**Insurance Claims- Fraud Detection**

**Problem Defination:**

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, 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.

Financial frauds are associated with sophisticated urban areas. But when it comes to insurance frauds, rural India has taken the lead due to various reasons. While baby steps like a fraudster database is being taken, such malpractices may not be contained without strict punishments under penal code

Insurers have identified at least 80 districts across the country which have excelled in fraudulent claims over the past decade. They have identified rings that operate with the efficiency of a corporation with well-trained men and women who collect data with the efficiency of a 21st century start-up.

A combination of poor due diligence in writing policies by insurance companies and the organisational efficiencies of criminals in identifying those who are on deathbed and in enlisting do.

Insurance fraud has been around since the beginning of insurance organizations. These are varied and complex crimes that often go unnoticed and cost the insurance industry billions a year. Different types of insurance are prone to different crimes, however, in most cases, it manifests deliberate damage to the insured item or the purpose to obtain goods without paying. Detecting insurance fraud can be difficult since not every claim can be investigated thoroughly.

What are the benefits that Machine Learning can bring for the evaluation and resolution of insurance frauds?

* All claims suspected of fraud will be more accurately detected.
* Data is processed in very short periods of time.
* The system can demonstrate where connections can exist between various factors that may be imperceptible to human eyes.
* The continuous revision of this type of schemes and the application of variations in data analysis will allow anticipating the discovery of new fraud schemes.

As we have seen, fraud detection is a knowledge-intensive activity that allows classifying correctly whether the transaction or claim is legitimate or fraudulent.

The popular form of machine learning applied to the insurance industry is called deep anomaly detection. Anomaly detection works by analyzing normal, genuine claims made by the customer and forming a model of what a typical claim looks like. This model is then applied to large data sets.

Machine Learning: A Big Step in Fraud DetectionMachine Learning is a part of Artificial Intelligence (AI). The idea behind Artificial Intelligence is to create a computerized system that can engage in complex analysis and not only replace human input but improve upon it. Machine Learning applies AI and “gives” systems the ability to learn and improve from experience, with no extra programming. In order to do this, systems analyze large, labeled data sets. AI may take over menial tasks and free human agents to do more complex analysis.

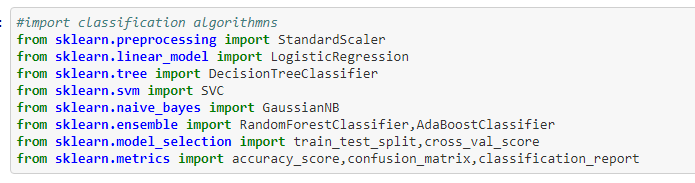
Now lets bulid model.

**Importing the Libraries**:

First we have to import necessary libraries .



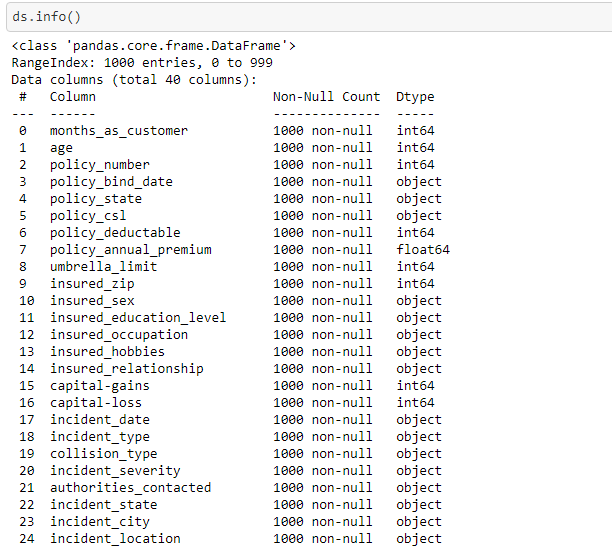
As this is classification problem, we have to import classification algorithms

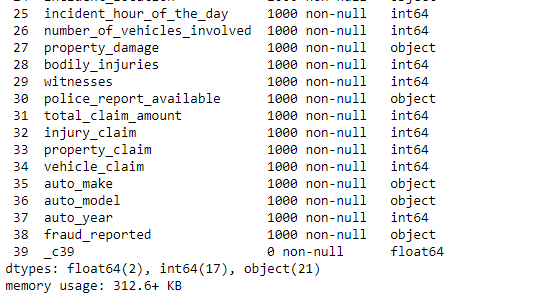


**Getting data**:

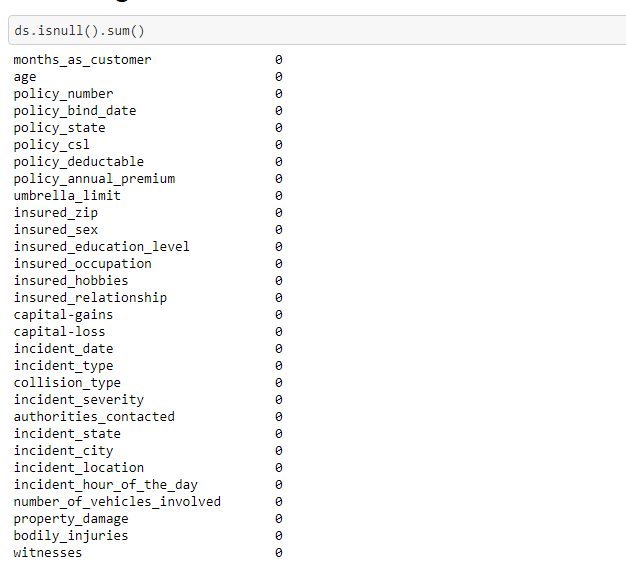


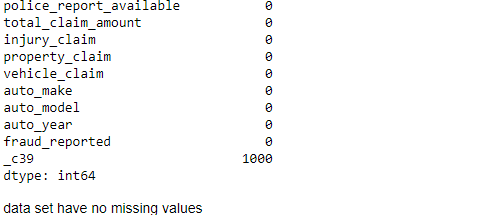
**Short information about data set:**





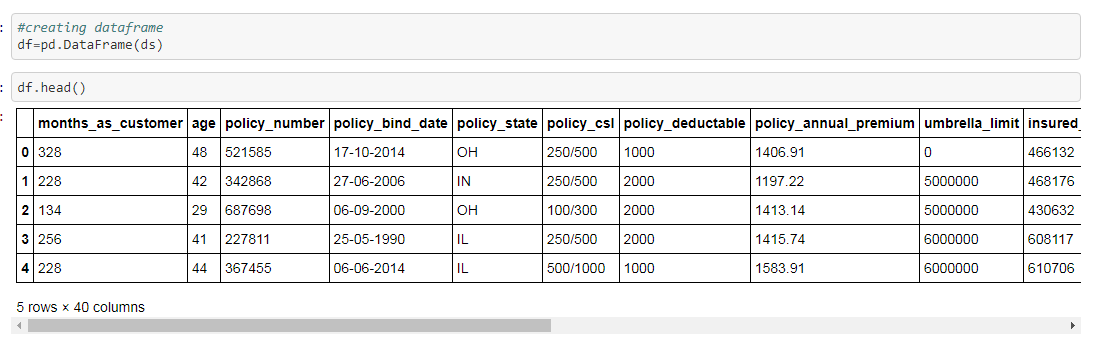
Data set have 39 columns and each column has 1000 values.There are 2 float type features ,16 int type features, and 21 object type features.

