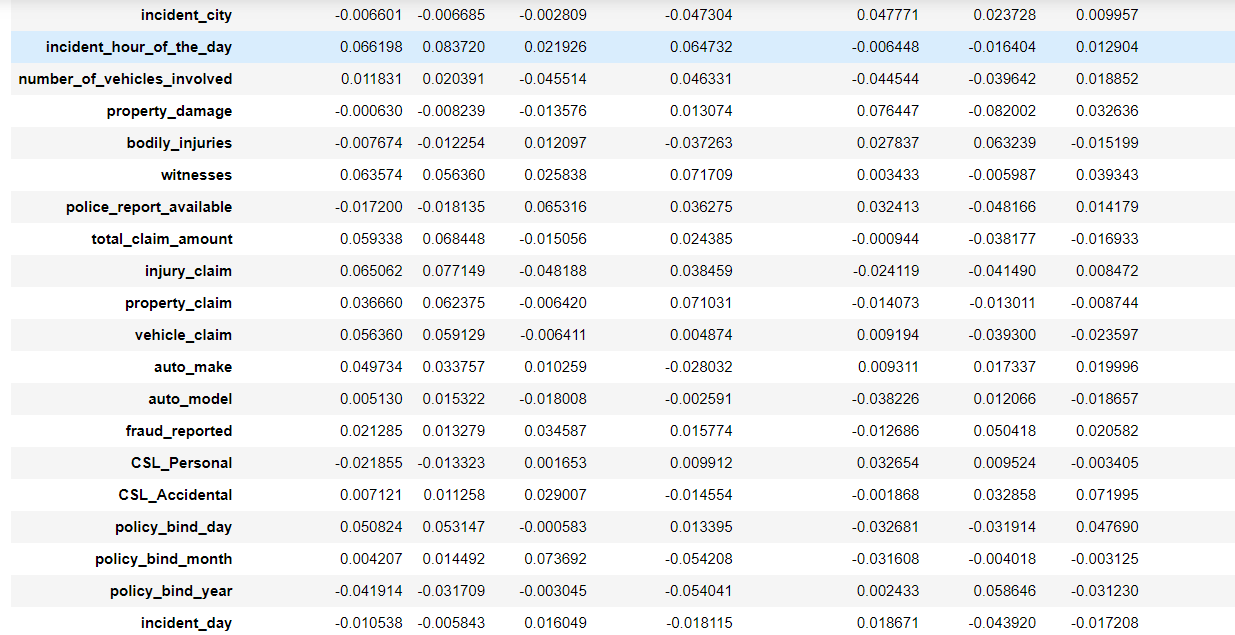
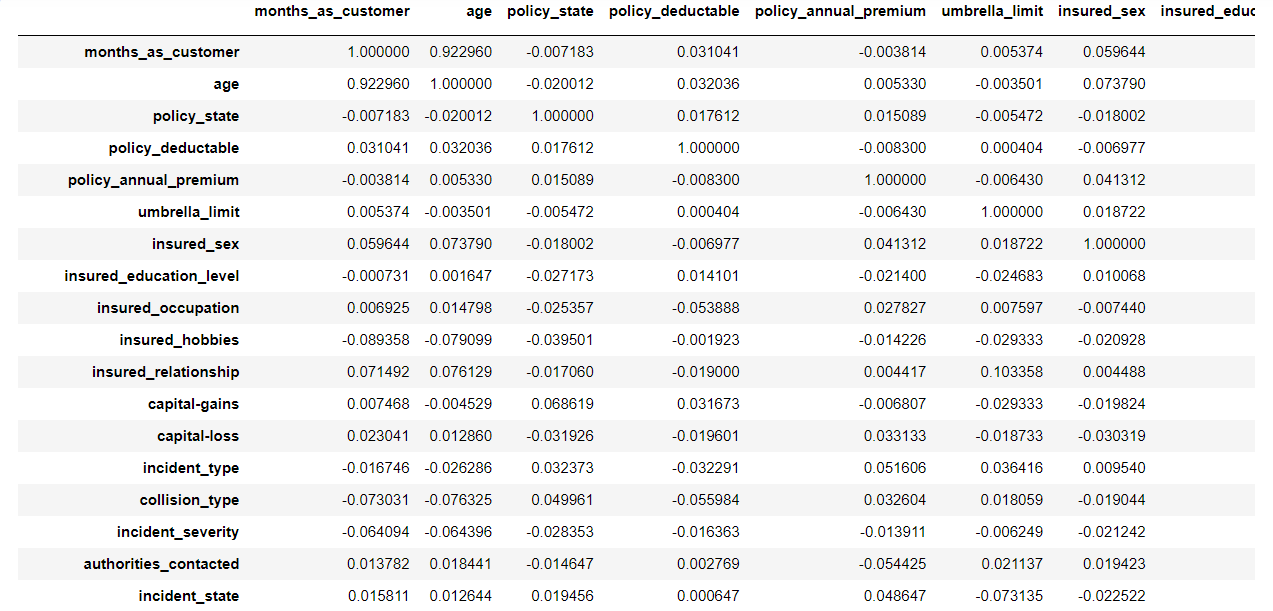
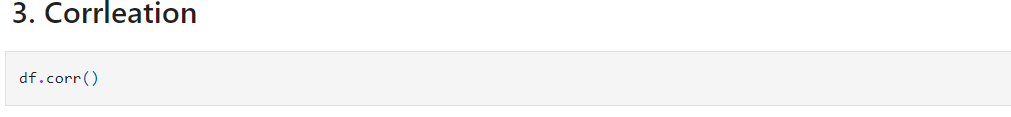


