# CUSTOMER CHURN ANALYSIS

Submitted by:

Poojashree D S

# Problem Definition

The aim of the project is “to provide a solution for the prediction of customer churn in telecom sector using machine learning in big data platform” involving the background of the research, problem statement, aims and objectives of the research, research questions, significance of the research, as well limitations of the research

Machine learning can possibly be the sort of tools which could help telecom companies in the churn prediction model. Machine learning is a kind of artificial intelligence tool which gives the capability to let a computer learn the algorithm instinctively without human contribution. Comparatively, churn prediction in telecom has been considered as a unique application domain to churn prediction than other subscription based industries as a result of the variety, volume and biases of the information.

This research investigates briefly about techniques involved in machine learning algorithms for acquiring accurate prediction levels of telecom customers.

To compare models and select the best for this task, the accuracy is measured. Based on other characteristics of the data, for example the balance between classes (number of “churners” vs. “non-churners” in data set) further metrics are considered if needed.

The major contribution of the current study is to present a model of churn prediction that helps telecom companies to find users who are more intending to churn.

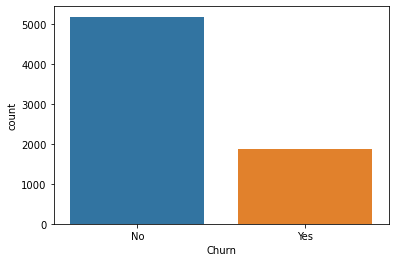
**Data Analysis**

The crucial element in machine learning tasks for which a particular attention should be clearly taken is the data. Indeed the results will be highly influenced by the data based on where did we find them, how are they formatted, are they consistent, is there any outlier and so on. At this step, many questions should be answered in order to guarantee that the learning algorithm will be efficient and accurate. Many sub steps are taken to get, clean and transform the data. I am going to explain each one of them to show how they have been applied to my project and why they are useful for the machine learning part.

We will go step by step for building a machine learning algorithm for the prediction of loan defaulters based on certain variables present in the dataset. Our main goal is to correctly identify defaulter's (True positives) so that the lending club can decide whether a person is fit for sanctioning a loan or not in the future.

 This dataset contains a total of 7,043 customers and 21 attributes, coming from personal characteristics, services signatures, and contract details. Out of the entries, 5,174 are active customers and 1,869 are churned, which demonstrates that the dataset is highly unbalanced. The target variable for this assessment is going to be the feature Churn.

The main goal is to develop a machine learning model capable of predicting customer churn based on the customer’s data available. I will use mainly Python, Pandas, and Scikit-Learn libraries for this implementation.



From the graph on the above, we can see that the dataset is highly unbalanced, containing 73.5% of ‘No’ entries which means active accounts and 26.5% of ‘Yes’ which means defeated accounts. The attributes explain as follows

PhoneService — Whether the customer has a phone service (Yes, No)

MultipleLines — Whether the customer has multiple lines (Yes, No, No phone service)

InternetService — Customer’s internet service provider (DSL, Fiber optic, No)

OnlineSecurity — Whether the customer has online security (Yes, No, No internet service)

OnlineBackup — Whether the customer has online backup (Yes, No, No internet service)

DeviceProtection — Whether the customer has device protection (Yes, No, No internet service)

TechSupport — Whether the customer has tech support (Yes, No, No internet service)

StreamingTV — Whether the customer has streaming TV (Yes, No, No internet service)

StreamingMovies — Whether the customer has streaming movies (Yes, No, No internet service)