**Chec king missing values**:

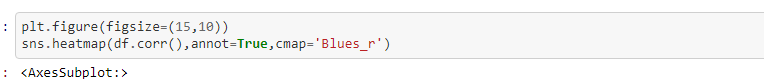


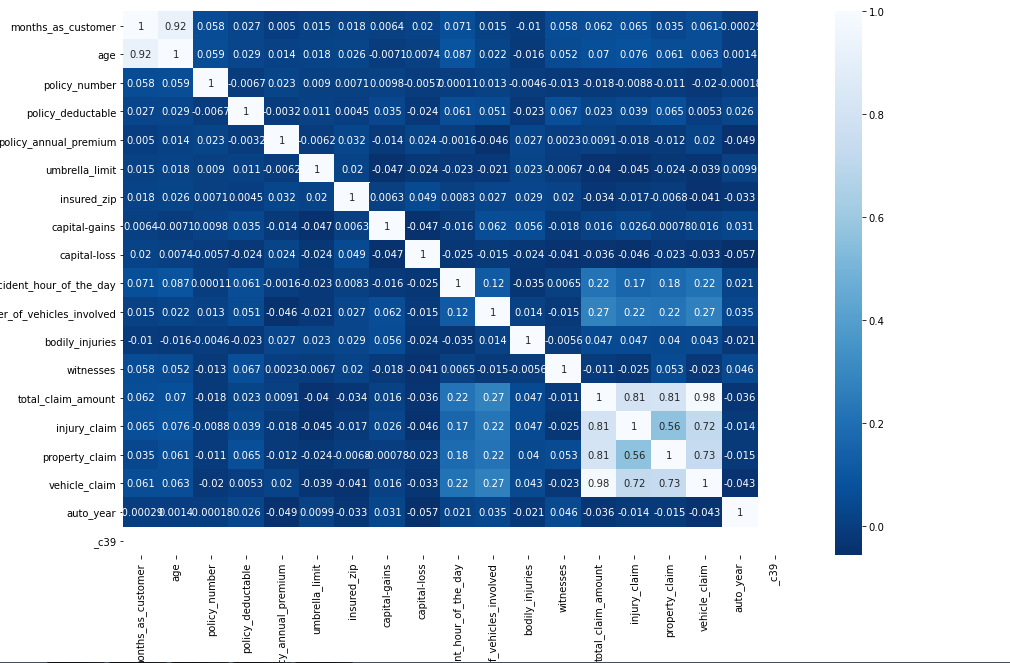
As shown in above figure, there are no missing values in data set.

**Creating data frame**:



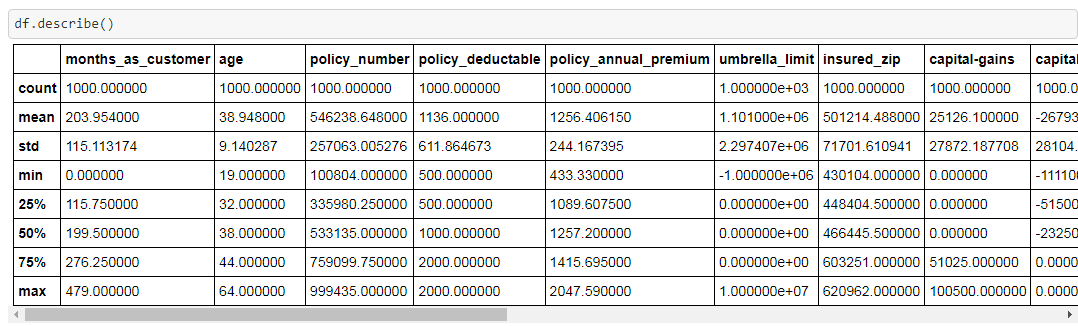
**Checking corelation**:





Above Heatmap shows corelation between feautures.

**Stastical summary:**



This function gives much information about data set such as count function tells that there are no missing values. Also it provides mean, median, max values from which we can derive many conclusions

Followings are countplot of categorical data which shows

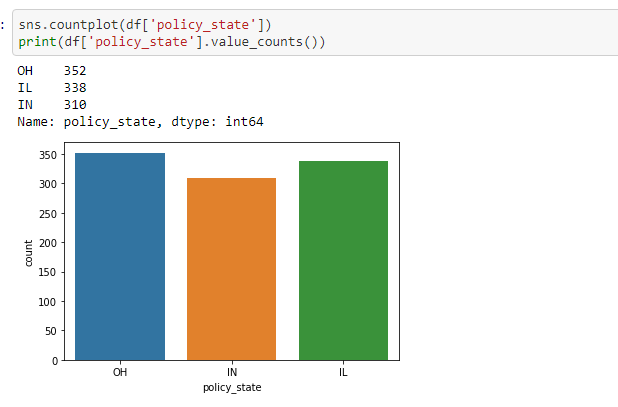
different values of particular column and counting of each

value.