* **Skewness is important feature for continous data. There is no relevence of skweness for discrete numerical feature like month and categorical feature.So we gone ignore skewness present in discrete numerical and categorical feature.**
* **We also going to ignore sknewness in target feature.**

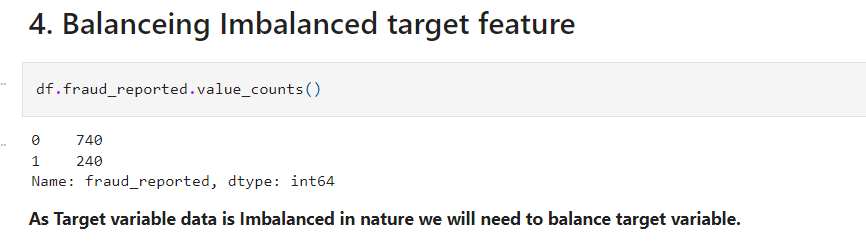


* Out above features 'umbrella\_limit','total\_claim\_amount' and 'vehicle\_claim' are continous variable with skew data. The variable 'incident\_month' is skewed but it is discrete in nature.So ignore it.
* We will use yeo-johnson method to transform negatively skewed data.

**For 'total\_claim\_amount','vehicle\_claim' skewness has not been removed but it got reduced**

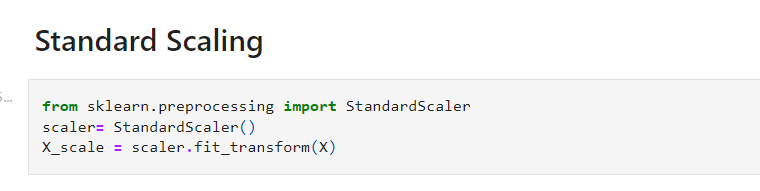


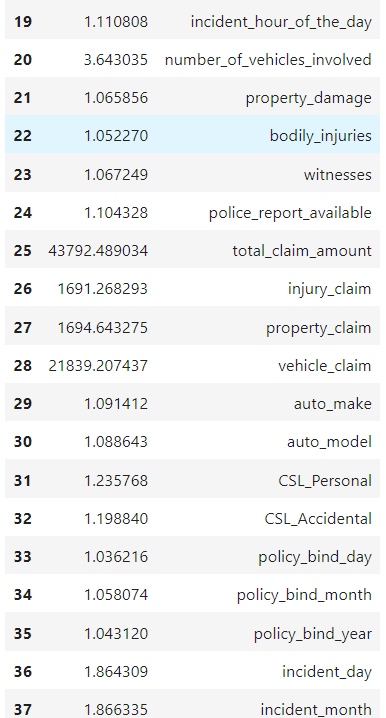
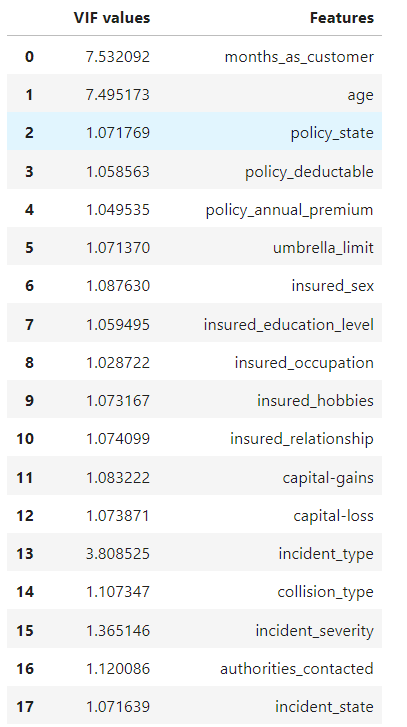
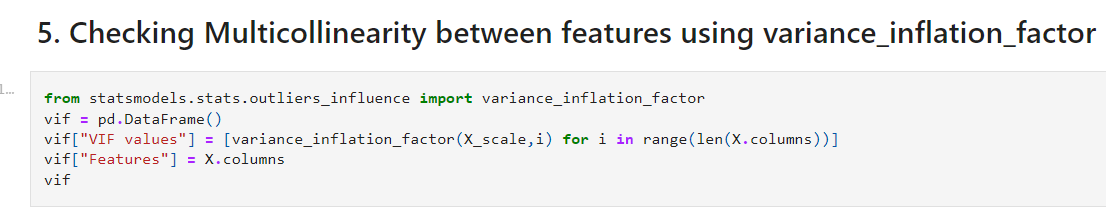
* incident\_severity is correlated with target variable with correlation of 0.4. Other variable are poorly correlated with target variable.
* Other variable are poorly correlated with target variable.
* injury\_claim,property\_claim,vehicle\_claim are highly correlated with each other.
