**Customer Churn Analysis**

**Problem Definition:**

Customer churn prediction is used in many businesses to evaluate a company’s loss rate. Customer churn occurs when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn. keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals. Predicting the churn rates accurately is important as it helps the business in better understanding future expected revenue. It can also help in identifying mistakes and improve in areas where there is a lack of customer satisfaction.

Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can prioritize focused marketing efforts on that subset of their customer base.

Customer churn rate can be calculated by dividing no of customers lost in a given time interval by the total no of customers multiplied by 100. For example: If there are 200 customers in a company and it has lost 5 customers in a month then that month’s churn rate will be (5/200)\*100 = 2.5%. So that month’s churn rate is 2.5% of that company.

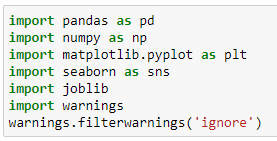
Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

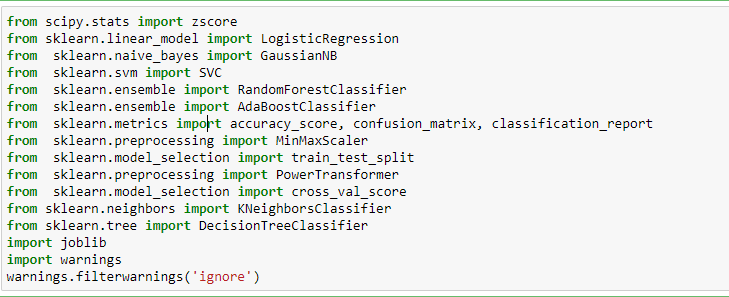
In this article, we will examine data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

The link to find the .csv file is: <https://github.com/dsrscientist/DSData/blob/master/Telecom_customer_churn.csv>

**Importing the data:**

We need to import all the relevant libraries:





We have now imported all the important libraries that we will be needing during analysis.

We need to import the .csv file into the Jupyter notebook as shown below. 

This data set contains both Independent and Dependent (target) variables.

Independent variable: They are also known as Input variables. These are the input for a process that is being analyzed.

Dependent variable: They are also known as Output or Target variables. They are dependent on Independent variables for their outcome

After importing the dataset, display a sample of data. The variables in the dataset are as follows:

* customerID
* gender
* SeniorCitizen
* PartnerDependents
* Tenure
* PhoneService
* MultipleLines
* InternetService
* OnlineSecurity
* OnlineBackup
* DeviceProtection
* TechSupport
* StreamingTV
* StreamingMovies
* Contract
* PaperlessBilling
* PaymentMethod
* MonthlyCharges
* TotalCharges
* Churn

**Data Analysis (EDA):**

**Now we need to understand the dataset by performing Exploratory Data Analysis.**

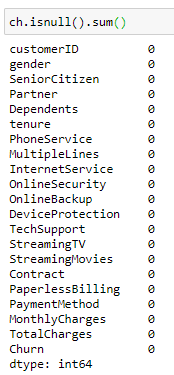
Let’s check the shape of the data set:



We can see that there are 7043 rows and 20 columns in the dataset.

We cannot have null values in the data as this will affect the data and eventually, the predicted result will not be correct. Therefore we must check for any null values in the dataset.

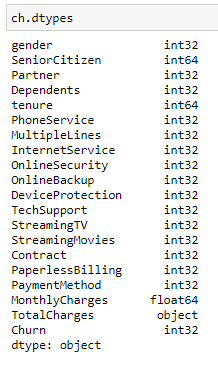
Checking for null values in the data set:



We can see that there are no null values in the dataset and we are good to go.

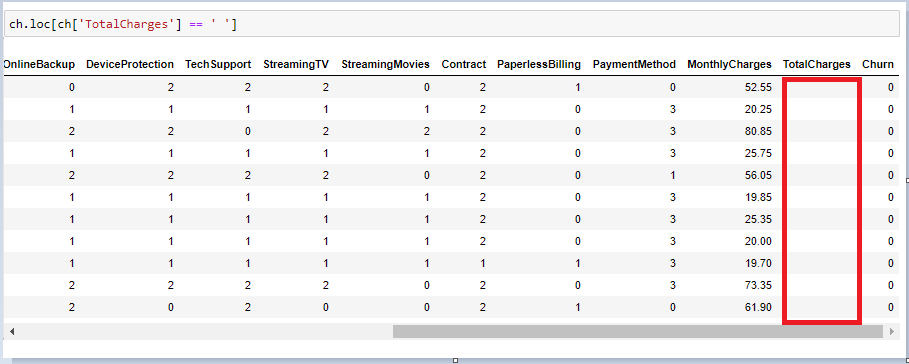
we can drop ‘customer id' Column as each customer id is unique and it won't be helpful.