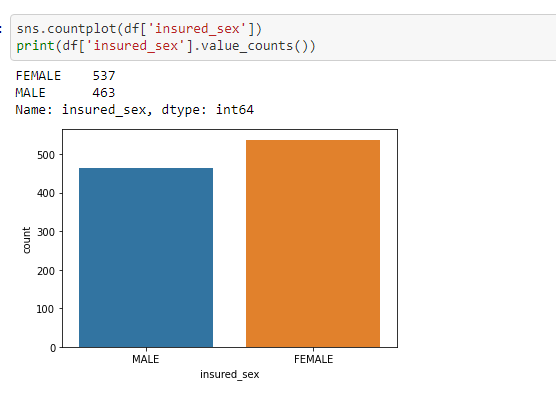
1:Countplot of policy\_csl



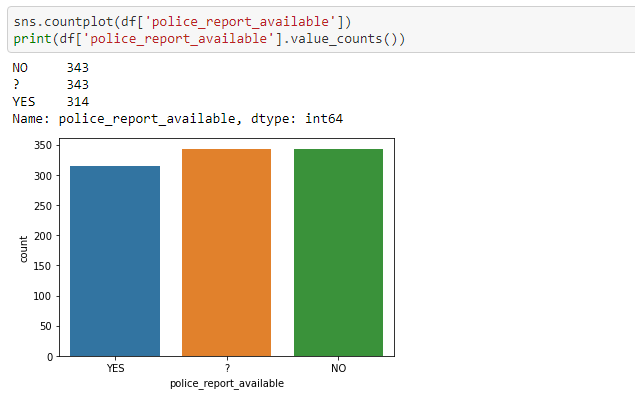
2:Countplot of policy\_state



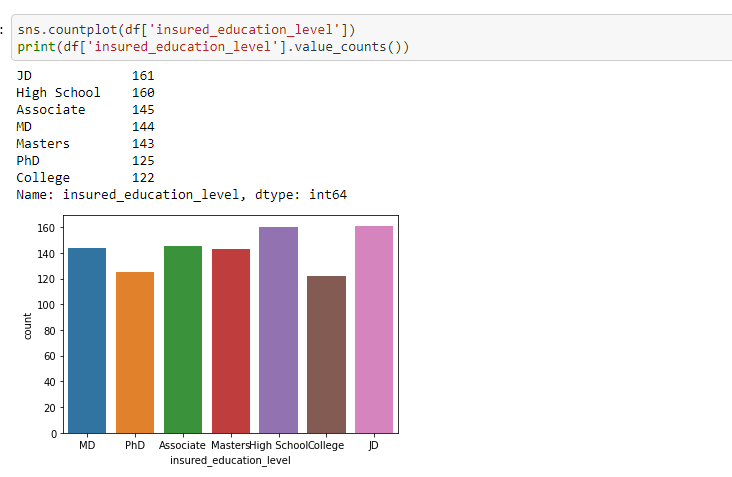
3:Countplot of insured\_sex



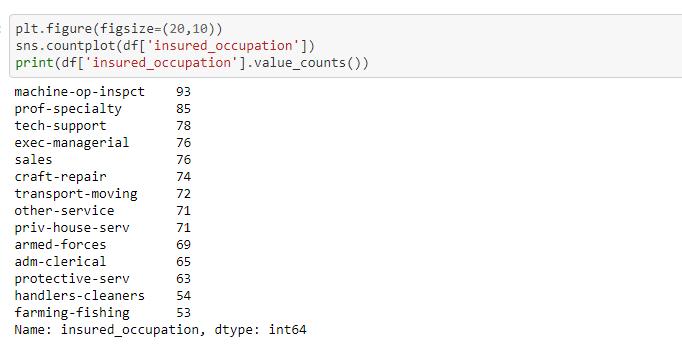
4:Countplot of police\_report\_available:

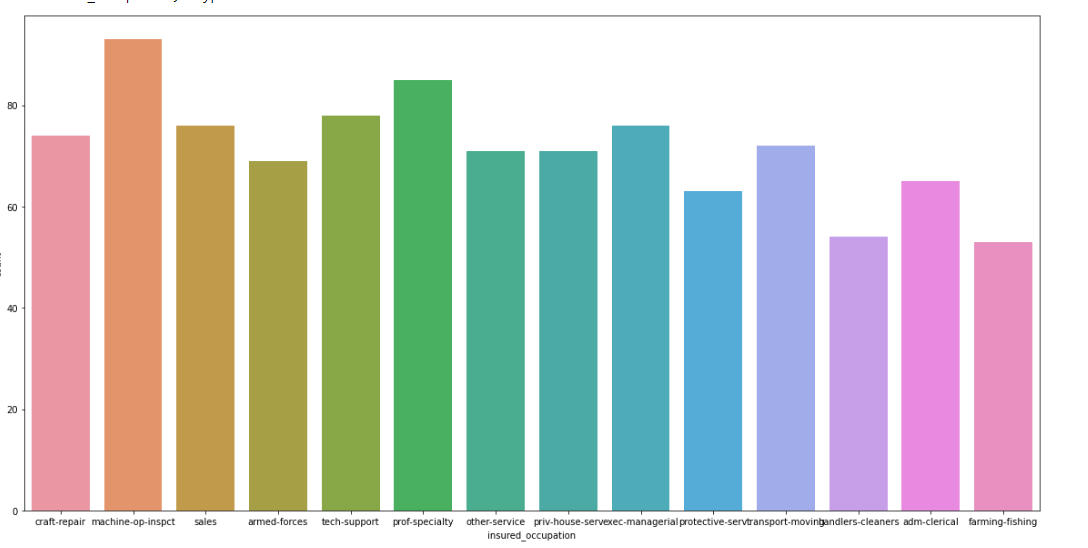


5:Countplot of insured\_education\_level:



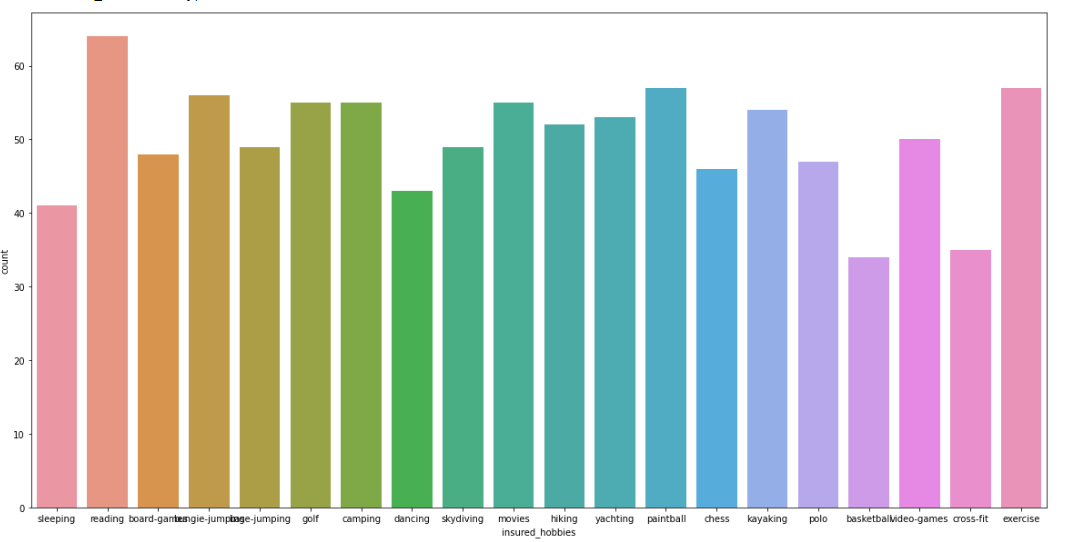
6:Countplot of insured\_occupation

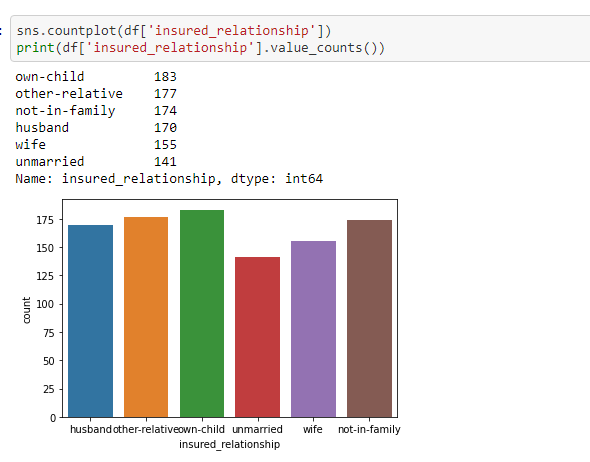


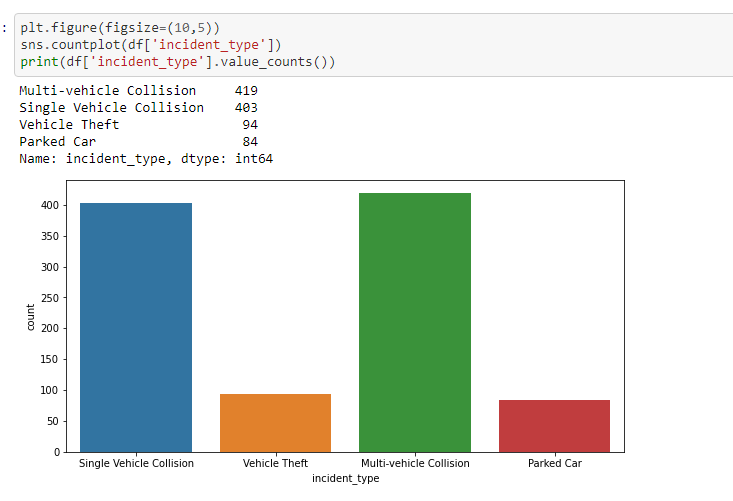


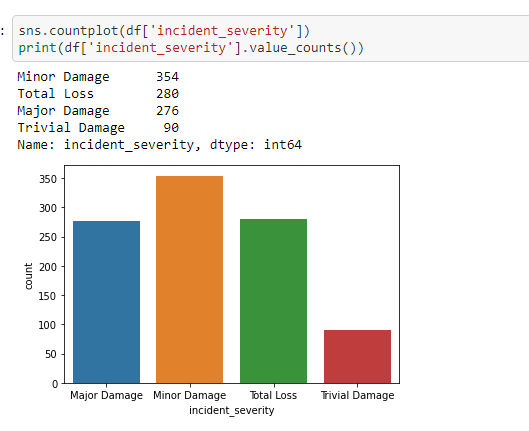
7:Countplot of insured\_hobbies





8:Countplot of insured\_relationship: 

9:Countplot of incident\_type:10:countplot of incident\_severity

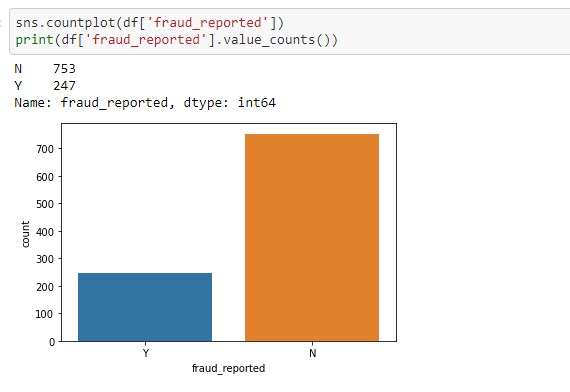


:

11:countplot of property\_damage:

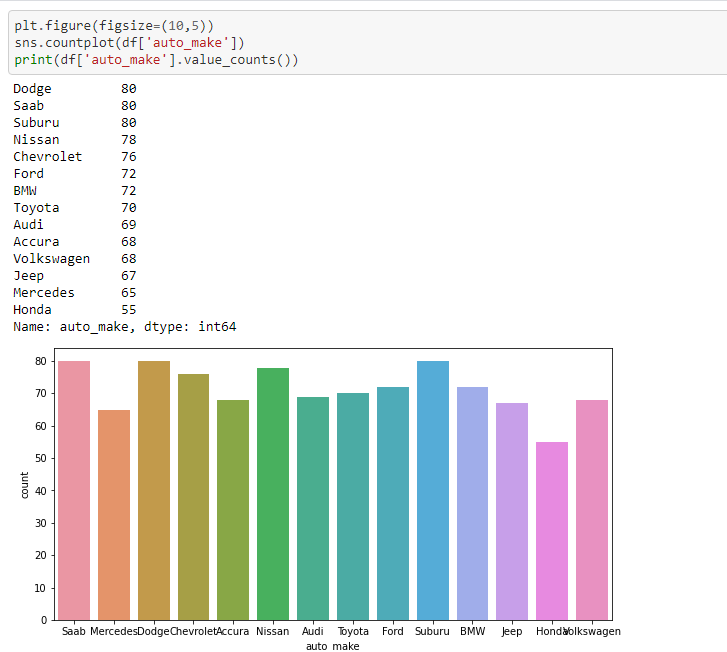


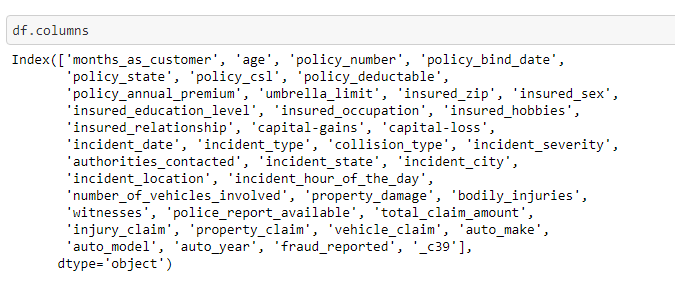
12:countplot of fraud\_reported:



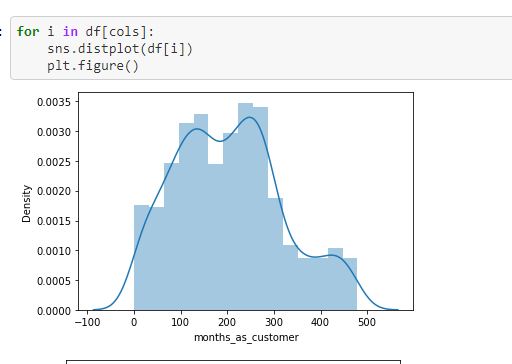
As shown in above count plot of target variable, there is class imbalance problem which we need to handle later on.