* incident\_hour\_of\_the\_day is highly negative correlated with incident type.





*We have successfully resolved the class imbalanced problem and now all the categories have same data ensuring that the ML model does not get biased towards one category.*

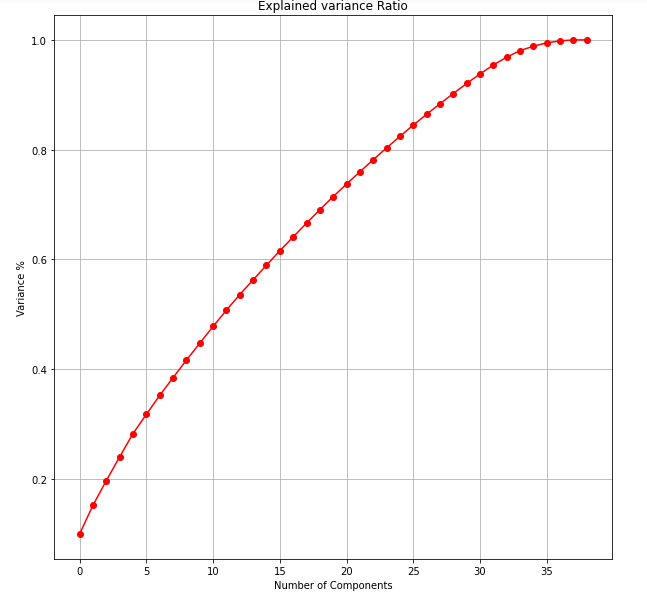
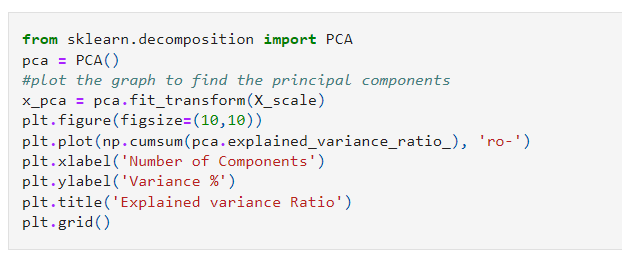




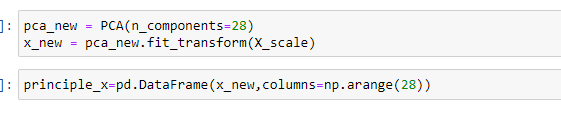
* Strategy to Address Multicollinearity :
* Removing Some of highly correlated features. But this will not work here as most of input features are correlated with each other either moderated or poorly.
* Another way to address Multicollinerity is to Scaled Data and then apply PCA.
* We will go by Second way for further investigation. As For some Independent feature VIF is exceed permissible limit of 10**.**

**PCA**

Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It's a widely used technique in data analysis and predictive modeling, particularly in the fields of machine learning and signal processing. The main goals of PCA include data reduction, feature extraction, and data visualization.