Tenure — Number of months the customer has stayed with the company

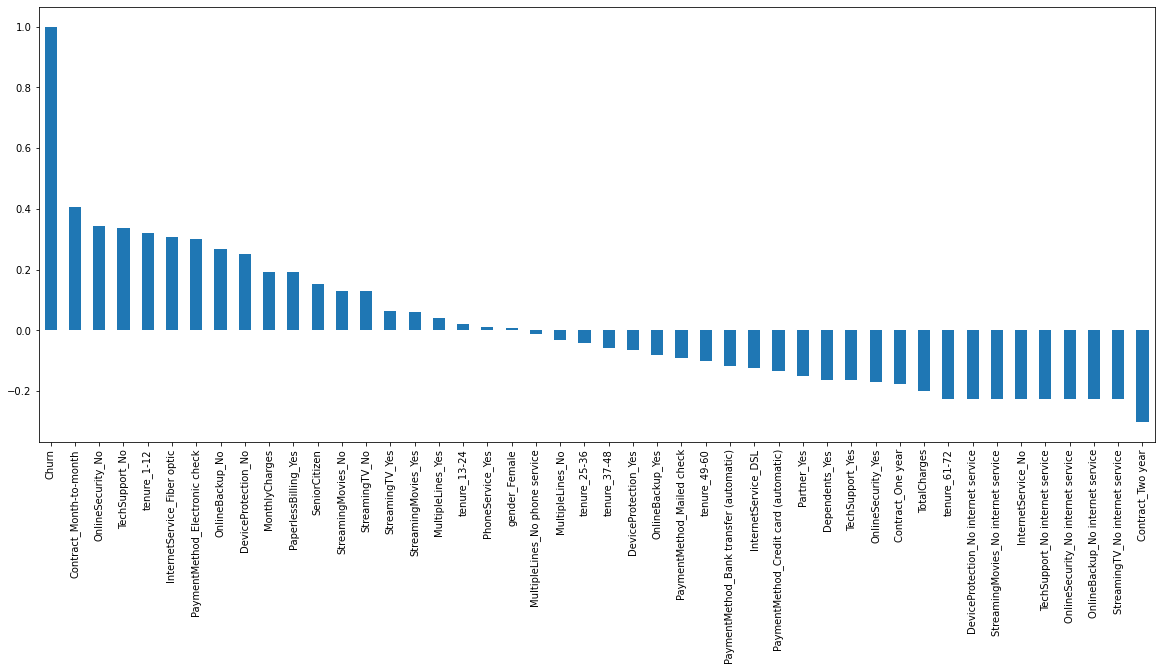
Contract — The contract term of the customer (Month-to-month, One year, Two year)

PaperlessBilling — Whether the customer has paperless billing (Yes, No)

**Exploratory Data Analysis**

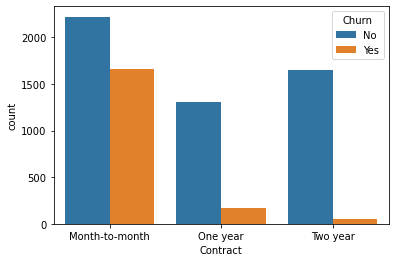
As the purpose of this experiment is to **identify patterns** that can yield to customers churn, I will be focusing mainly on the churn portion of the dataset for the exploratory analysis.

Churn is the target variable. The below charts represent the how much Churn depend on the feature variables

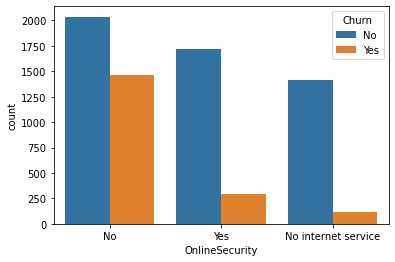
****

The above chart shows the Churn is more dependent on the contract\_Month-to-month, OnlineSecurity\_No and no Tech support.

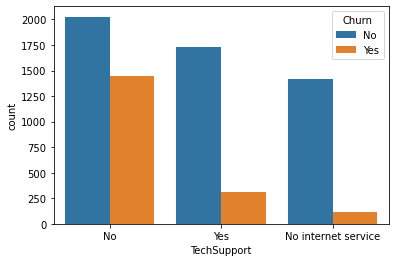
Less Churn when phone service is there.



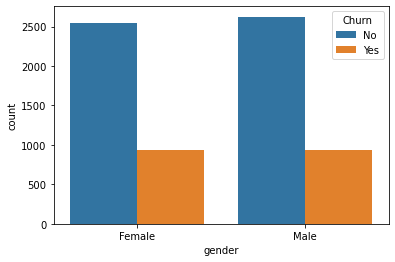
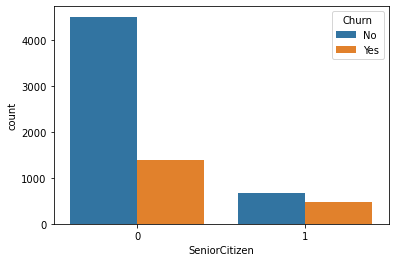
From the above graph we can see that the one who have one month contract is more likely to Churn

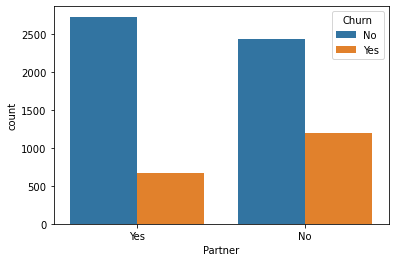
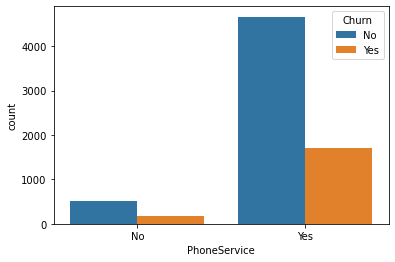