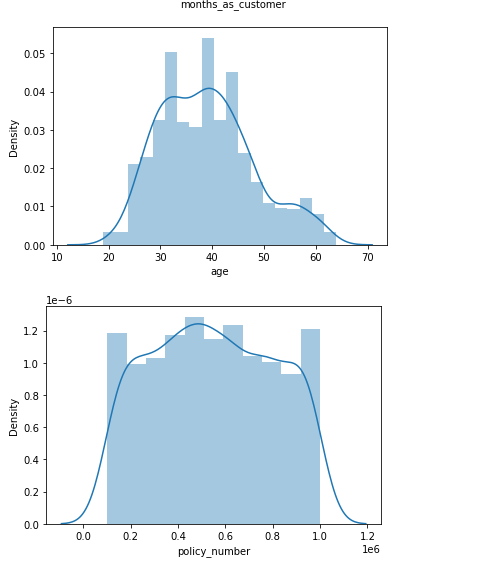
14: Count plot of:auto\_make:

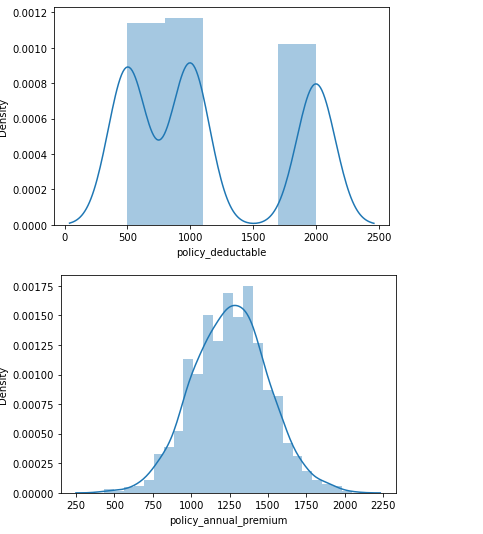


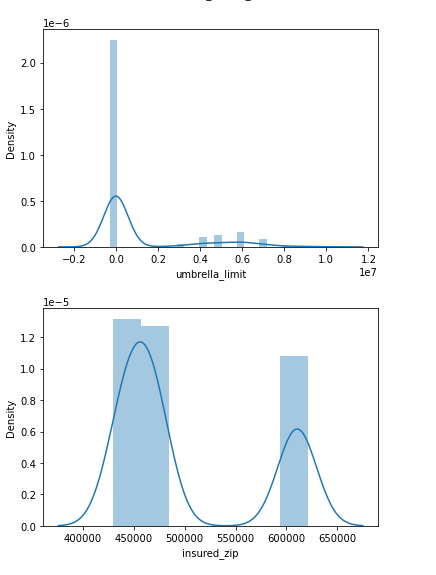


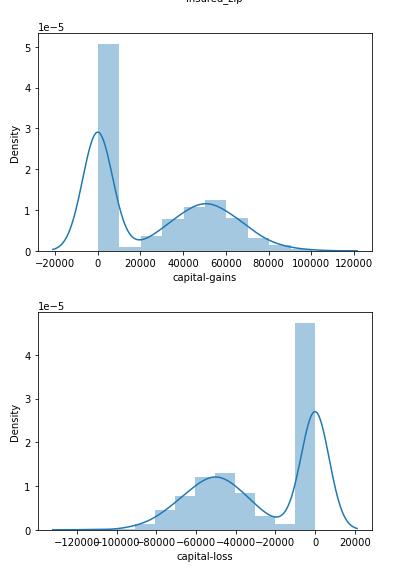
Followings are distribution plots of numerical data.