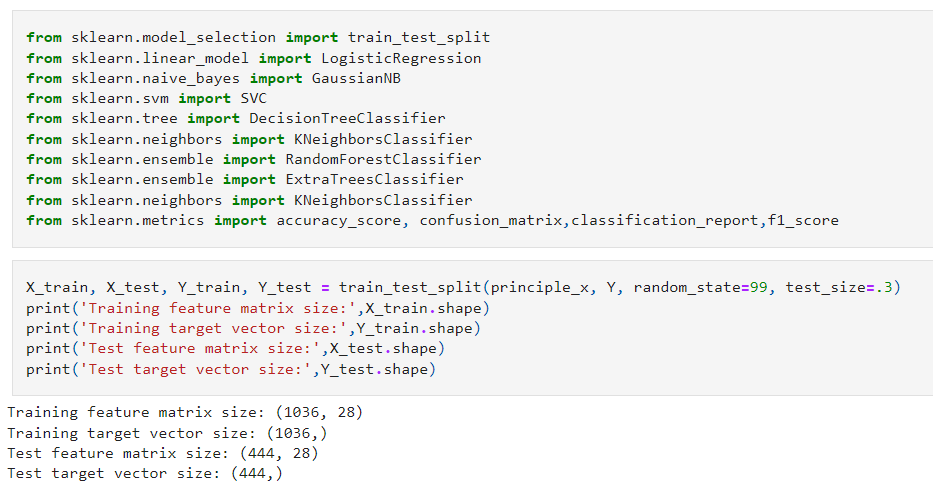


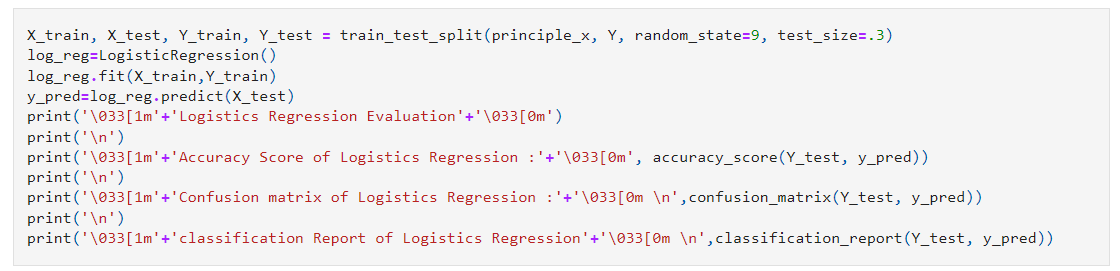
* AS per the graph, we can see that 28 principal components attribute for 90% of variation in the data.
* We shall pick the first 28 components for our prediction.

