**Report on COVID19 outcome**

**Brief description and motivation of the problem for Machine Learning?**

Coronavirus is a disease that has no clearly defined treatment. The coronavirus 2019 (COVID19) originated from China.

The goal of this project is to predict a COVID19 outcome probability based on different input parameters that will be most related to target variable outcome. It is really important to find out whether the patients is death, recovered, treatment etc.

We have a huge dataset of data, where most features are categorical and also contain many missing values in numerical columns as well as in categorical columns. I think that correct mean encoding should be important. Also the number of columns is quite high so it could be tempting to make some automatically processing for all columns. I personally think that it is important to analyze each variable and it could help to do a better processing. For this problem, we applied different supervised machine learning models to predict coronavirus disease.

**What is/are the objectives of the problem(s) that are addressed in your project (Classification/ Regression/ Clustering Rules/ Text Analytics/ Reinforcement learning etc...)?**

We choose 3 models for our classification problem because in our dataset target variable is in categorical format so, when class label is in categorical then this problem is related to classification. Models are given below that we have choose for our prediction.

* Decision Tree
* Logistic Regression
* Random Forest

We will measure the performance of each models on different metrics that shown below:

* **Accuracy**

Measure to evaluate how accurate model’s performance is:

* **Precision**

Measure to evaluate how accurate model’s performance is:

* **Recall**

Measure to evaluate how accurate model’s performance is:

* **F1**

Provides information of both sides TN and TP

* **Confusion Matrix**
* **Classification Report**

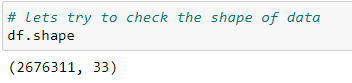
**Characterization of the data set: source URLS; size; number of attributes; has/ does not have missing values; number of examples etc. Clean and remove the missing values from the dataset. Provide a clear strategy?**

We have a latest data of covid19. This is a huge dataset so, we will explore the dataset to know more about the data shape, descriptive analyses, correlation, missing values, dtypes, column names etc.

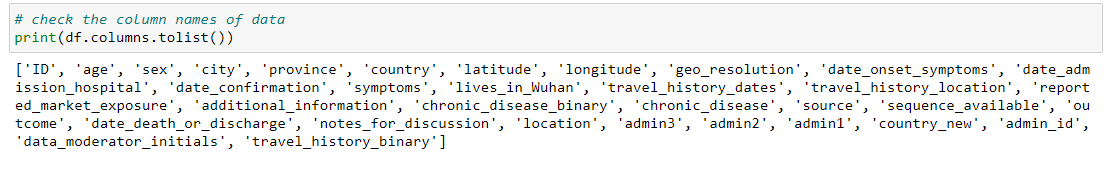
I have get the dataset from GitHub and source URL is:

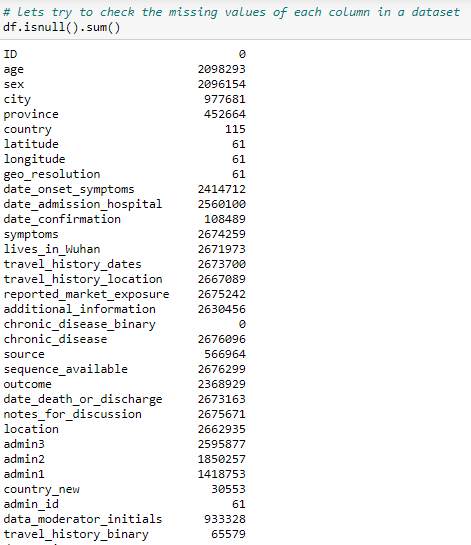
[**https://github.com/beoutbreakprepared/nCoV2019/tree/master/latest\_data**](https://github.com/beoutbreakprepared/nCoV2019/tree/master/latest_data)

Let’s try to look the shape of data:



From above that we have a data in which 2676311 rows and 33 columns.