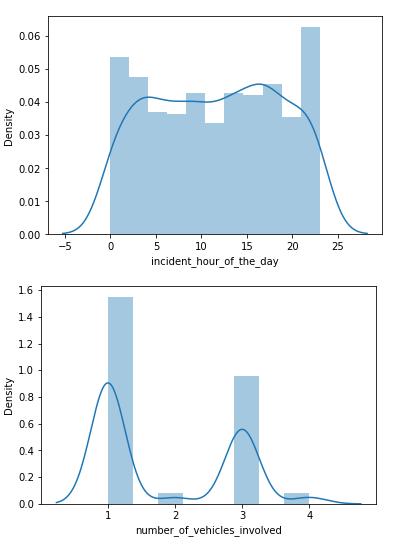


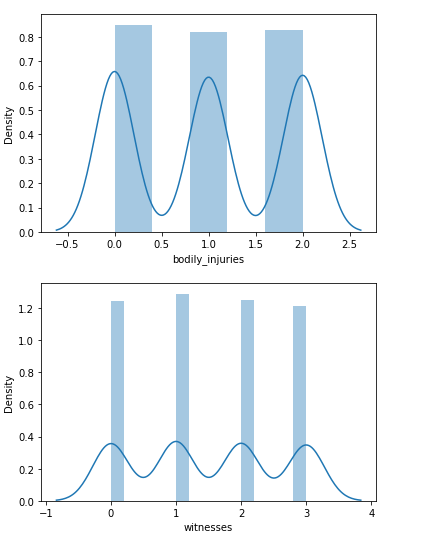


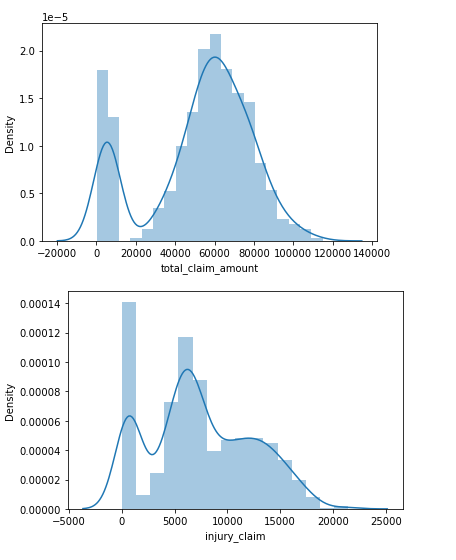


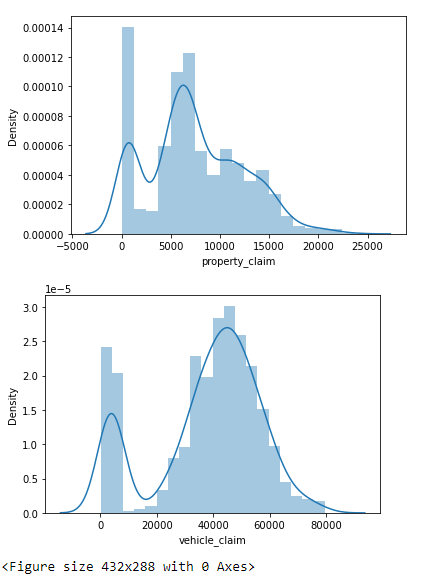






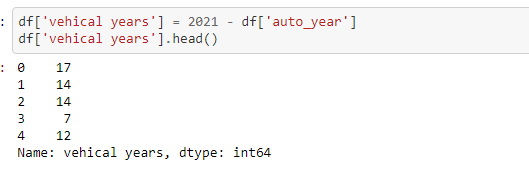




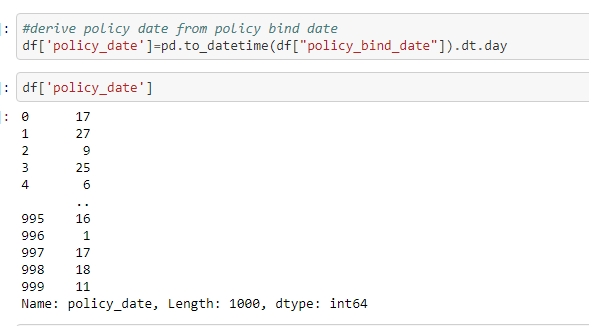


**Feature Engineering**: We can derive some more columns from existing column which are important.

1:Total vehical years derived from auto\_year column.



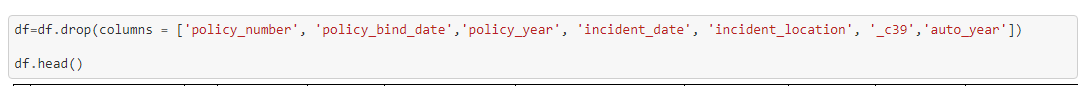
2:Policy\_date derived from policy\_bind\_date



3:Policy\_month derived from policy\_bind\_date

4:Total years of policy derive from policy\_bind\_date

**Dropped unnecessary columns**:



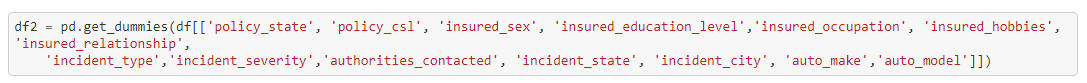
**Pre-Processing**:

Encoding: First we have to encode the categorical data into numerical data. There are different techniques of encoding:

* One Hot Encoder: Encode categorical integer features using a one-hot aka one-of-K scheme. The input to this transformer should be a matrix of integers, denoting the values taken on by categorical (discrete) features. The output will be a sparse matrix where each column corresponds to one possible value of one feature.
* Label Encoder: Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.
* OrdinalEncoder: In ordinal encoding, each unique category value is assigned an integer value. For example, “red” is 1, “green” is 2, and “blue” is 3. This is called an ordinal encoding or an integer encoding and is easily reversible. Often, integer values starting at zero are used.

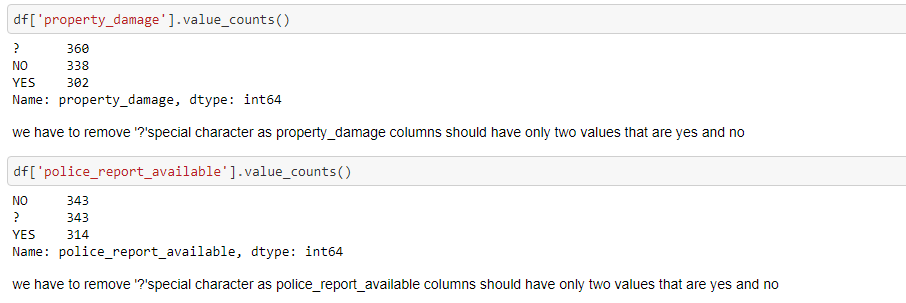
In this project we used OneHotEncoder and Label Encoder to encode the categorical data into numerical data.

1:OneHotEncoder:To encode the data using OneHotEncoder, we used get\_dummies method of that.

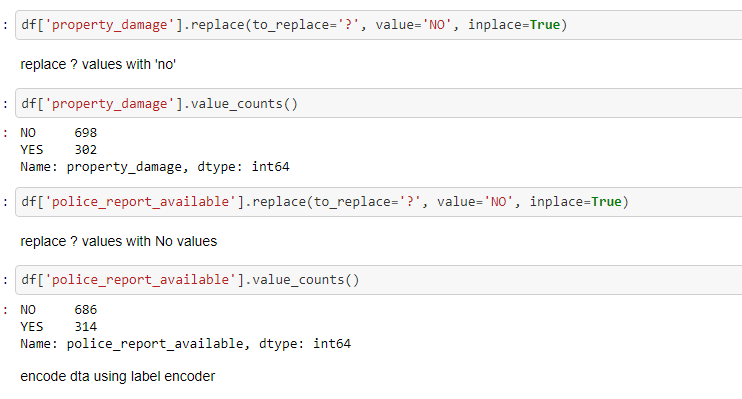


In some columns,there is special character that is ‘?’,so we have to replace it with correct value.

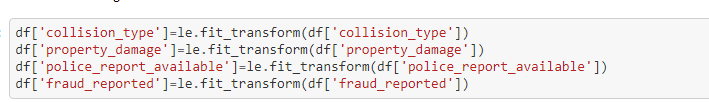
Proprty\_damage and police\_report\_available columns have special character ‘?’,Following fig shows total counting of values.



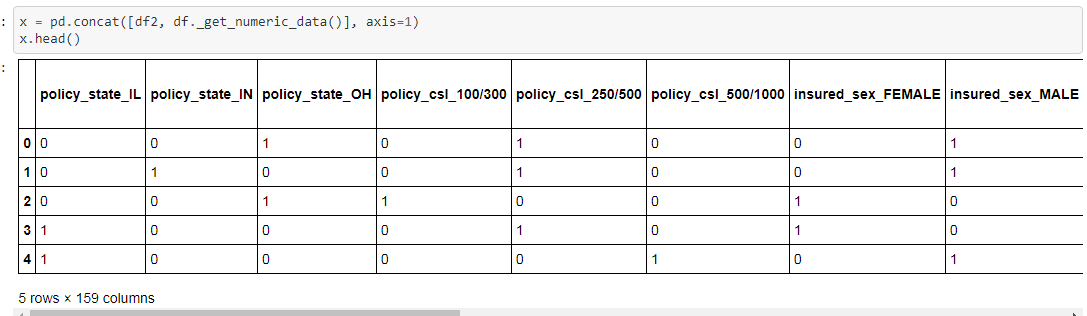
So let’s relace ‘?’ with ‘NO’ value of that columns.