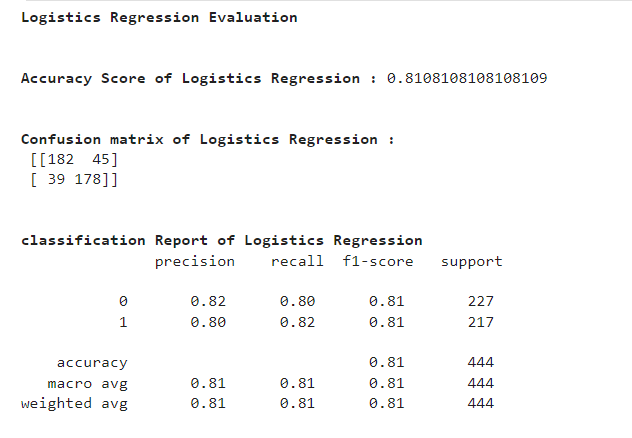


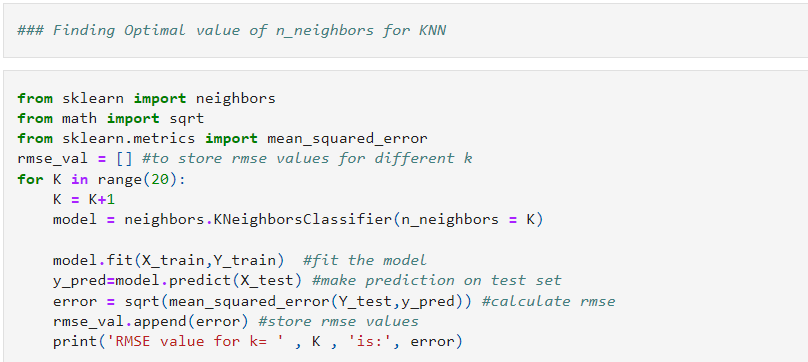
**5. Building Machine Learning Models:**

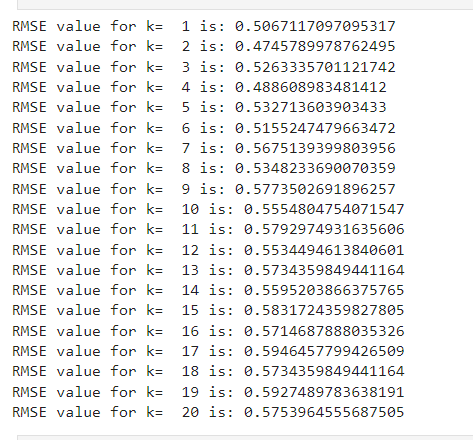
Importing Libraries for machine learning.