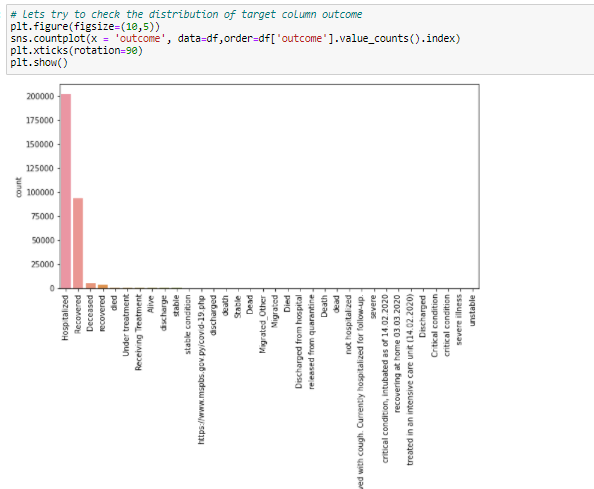




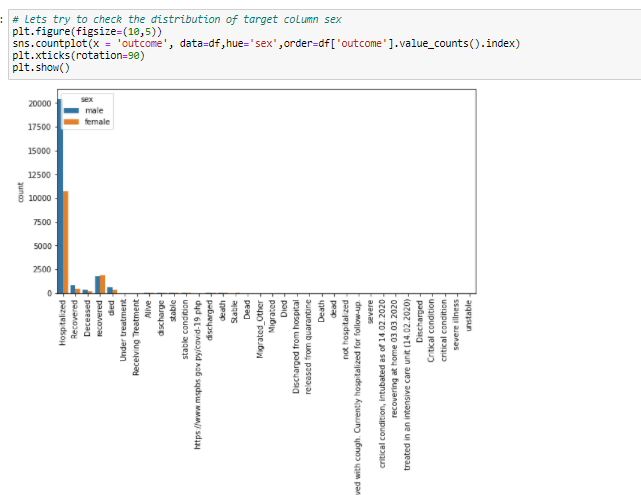
From above fig we can see that too much missing values in every columns. So ,we will handles these missing values according our problem.

**Data Visualization**

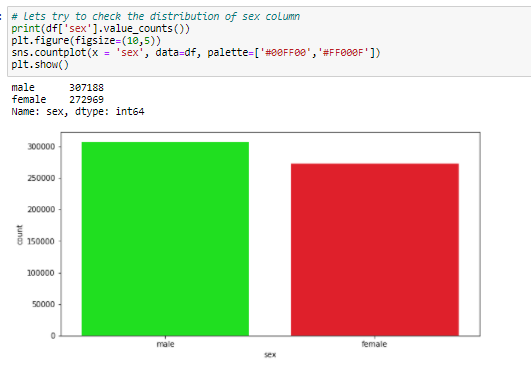
In this section, we will visualize all the columns and check the distribution and relationship with others columns and also we will get more analysis from the graph.



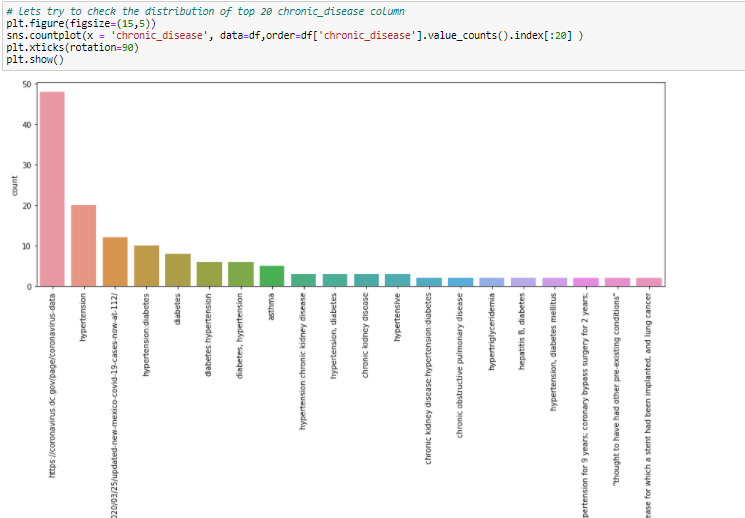
Form above Fig we plot a graph of each category count and we can see that mostly distribution of hospitalized and recovered, other categories are too low distribution.