**Handling Missing data:**

In the ‘TotalCharges’ Column we can see that few values are missing. Also from the above, we can see that the TotalCharges column is in object type but it has numerical values. Therefore we have to handle the missing values to convert them into numerical data type (int or float data type)



From the above figure, we can see that there are 12 missing values in the ‘TotalCharges’ column as highlighted. As the dataset has 7043 rows we can drop these missing values as it will have a negligible effect on the dataset.



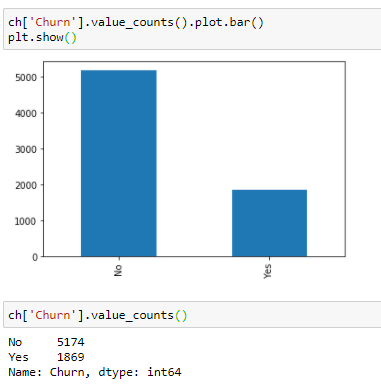
Also as the ‘TotalCharges’ column is in object data type and it contains numerical values or continuous values, we must convert it into the ‘TotalCharges’ column.



**Data Visualization and EDA Concluding Remarks:**

In the given data, ‘churn’ feature is the Target feature or variable. The unique values of this feature are only 2 i.e Yes and No, which means it has only two classes. So, as there are only two unique values this is a ‘Classification Problem.’

The dependent or target variable has 5174 No’s and 1869 yes cases which can be seen below in the form of value counts and bar chart.



This means that 5174 customers were not churned (retained) and 1869 customers were churned.

We can even get the Churn rate from this by dividing no of customers lost in a given time interval by the total no of customers multiplied by 100.

In this case, about 73.46% are retained and the churn rate is 26.53%.

Comparing the Independent variables with the dependent variable (churn):



As shown above, we can see that there’s a very slight difference between the two genders, which says that gender doesn’t play a role in customer churn.



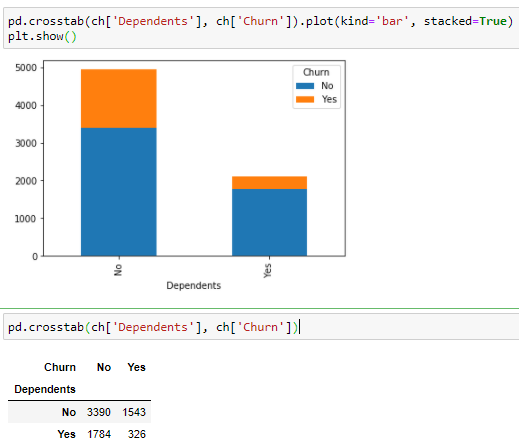


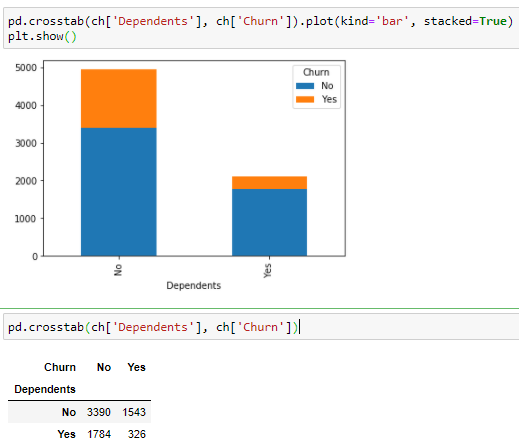
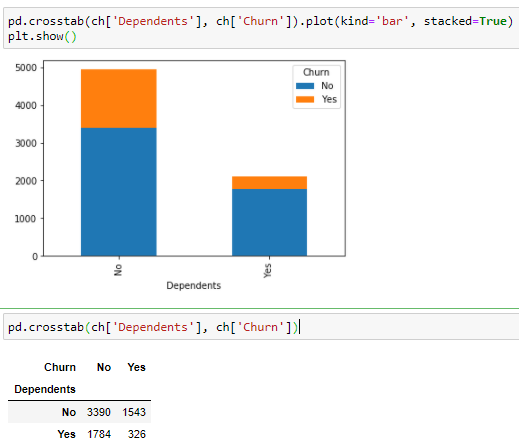
We can see that the churn is lesser if the customer is a senior citizen as they might have to start searching for a new company which would again take a lot of time and thus think it would be better to stick to the present company.