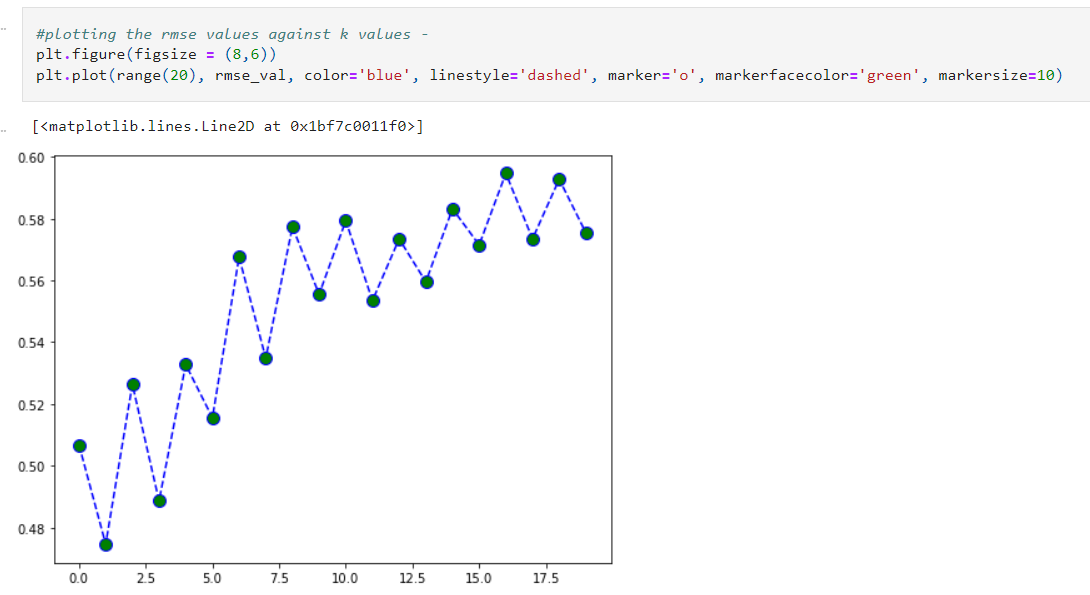


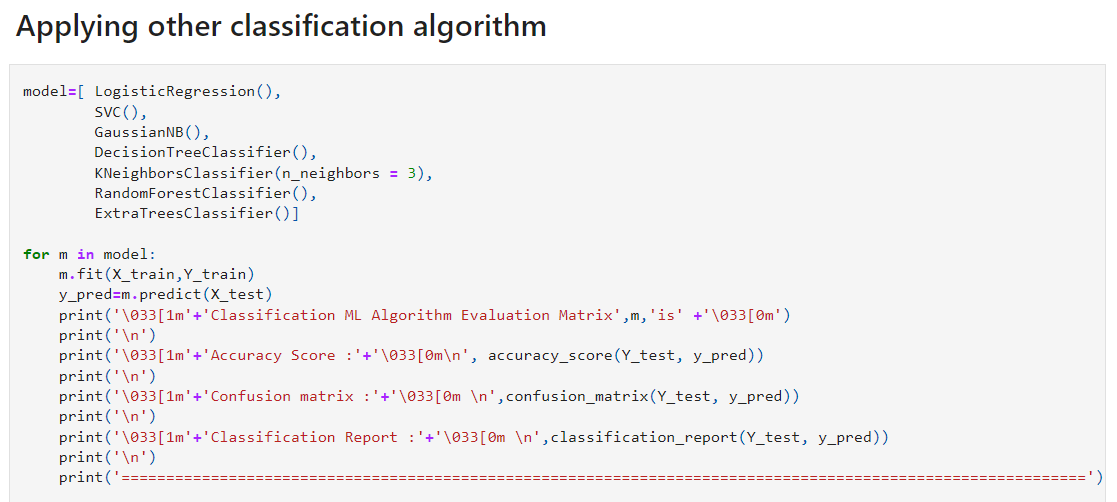


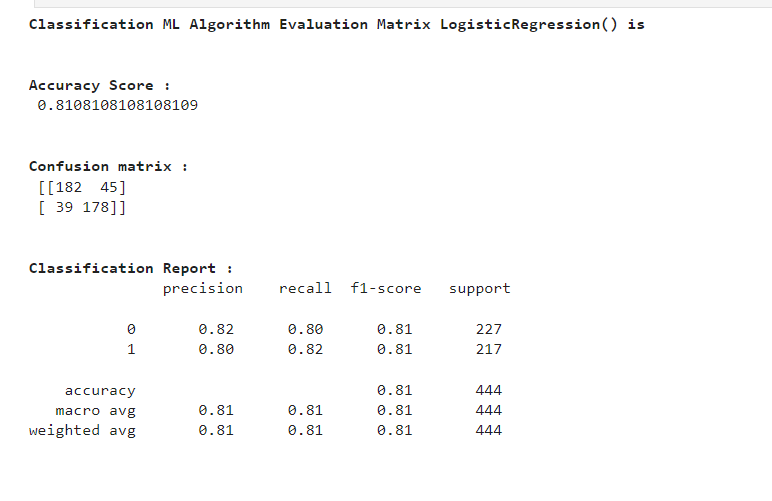


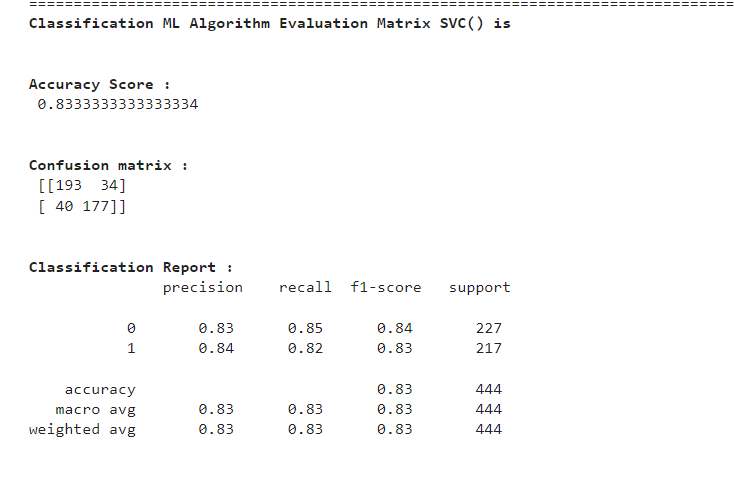


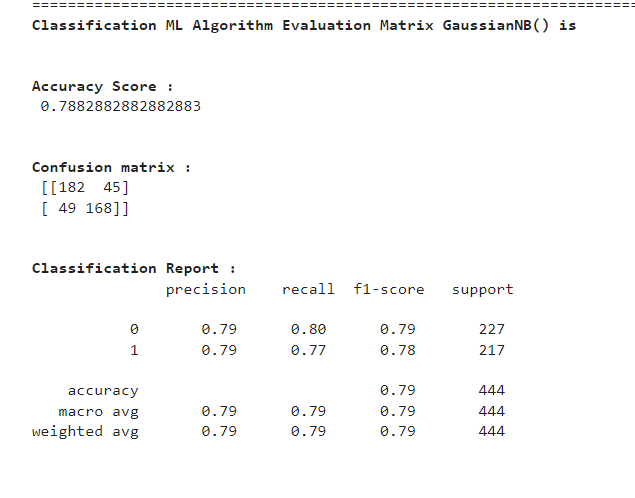