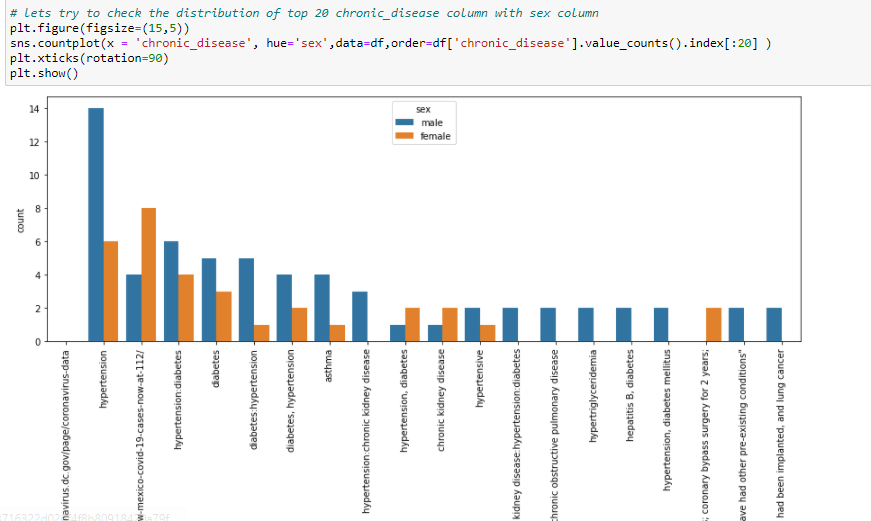


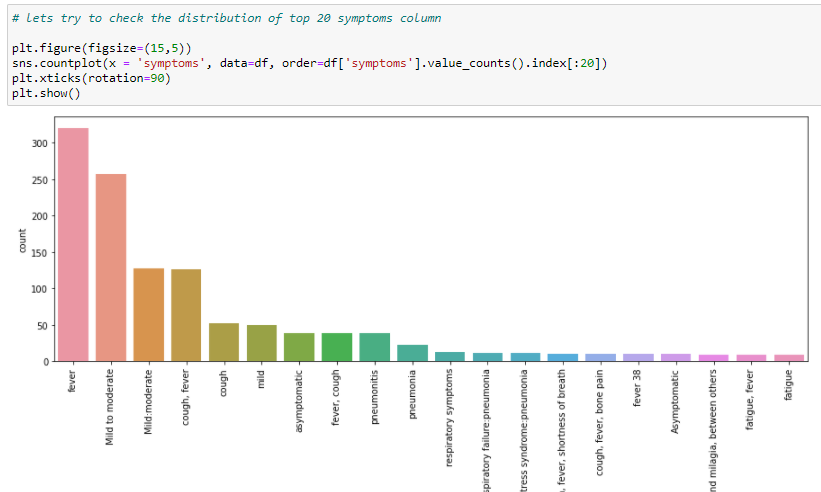
From above graph we check the distribution of outcome against sex column, so we can see that mostly male patients hospitalized as compare to female.



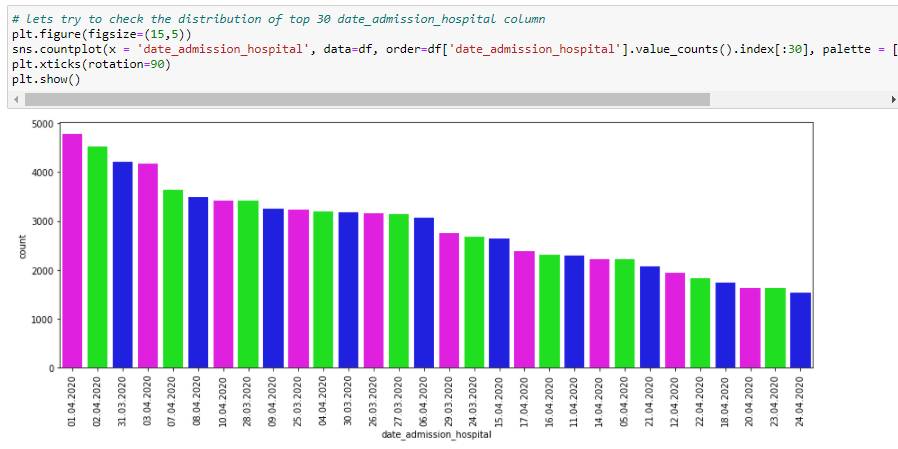
From above graph we can see the distribution of male is greater as compare to female.

From above graph we plot the top 20 chronic diseases in which also we can see that some invalid values but we also see the distribution of hypertension and others chronic disease in graph.

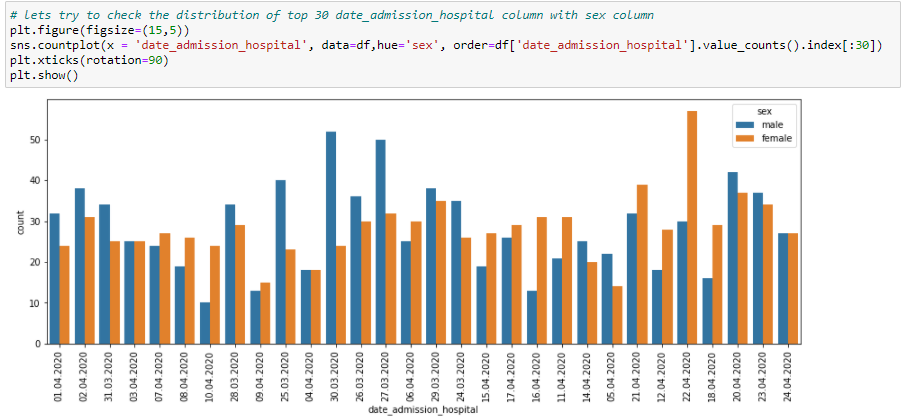
From above we also check the chronic disease against sex so, in this distribution male mostly affected on hypertension chronic disease.

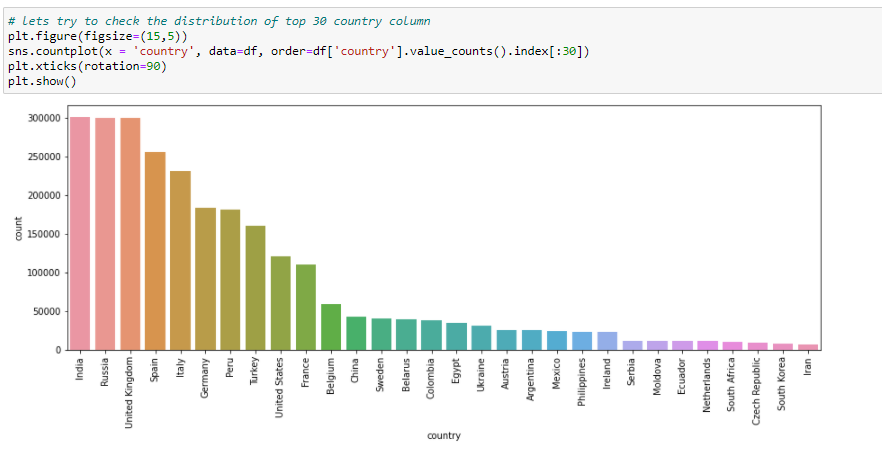


From above graph we check the top 20 distribution of symptoms in covid19 data so, we can see that fever symptom mostly distributed as compare to others symptoms.