We can see that churn is less if there’s a partner. There might be instances where there would be one or more partners who would not agree to leave the company because of differences in opinion. That’s why maybe the churn is less where there’s a partner.



Here we can see that if there are dependents then the churn is less when compared to if there are no dependent. The reason might be similar to partner one.

We can see that the churn is more if there’s a phone service.



Here also we can see that the churn is less where there is no phone service.



Here we can see that most customers that churned had the Fiberoptic internet service. Maybe the company Should have only DSL internet service.

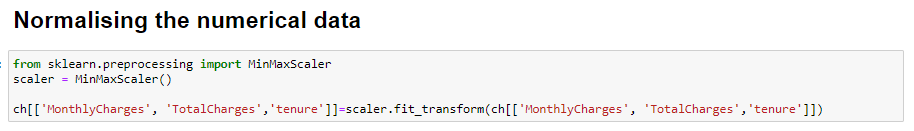
**Pre-processing Pipeline:**

The data set has variables in both object type and numerical type (int and float)

Therefore we have to pre-process the data to move forward.

All the float type or int type variables should be converted into the same scale since the range of values of raw data varies widely, in some machine learning algorithms, objective functions do not work correctly without normalization.

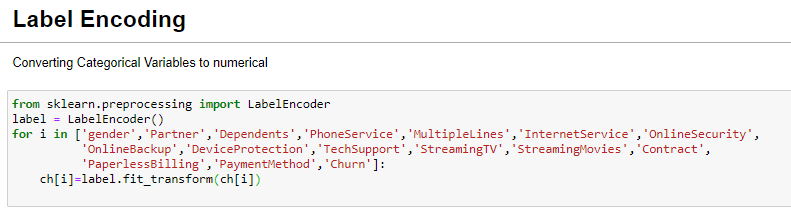
Therefore normalization is to be performed only on the numerical type (int and float type) variables.



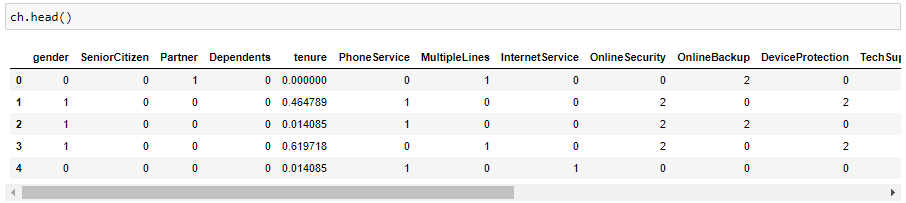
After the Normalising is done all the numerical data fall under the same range or scale

We can see that majority of variables are object-type. These variables contain string data that cannot be passed into the machine learning model as it won’t be able to recognize string data type. It only recognizes numerical data.

Therefore we need to convert the string data into numerical data. This can be done by manually encoding or by using an encoder such Label Encoder, one-hot encoder etc. For example: The target variable churn consists of only two unique values, Yes & No. after encoding this will get converted to 0 and 1. Similarly, if there are three unique values then it will be converted to 0,1, and 2.

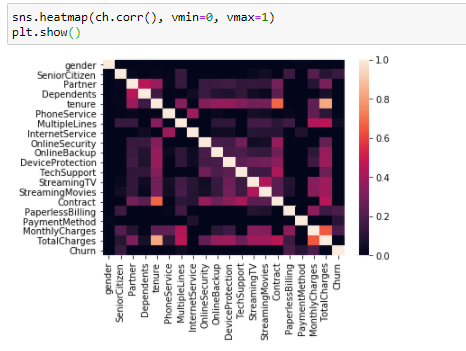


Sample data after preprocessing:



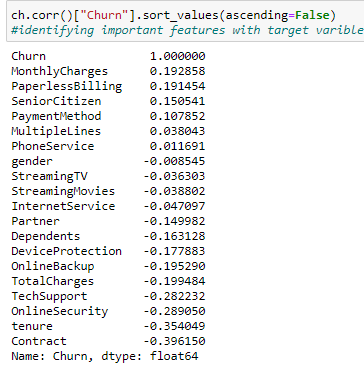
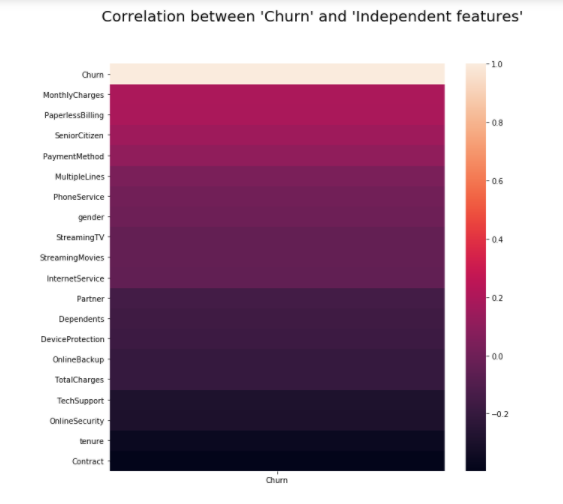
**Correlation between ‘Churn' and 'Independent features'**

Now we can check the correlation between all the variables. (Note: correlation of all independent variables can be only done after encoding as correlation does not consider string values)



We can see that correlation between independent variables is low(i.e. <0.7). We are good to go.

From the figure below we can see the important features in descending order from top to bottom.

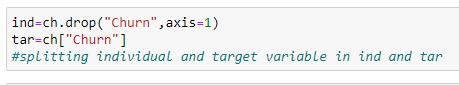
 

We can see that MonthlyCharger and PaperlessBilling are the top two important features for the taget variable ‘churn’

Now the preprocessing is completed. We now have to move to data modeling and prediction

**Building Machine Learning Models:**

We have to now split the data into independent and target variables.



Here the target variable is ‘Churn’ and the rest of them are independent variables.

We have to now split the independent and target variables into training and testing datasets as shown below.

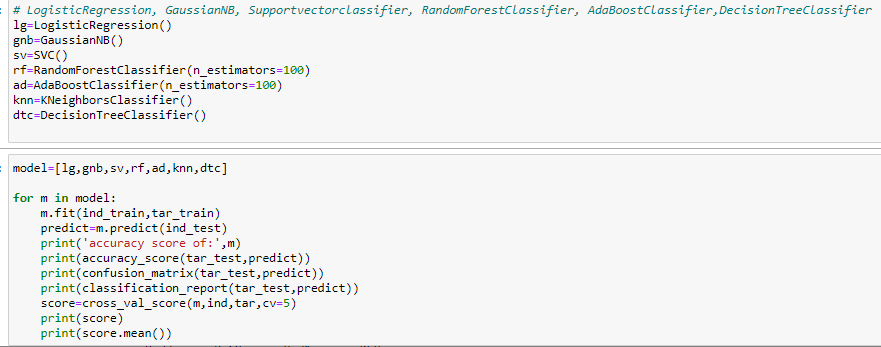


We will use a machine-learning algorithm to learn from the training set and use the model to predict the testing set and compare it with the predicted data with the target testing set to know how close the values. If the error between the predicted and target testing data is less that means the accuracy of the model is high and we can use this model to predict the result of similar datasets.

In this, we have used 7 Machine learning Algorithms

* LogisticRegression
* GaussianNB
* Supportvectorclassifier
* RandomForestClassifier
* AdaBoostClassifier
* KNeighborsClassifier
* DecisionTreeClassifier

We can train and predict the data using the above 7 ML algorithms and save the model which has the highest frequency.