Let’s encode remaining categorical data using Label Encoder:



After encoding data is as following:



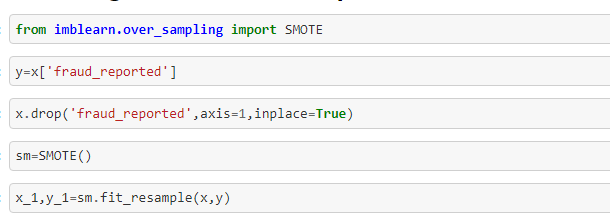
**Class Imbalance Problem:**

It is the problem in machine learning where the total number of a class of data (positive) is far less than the total number of another class of data (negative)

Our target variable is unbalanced means there are more values of ‘No’ than ‘yes’ as shown in above attritution count plot. So we have to handle it. Tis problem is solve using either by oversampling or undersampling.

* Over\_sampling: Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset. Random undersampling involves randomly selecting examples from the majority class and deleting them from the training dataset
* Under\_Sampling: Undersampling refers to a group of techniques designed to balance the class distribution for a classification dataset that has a skewed class distribution.

We solve this problem using oversampling method. For this we have to import necessary library



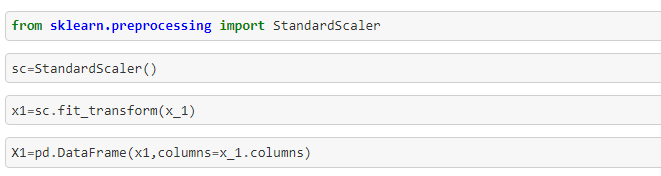
**StandardScaler:**

**Scaling** of Features is an essential step in modelling the algorithms with the datasets. The data that is usually used for the purpose of modeling is derived through various means

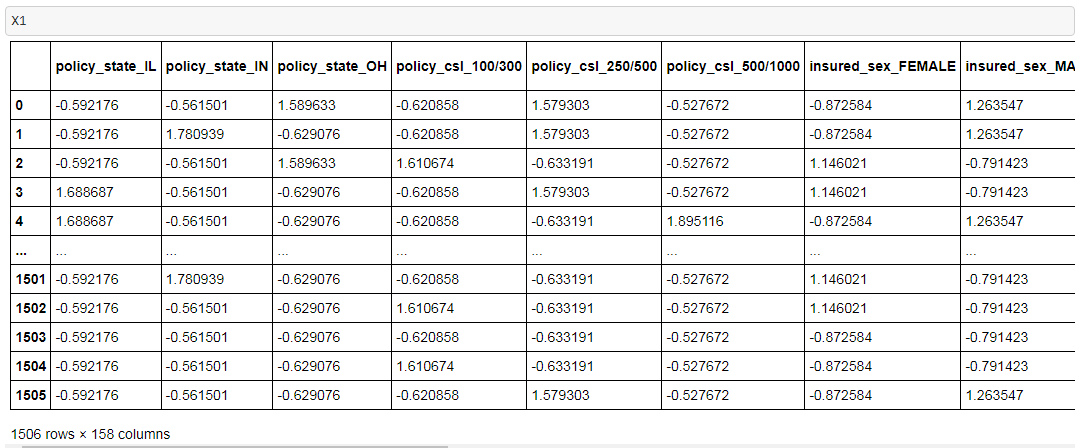
. StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. Deep learning algorithms often call for zero mean and unit variance

Standard scaler scales dataset into one form so model can perform in better way.

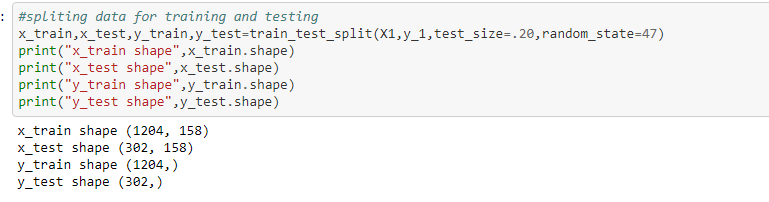
Following figure shows how we used standard scaler to standardizes data.



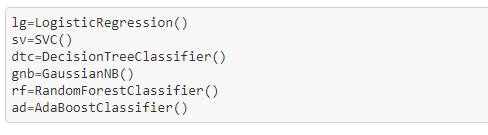
After scaling ,data is as following



Now data is ready for testing and training. But for this we have to split data for training and testing. Following figure shows splitting data for training and testing:

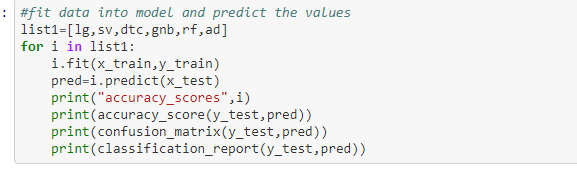


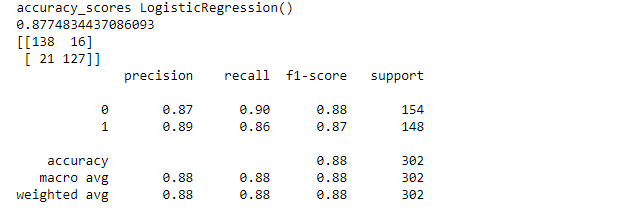
At start we imported classification algorithm,So now let’s create instance of algorithms.



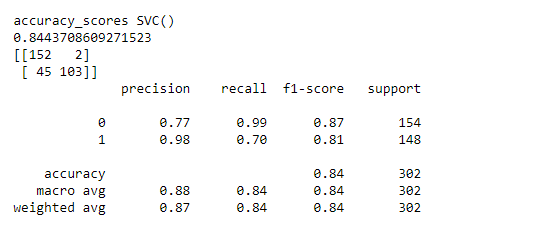
**Model Building:**

Now let’s fit data into model and predict the values and check accuracy\_scores.