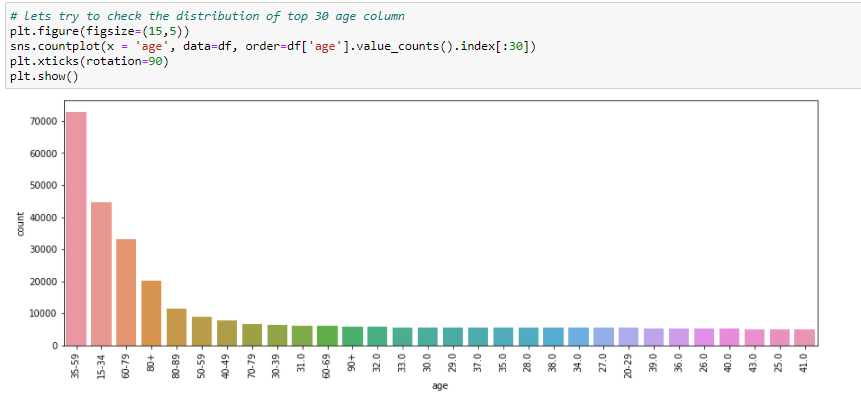


From above graph we plot the top 30 dates and check the distribution of each date so, we can see that patients mostly admitted at 01/04/2020.

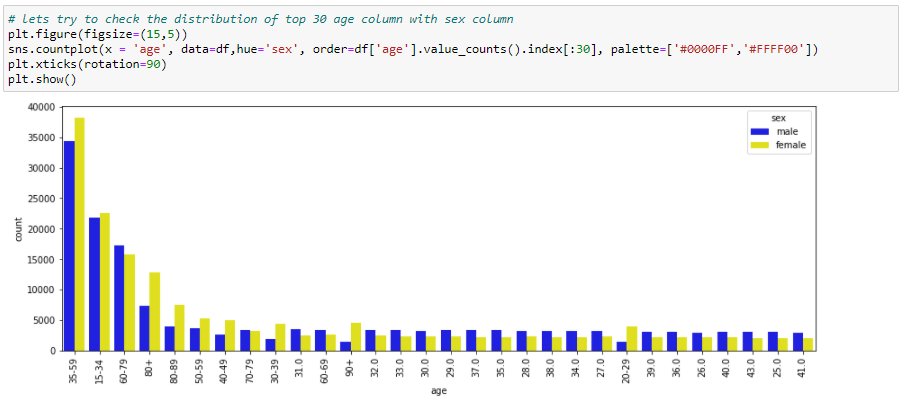
From above we also check which gender admitted mostly so, we can see that mostly male admitted as compare to female.



From above graph we plot the top 30 counties that are more effected by covid, India,Russia, UK are more effected as compare to others countries.



From above graph we can see that mostly effected people age range is 35-59.

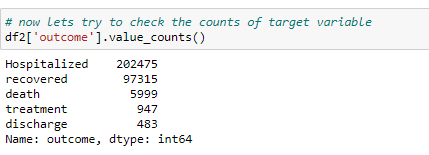


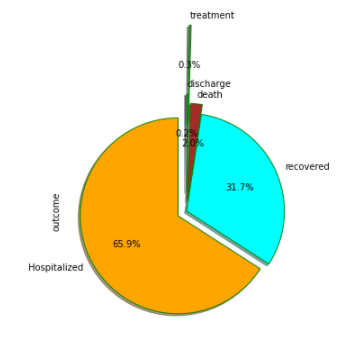
From above graph we check the age against gender. So, we can see that mostly female effected as compare to male.

**Feature Engineering/ Preprocessing**

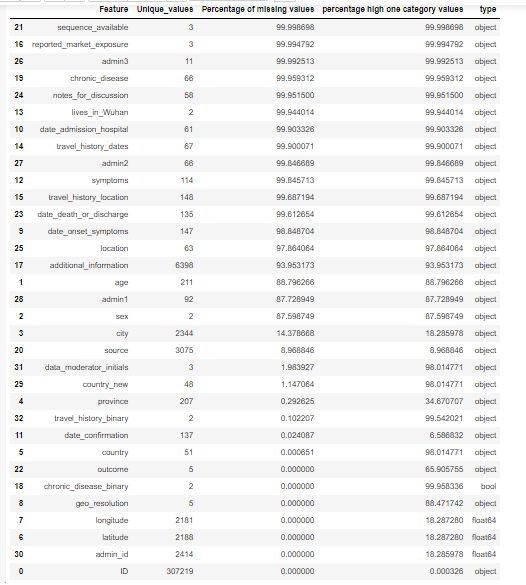
In this section, we will preprocessed our data and remove the columns that are not useful for our anylsis and will also handle this missing values but most important to handle the target variable outcome because we have a high cataegories values inside outcome column so, we will reduce the outcome and will extract the best categories for our prediction.