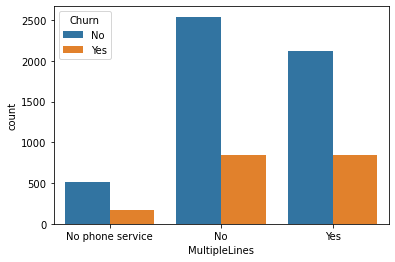
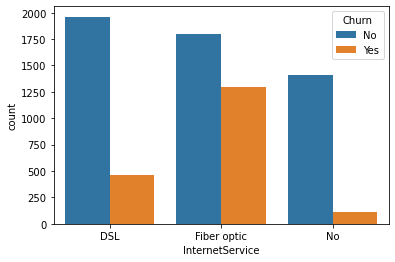
From the above graph we can see that the one who have no online security is more likely to Churn than the one who have

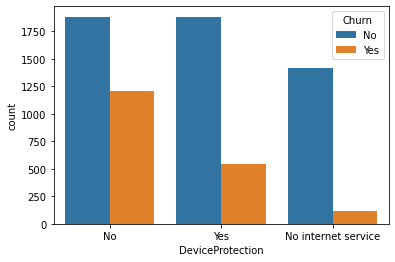


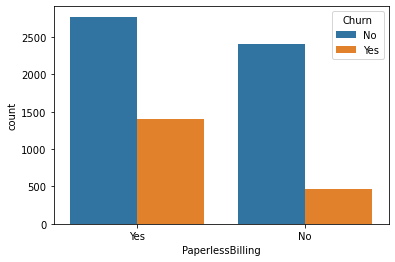
From the above graph we can see that the one who have no Tech Support is more likely to Churn than the one who have Tech support

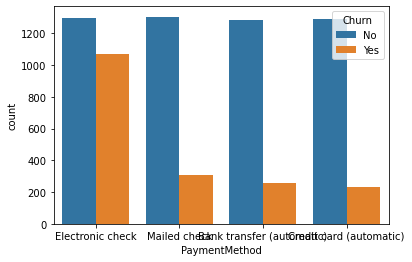
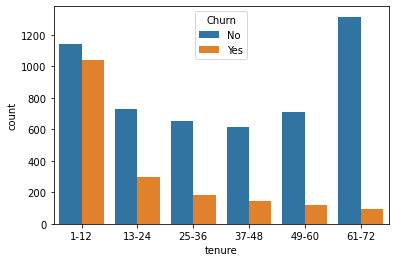
 

** **

**** ****

****

****

**** ****

dddcccccccc

ddd

* Senior citizens' churn rate is much higher than non-senior churn rate.
* Churn rate for month-to-month contracts much higher than for other contract durations.
* Moderately higher churn rate for customers without partners.
* Much higher churn rate for customers without children.
* Payment method electronic check shows much higher churn rate than other payment methods.
* Customers with InternetService fiber optic as part of their contract have much higher churn rate.
* The majority of customers that cancel their subscription have *Month-to-month*Contract type and Paperless Billing enabled
* Customers that have Payment Method as *Electronic Check*are more likely to leave

**Pre-processing**

Data Preprocessing includes data cleaning in which we manually remove unwanted columns. In this particular project we remove LoanID as it is not required.

First we have import pandas to load dataset which is present in the csv format , the following code is used to read the file

df = pd.read\_csv('loan\_prediction.csv')

After data collection, several steps are carried out to explore the data. Goal of this step is to get an understanding of the data structure, conduct initial preprocessing, clean the data, identify patterns and inconsistencies in the data (i.e. skewness, outliers, missing values) and build and validate hypotheses.The dataset consists of null values. I replaced the null values with their mean values and their mode . Mean is nothing but the average value whereas median is nothing but the central value and mode the most occurring value. Replacing the categorical variable by mode makes some sense.

For handling categorical variables, there are many methods like One Hot Encoding or Dummies. In one hot encoding method we can specify which categorical data needs to be converted . However, as in my case, as I need to convert every categorical variable into numerical, I have used the get\_dummies method.

Then I remove columns which are giving the same information.

**Split train and test data**

The dataset was split into 70% for training and 30% for testing. The training set will be used to generate the model the chosen algorithms will use when exposed to new data**.** The test set is the final dataset I will touch to measuremodel performance based on some metrics.To split,train and test data i imported train test split.