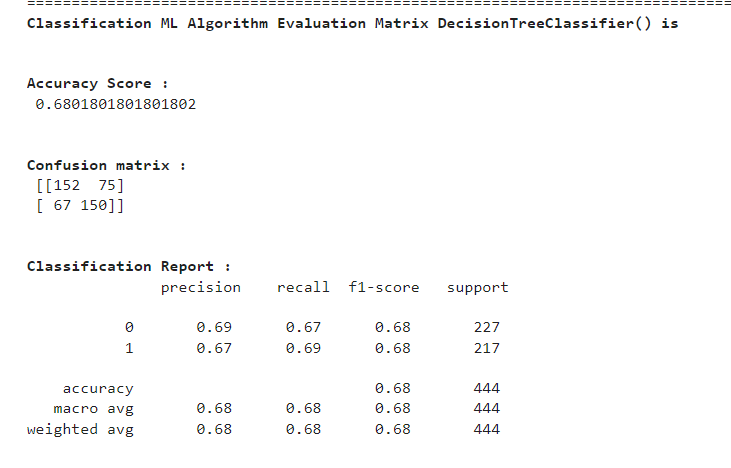
  
At k=2, we get the minimum RMSE value which approximately 0.4745789978762495, and shoots up on further increasing the k value. We can safely say that k=2 will give us the best result in this case.