We extracting the data where outcome value are not null and also feature engineering on target column to replace all the categories into specific category that shown in below:



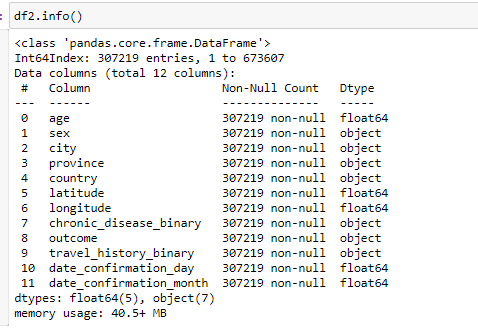


From above pie chart we can see that mostly patient hospitalized and recovered.

Now, we will remove all the columns where columns missing values greater then 90% and high one category values greater then 90%. 

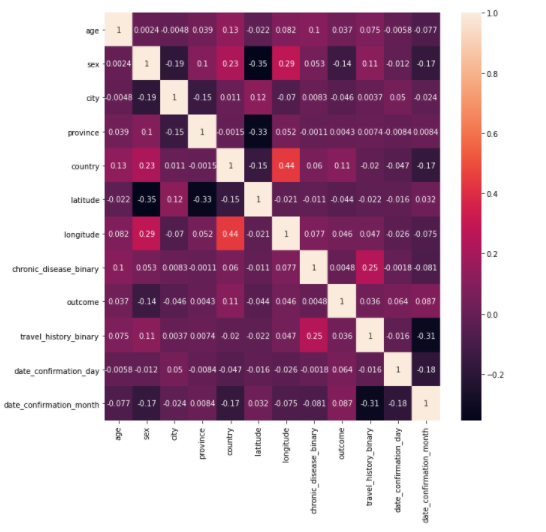
From above graph we remove the columns where both missing values and high one category values greater then 90%.

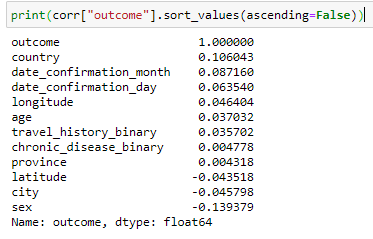
So , we fill the missing values of numeric column into mean and categorical column fill with Unknown value.



From above fig we can see that no missing values are present in our data.

We also check the correlation of each column that shown below:



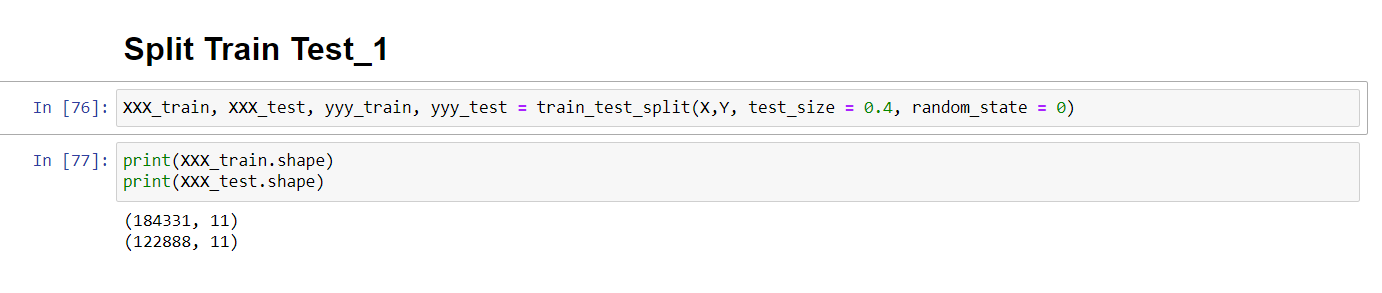


We also **label encoding** our categorical features in data. Categorical data refers to the information that has specific categories within the dataset. In this malware dataset above, there are many columns are categorical variables.