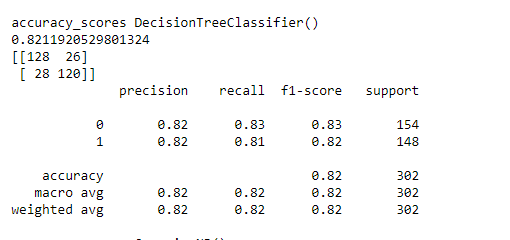




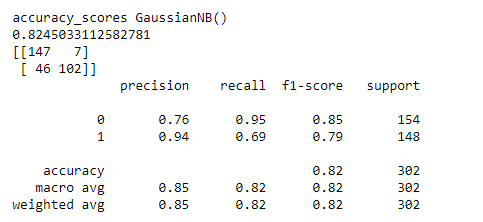
Accuracy\_Scores of Support Vector Machine:



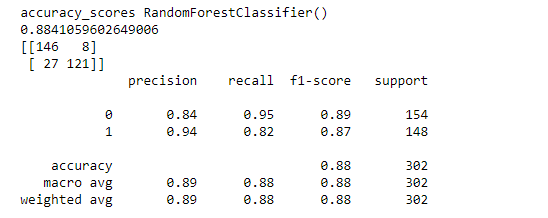
Accuracy\_Scores of DecisionTreeClassifier



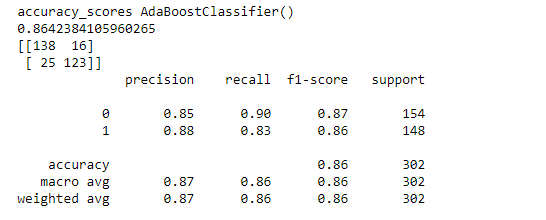
Accuracy\_Scores of GaussianNB:



Accuracy\_Scores of RandomForestClassifier:



Accuracy\_Scores of AdaBoostClassifier:



As we can see, the Random Forest classifier and LogisticRegression gave high accuracy. But first, let us check, how they both perform, when we use cross validation.

**Cross Validation**:

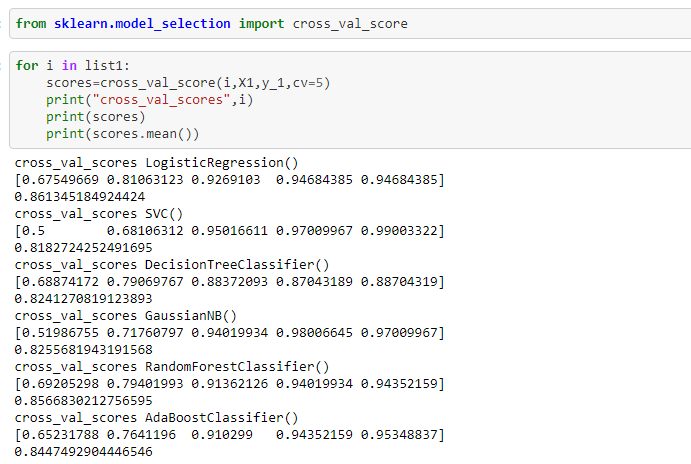
Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. It is a popular method because it is simple to understand and because it generally results in a less biased or

less optimistic estimate of the model skill than other methods, such as a simple train/test split.

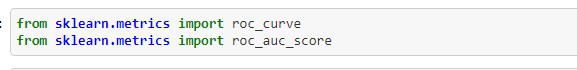
Following are cross\_val\_scores of above classification algorithms:

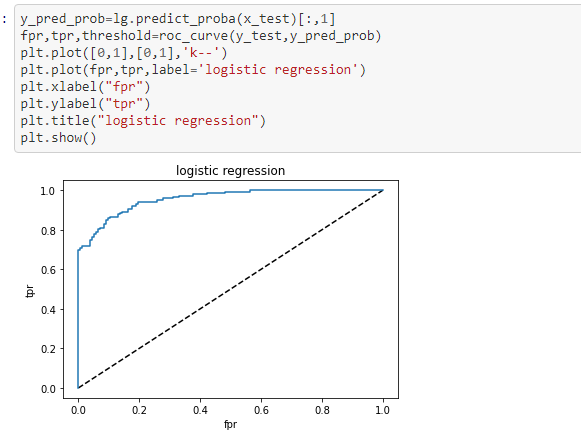


From above figure, we can say that LogisticRegression performs well.

**AOC-ROC Curve**:

AUC - ROC curve 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 0 classes as 0 and 1 classes as 1



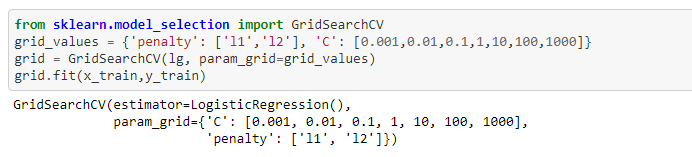


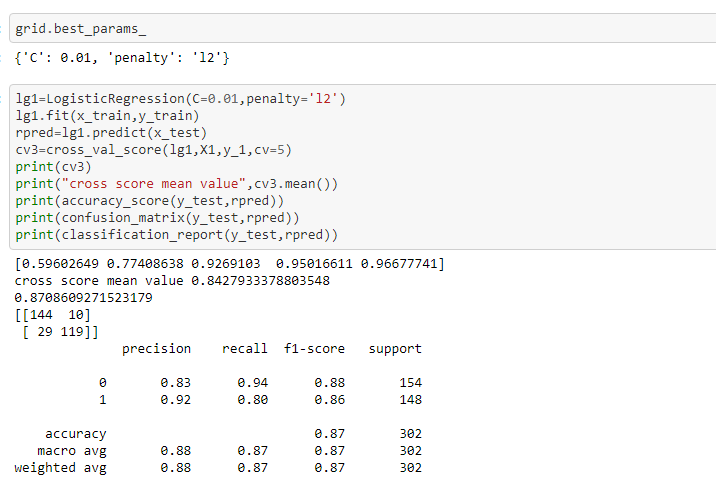
**Hyper Parameter Tunning**:

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), hyperparameter optimization or tuning is the problem of choosing a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)" \o "Hyperparameter (machine learning)) for a learning algorithm. A hyperparameter is a [parameter](https://en.wikipedia.org/wiki/Parameter) whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined [loss function](https://en.wikipedia.org/wiki/Loss_function) on given independent data.[[1]](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-abs1502.02127-1) The objective function takes a tuple of hyperparameters and returns the associated loss.

Some examples of model hyperparameters include:

* The penalty in Logistic Regression Classifier i.e. L1 or L2 regularization.
* The learning rate for training a neural network.
* The C and sigma hyperparameters for support vector machines. 

Derive best parameter and use these parameters to improve performance of the model. 

After hyper parameter tunning model performance increased in little bit

**Logistic Regression**:

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common [logistic regression models](https://www.sciencedirect.com/topics/computer-science/logistic-regression-model) a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful [analytic technique](https://www.sciencedirect.com/topics/computer-science/analytics-technique).

Logistic regression, despite its name, is a classification model rather than regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. It is an extensively employed algorithm for classification in industry. The logistic regression model, like the Adaline and perceptron, is a statistical method for binary classification that can be generalized to multiclass classification.

**Conclusion**

The machine learning models that are discussed and applied on the datasets were able to identify most of the fraudulent cases with a low false positive rate i.e. with a reasonable precision. This enables loss control units to focus on new fraud scenarios and ensuring that the models are adapting to identify them.