**from** **sklearn.model\_selection** **import** train\_test\_split

train\_test\_split will split the data into 70% for training and 30% for testing.

**Scaling of the data**

Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.I used the standard scalar to scale the data, then I started to build the model by splitting the model to train and test.

**Balancing the Data**

The data is highly imbalanced.it can reduce the accuracy score or model may give biased output so to avoid this we have to balance the data.

I use SMOTE to balance the data. It increases the number of cases to balance the data so the model is not biased.

As the data cleaning and data structuring are done, we will be going to our next section which is nothing but Model Building.

**Building Machine Learning Models**

For performance assessment of the chosen models, various metrics are used. I used accuracy score, confusion matrix and classification report.

· Accuracy score is the ratio of number of correct predictions to the total number of input samples.

· A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

The classification report visualizer displays the precision, recall, F1, and support scores for the model.

**Models**:

In this experiment, I applied eight different ML algorithms to analyze and compare the Accuracy score obtained by each of them. Those are listed below:

* LogisticRegression
* DecisionTreeClassifier
* SVC
* RandomForestClassifier
* KNeighborsClassifier
* AdaBoostClassifier
* GradientBoostingClassifier
* lightgbm

**import** **lightgbm** **as** **lgb**

lgm = lgb.LGBMClassifier(silent=**False**)

lgm.fit(x\_train,y\_train)

predl = lgm.predict(x\_test)

print('accuracy score :',)

print(accuracy\_score(y\_test,prede))

print(confusion\_matrix(y\_test,prede))

print(classification\_report(y\_test,prede))

accuracy score :

0.8441223832528181

[[1302 219]

[ 265 1319]]

precision recall f1-score support

0 0.83 0.86 0.84 1521

1 0.86 0.83 0.84 1584

accuracy 0.84 3105

macro avg 0.84 0.84 0.84 3105

In the lightgbm True positive values are 1302 and True negative values are 1319, which shows it is a pretty good model.

## 

## Cross validation:

I applied cross-validation method which is a technique that partitions the data into subsets, training the data on a subset and use the other subset to evaluate the model’s performance.

The top performance is given by the RandomForestClassifier, and the difference between the Accuracy Score and cross validation is very less so this is not due to oversampling or undersampling. But there is still room for optimization.

scr = cross\_val\_score(lgm,x,y,cv=5)

print('coss validation score is',scr.mean())

coss validation score is 0.8369899342257339

**Hyper Parameter Tuning**

To improve overall performance I tuned classifiers parameters using GridSearchCV for lightgbm,I used parameters learning\_rate, max\_depth and num\_leaves to tune the model.

final\_mod =lgb.LGBMClassifier(learning\_rate= 0.01,max\_depth= 13,num\_leaves= 1200)

final\_mod.fit(x\_train,y\_train)

predict= final\_mod.predict(x\_test)

accu = accuracy\_score(y\_test,predict)

print(accu)

0.8421900161030595

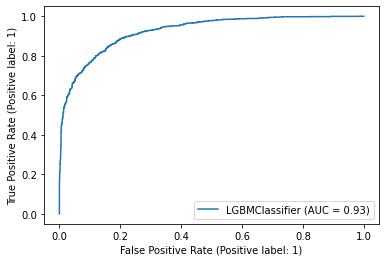
The model Accuracy score shifted from 0.82 to 0.84.

This means the implementing ML model based on lightgbm delivers 84% of accuracy while predicting customer Churn.

**Saving the model**

I saved the model for future use in joblib as 'telight.pkl' and loaded the saved model and saved the prediction values in the csv file.

**AUC-ROC Curve for the Model**

****

The AUC ROC score for this model is 0.791

# Conclusion

The aim of the project is building and comparing several customer churn prediction models.

In EDA we remove all the null values present and replace them with the mean and mode of the column

Using the visualization we are able to conclude the relation between the data

Finally able to build the model and cross validate it, balances it and hyper tuned the model

And lastly saved the model in joblib

No algorithm will predict churn with 100% accuracy. There will always be a trade-off between precision and recall. That's why it's important to test and understand the **strengths and weaknesses**of each classifier and get the best out of each.

If the goal is to engage and reach out to the customers to prevent them from churning, it's acceptable to engage with those who are mistakenly tagged as ‘not churned,’ as it does not cause any **negative impact**. It could potentially make them even happier with the service. This is the kind of model that can **add value** from day one if proper action is taken out of meaningful information it produces.