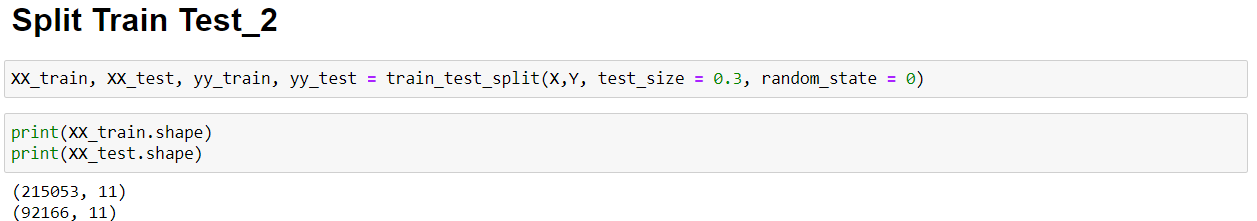
Machine Learning models are primarily based on mathematical equations. Thus, you can intuitively understand that keeping the categorical data in the equation will cause certain issues since you would only need numbers in the equations

**Train the ML models based on three different splits and discuss the variation in accuracy/ score obtained from the models in the training as well as testing?**

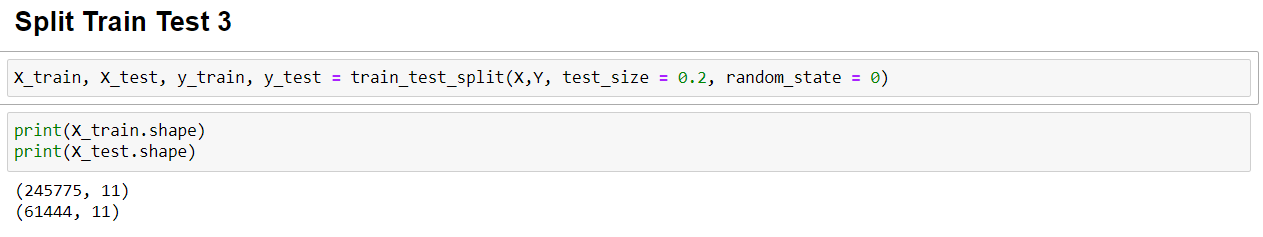
First time we split our data in 60% data for training and 40% for testing. Every dataset for Machine Learning model must be split into two separate sets – training set and testing set.



Second time we split our data in 70% data for training and 30% for testing.



Third time us splitting the 80% data for training and 20% for testing.

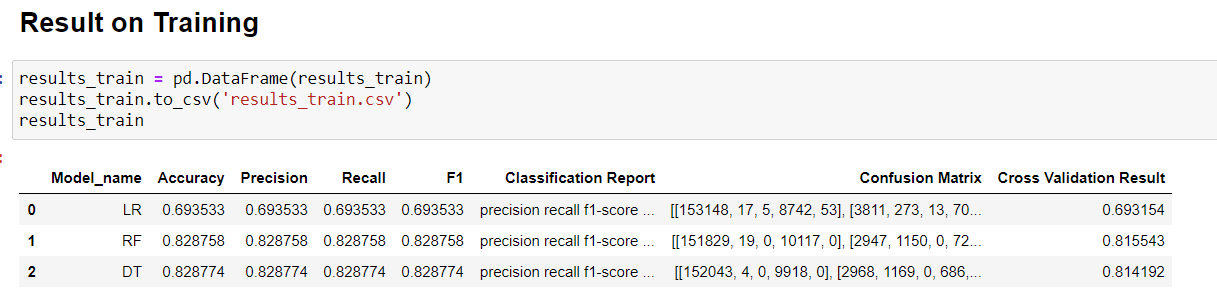


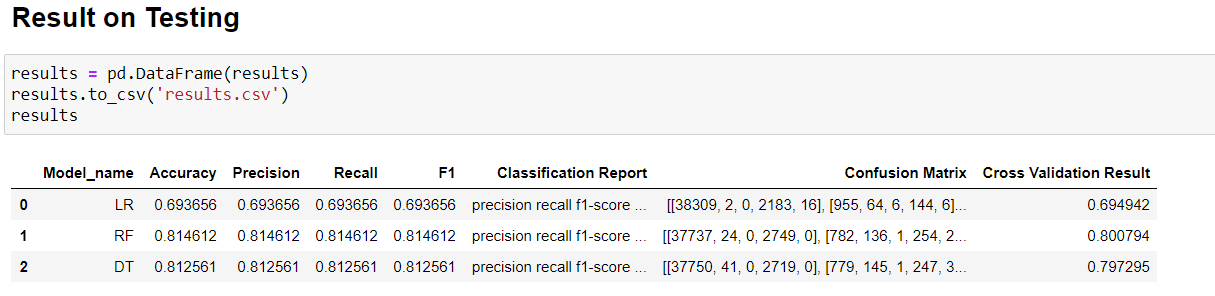
Usually, the dataset is split into 70:30 ratio or 80:20 ratio. This means that you either take 70% or 80% of the data for training the model while leaving out the rest 30% or 20%. The splitting process varies according to the shape and size of the dataset in question but as a requirement we split our dataset in three different splits.

After splitting dataset we are doing **standard scaling** our data. Feature scaling marks the end of the data preprocessing in Machine Learning. It is a method to standardize the independent variables of a dataset within a specific range. In other words, feature scaling limits the range of variables so that you can compare them on common grounds.

We also apply **the K Fold cross validation** to check the performance on each models. Cross-validation is a statistical method used to estimate the skill of machine learning models.