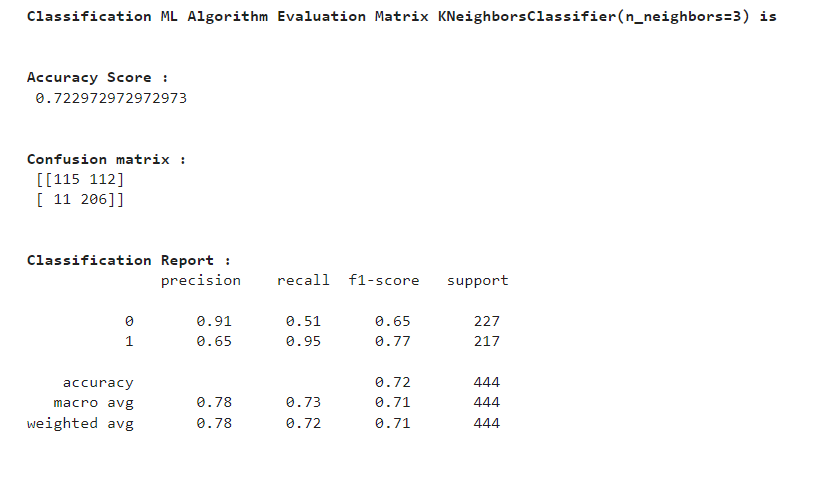


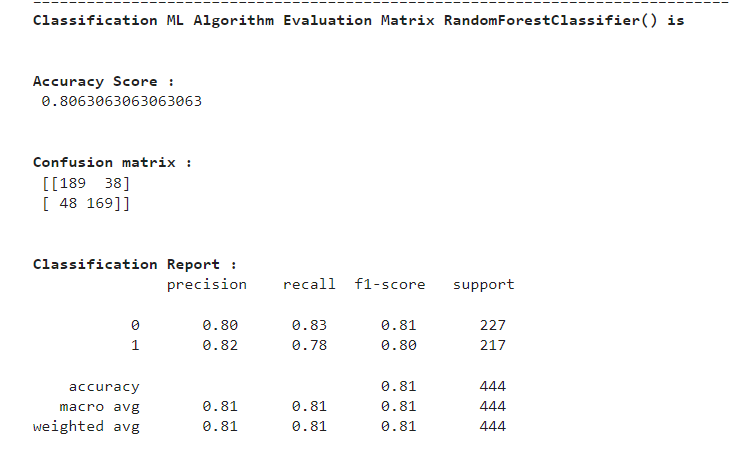


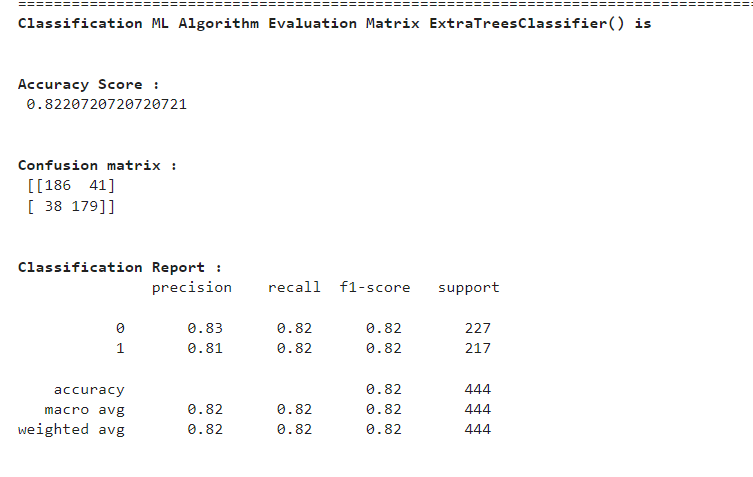








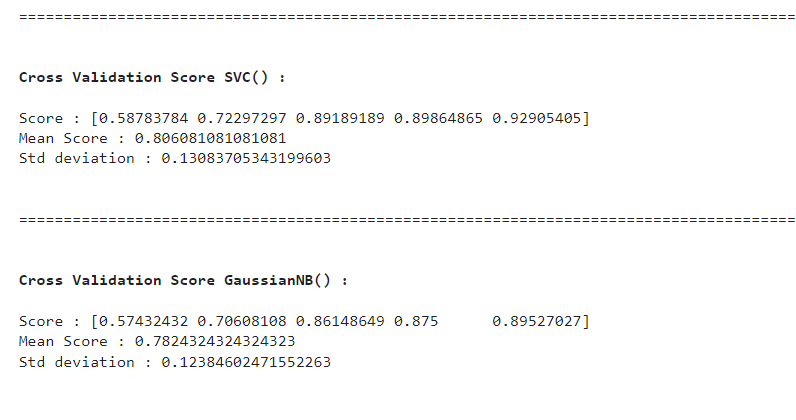
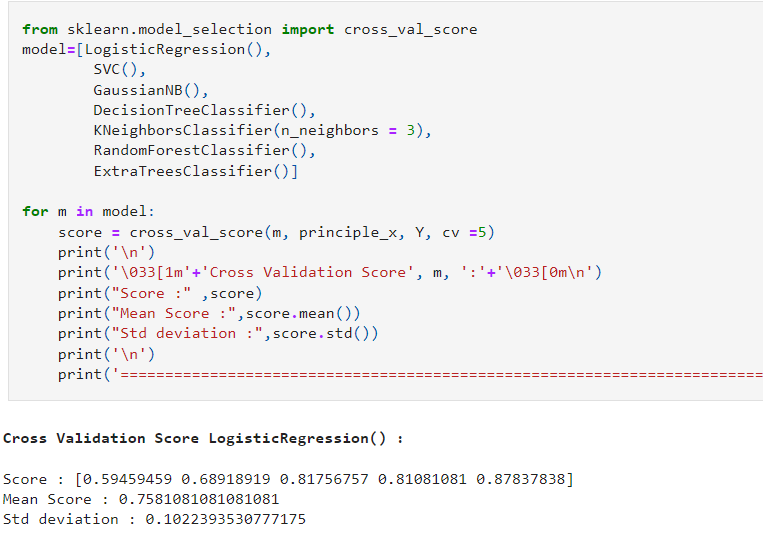




**Cross Validation**

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used to assess how the results of a statistical analysis will generalize to an independent data set, especially in situations where the goal is predictive modeling and one wants to estimate how accurately a predictive model will perform in practice.

The basic idea behind cross-validation is to divide the data into several segments: one segment is used to train the model, and the other is used to test the model. This process is repeated multiple times, with different segments used as the training and testing data each time. This helps in mitigating the problem of your metric of choice being dependent on the way you split up your data.





**6. Concluding Remarks:**