**Cross Validation:**

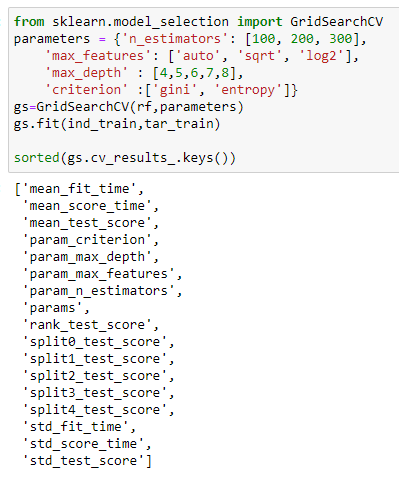
K-Folds cross validation is one method that attempts to maximize the use of the available data for training and then testing a model. It is particularly useful for assessing model performance. "Cross\_val\_score" splits the data into say 5 folds, then for each fold, it fits the data on 4 folds and scores the 5th fold. Then it gives you the 5 scores from which you can calculate a mean and variance for the score. It is useful to tune parameters and to get an estimate of the score.

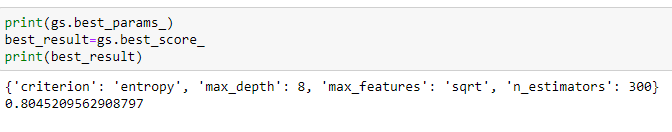
Accuracy score, Cross\_Validation score and Standard Deviation are given in the below table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Machine Learning Model** | **Accuracy** | **Cross\_Validation score** | **Standard Deviation** |
| LogisticRegression | 0.801 | 0.80276 | 0.011329 |
| GaussianNB | 0.753 | 0.75128 | 0.00988 |
| Supportvectorclassifier | 0.788 | 0.791811 | 0.011381 |
| RandomForestClassifier | 0.794 | 0.795646 | 0.011311 |
| AdaBoostClassifier | 0.792 | 0.800765 | 0.014008 |
| KNeighborsClassifier | 0.757 | 0.756399 | 0.008193 |
| DecisionTreeClassifier | 0.731 | 0.739906 | 0.016644 |

According to Cross val score and accuracy we can see that the RandomForesrClassifer has the least difference between Accuracy and Cross val score, therefore we select RandomForesrClassifer model.

We can use Gridsearch CV to get the best parameters of the selected model as shown below:





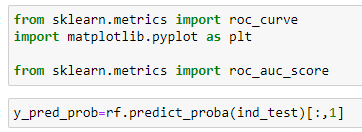
We can see that by using the best parameters the accuracy score of the model has increased.

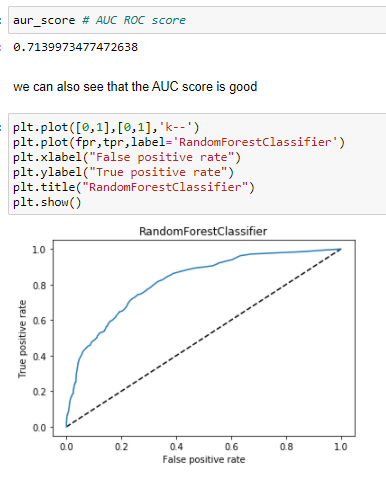
# AUC ROC CURVE:

# AUC: Area Under the curve;   ROC: Receiver Operator Characteristic

# The greater the ROC score the better is the model. If ROC=1, then it perfectly fits.

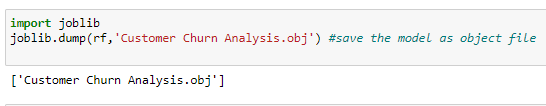
# If the maximum of the area falls under True positive then the model is doing good.





We can see that the ROC Score is 0.71 and the area under the curve falls under True Positive Rate, Therefore, we can conclude that the model is performing well and we need to save the model in .obj file for future use.

By using ‘dump’ from joblib we can the best model (Randomforectclassifier in our case) in .obj as by doing this we can save the model and by using ‘load’ function we can load the model and predict the model with a different dataset.



**Concluding Remarks:**

From the above results of the data modeling and prediction we can see that the Decision Tree Model is performing well as the accuracy score, cross val score and Roc score are good also the maximum of the area under the curve fall under true positive rate. Therefore we can save the model as .obj file so that it can be used to predict the result of the different data sets.

In this kind of problems Pre-processing and data-cleaning is the most important thing. We need to handle both the categorical and numerical data properly and also need to check by building different ML model on the same dataset. We need to check accuracy and cross val score of each model and chose the one which has the best of the same and also which has the least defference between them (i.e cross\_val\_score and accuracy).