It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

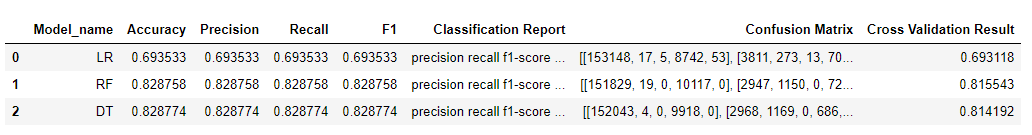




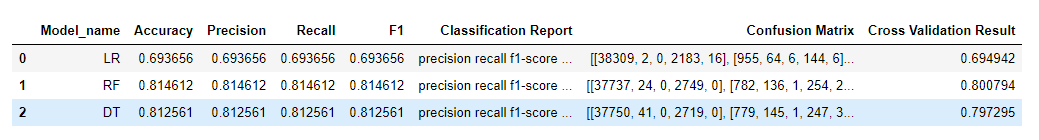
From all of three splits, the dataset which is split into 80:20 ratio give high accuracy as compare to other two split. As we can see that Random Forest giving good performance on training data and testing as compared to Decision Tree Classifier and Logistic Regression.

**Interpret the results based on problem specification and objectives. The ML modelling results should neither overfitted nor underfitted. Justify with arguments?**

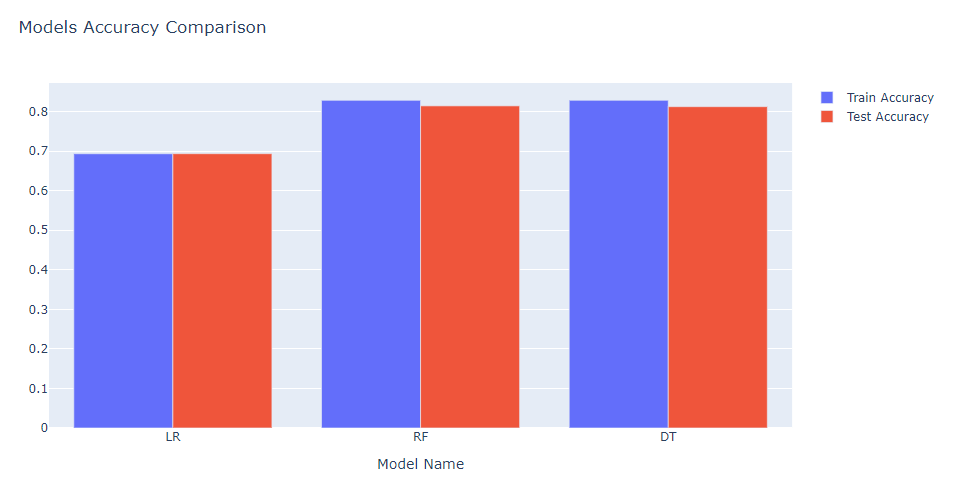
Now we check the performance of each models on training and testing data.

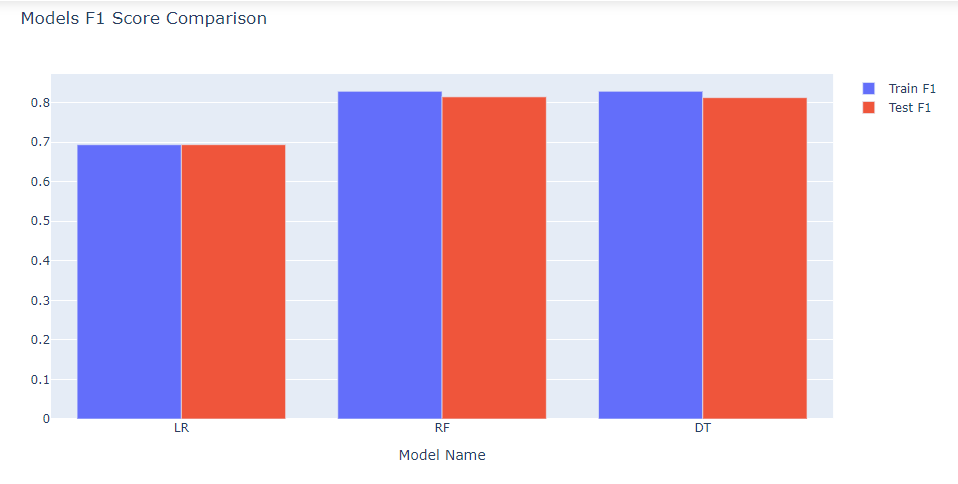


From above fig we can see that Random Forest giving good performance on training data as compare to other classifiers.

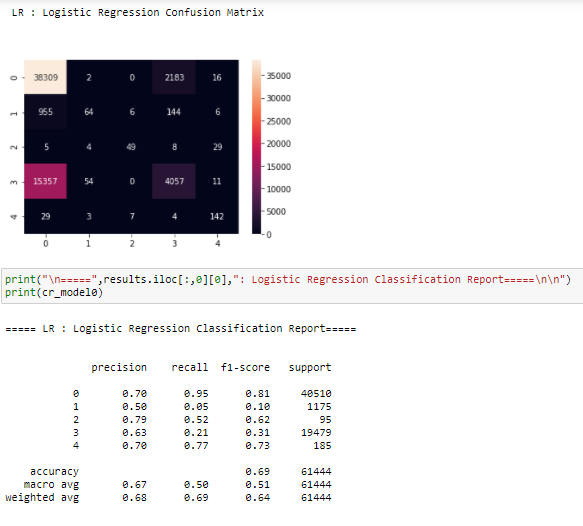


From above fig same Random Forest also giving good performance on testing data as compare to others.

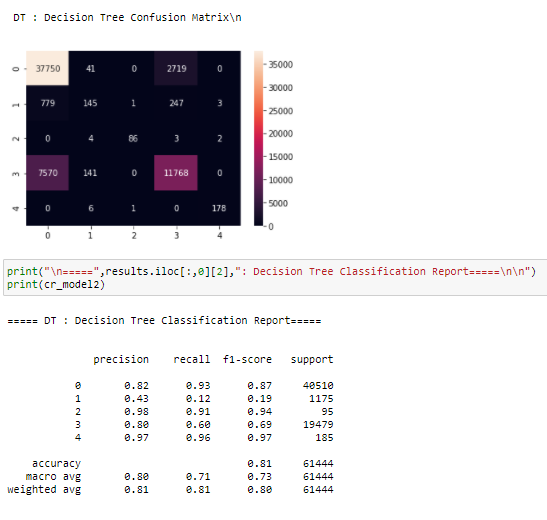




We also check the confusion and classification report of each models that are given below:







As we know Overfitting is when the model's error on the training set (i.e. during training) is very low but then, the model's error on the test set (i.e. unseen samples) is large!

 Underfitting is when the model's error on both the training and test sets (i.e. during training and testing) is very high.

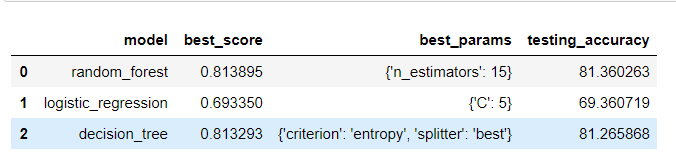
But our model is free from Overfitting and Underfitting because it giving good accuracy in training data as well as in testing data. As we seen above in classification report or in confusion matrix. And also we do feature engineering and apply many preprocessing techniques as well. Due to this our model is not underfitt or overfitt as well.

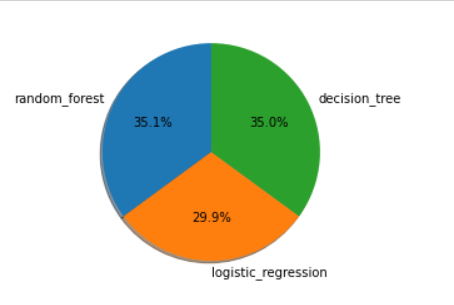
**Grid Search CV (Hyper parameters Tuning)**

We also apply the Grid Search technique to get the best parameters of each models and best scores of each models. Grid Search CV was the greatest invention of all time. It runs through all the different parameters that is fed into the parameter grid and produces the best combination of parameters, based on a scoring metric of your choice. Obviously, nothing is perfect and Grid Search CV is no exception:

* “best parameters” results are limited
* process is time-consuming

The “best” parameters that Grid Search CV identifies are technically the best that could be produced, but only by the parameters that you included in your parameter grid.





**Provide the explanation of code that will be used to solve the problem. Comments must be provided along with code?**

1st: We install important libraries which we need to load data and perform some calculations through these libraries.

2nd: Embedding these libraries we load our dataset and after that we check the correlation with other features and perform visualization to understand the data more correctly and easy.

3rd: We have many categorical column in our dataset, first we separate them and then we convert them into numerical by using one hot encoding technique. And we also fill the missing values in our dataset by using fillna and apply many visualization techniques to find different insights from the data.

4rd: We split our data in (x, y) in three different splits and after that we apply holdout method on (x, y) and due to variance in data we minimize our variance by using StandardScaler and cross validation techniques these are all also called preprocessing techniques.

5th: Now we fit our models (Random forest, Decision tree, and Logistic regression) in our data.

6th: We evaluate our model means we test our model by predicting some values. And also checking the accuracy and errors of our models.

7th: We use Grid Search technique to improve model performance. In this we basically do hyper parameter tunning.Model hyper parameter are used to optimize the model performance.

**Conclusion**

We conclude that random forest giving the good performance as compare to decision tree classifier and logistic regression on different metrics e.g. recall, precision, f1score, cross validation and hyper parameters tuning. So we will choose the Random Forest for our final prediction.

And also we will choose the dataset which have the ratio of the 80% data for training and 20% for testing as compare to other two splitting dataset.