Project: Telecom Churn Prediction

Aim:

To predict the customers who are likely to churn in the next N months & facilitate in taking business actions for reducing the churn.

Objective:

In any service providing industry, when a customer decides to stop using the service either by cancelling the subscription or not paying for the service, we call this customer churn.

Churn is defined as how many customers are not using the service for a certain period.

Hence, customer churn is one of the essential metrics that every business must evaluate to grow. The churn rate is calculated by dividing the number of lost customers by the last number of customers. Thus, a company churn rate must be as low as possible, ideally 0%.

But why is it so important to calculate the churn rate? Does it affect the business if you lose around 5% of customers? Yes, the answer is that it costs more to acquire a new customer than retain the existing customers. Retaining the current customers, any company can spend less on operating costs needed to reach new customers.

So, we will use advanced machine learning techniques to predict the potential churners who are about to leave a company’s service and take the necessary steps to prevent it.

This project aims to build a deep learning model that will help predict customers who are likely to churn in the next N months and facilitate in taking business actions for reducing the churn.

Data Description:

The available dataset is Telco-Customer-Churn

This dataset has 7043 rows and 21 columns present.

The 21 features of this dataset are as follows:

1. Churn – the target variable, if the customer is churned or not (Yes / No)

2. customerID – The unique identification of every customer

3. gender- If the customer is a male or a female (Female / Male)

4. SeniorCitizen – If the customer is a senior citizen or not (0 / 1)

5. Partner – If the customer has a partner or not (Yes/No)

6. Dependents – If the customer has any dependents (Yes / No)

7. Tenure – The time period(months) the customer has stayed with the company.

8. PhoneService – If the customer has a phone service or not (Yes/No)

9. MultipleLines – If the customer has multiple lines or not (Yes/No/No Phone service)

10. InternetService – If the customer has any internet service or not (DSL/ Fibre optics/ No)

11. OnlineSecurity – If the customer has any online security (Yes/No/No internet service)

12. OnlineBackup – If the customer has any online backup (Yes/No/No internet service)

13. DeviceProtection – If the customer has device protection (Yes/No/No internet service)

14. TechSupport – If the customer has tech support (Yes/ No/ No internet service)

15. StreamingTV – If the customer has any streaming TV (Yes/ No/ No internet service)

16. StreamingMovies – If the customer has streaming movies (Yes/ No/ No internet service)

17. Contract – The customer term period with the company (Month-to-month, One year, Two years)

18. PaperlessBilling – If the customer has paperless billing or not (Yes/ No)

19. PaymentMethod – The payment mode of each customer (Electronic check, mailed

check,Bank transfer, Credit card)

20. MonthlyCharges – The amount that is charged to the customer every month

21. TotalCharges – The total amount charged to the customer

Contents:

1. [Dataset Information](#Dataset_info)
2. [Exploratory Data Analysis (EDA)](#EDA)
3. [Feature Engineering](#FeatureEnginnering)
4. [Modeling](#Modeling)
5. [Conclusion](#Conclusion)
6. **Dataset Information:**
   * Importing the common libraries such as numpy, pandas, matplotlib, seaborn.
   * Importing and loading the Dataset.
   * Viewing the dataset.
   * Fetching the information about the data, i.e dtype, null values if any.
7. **Exploratory Data Analysis:**
   * Checking the number of rows and columns in the data
   * Getting to know the column/ feature names
   * Checking the number of unique labels in all the columns/features.
   * Creating a table that consists of feature name, dtypes, missing values, number of unique values
   * Getting the statistical insights form the data
   * Understanding the target variable by plotting a bar graph
   * Performing univariate analysis for all the columns individually wrt churn(target variable)
   * Creating a copy of the df and naming it new\_df
   * As most of the features are of dtype object, we need to encode them to numeric. Here we are doing label encoding for all the features with dtype object except for ‘TotalCharges’ and ‘customerID’
   * Viewing the new\_df after label encoding.
   * Now to understand the usecase better, we are dividing the features into groups such as:
8. Customer Information wrt Churn
9. Customer Information wrt customers who are more likely to churn(churn = yes/1)
10. Services Subscribed by customer wrt Churn
11. Services Subscribed by customer wrt customers who are more likely to churn(churn = yes/1)
12. Payment Information wrt churn
13. Payment Information wrt customers who are more likely to churn(churn = yes/1)
14. **Feature Enginnering(Data Preprocessing)**
    * TotalCharges must be numberic but it is given as object dtype, so we should convert it to numeric
    * Check for null values if any in TotalCharges.
    * Now that we have 11 null values found, we can drop those.
    * Checking for correlation of the features
    * Analyzing only the numerical variables(‘tenure’, ‘monthlycharges’, ‘totalcharges’) to understand the data
    * Visualizing the numerical variables(‘tenure’, ‘monthlycharges’, ‘totalcharges’) wrt churn
    * Checking for outliers
    * We see that there are no outliers.
    * Plotting to know the relationship between monthlycharges and totalcharges.
    * We see that as the total charges increase, the monthly charges also increase.
    * Normalizing the data, because the features are in different units, tenure is in months whereas monthly charges and total charges are in rupees.
    * After normalizing we split the variables into dependent and independent variables i.e into x and y.
15. **Modeling**
    * Lets first import few libraries like train\_test\_split, roc\_auc\_score, f1\_score, precison\_score, recall\_score., etc…
    * Data Balancing using SMOTE :
      1. In order to cope with imbalanced data, there are 2 options :
      2. Undersampling : Trim down the majority samples of the target variable.
      3. Oversampling : Increase the minority samples of the target variable to the majority samples.
      4. we have decided to go with oversampling beacuse we might lose data if we do undersampling.
      5. For data balancing, we will use imblearn.
      6. pip statement : !pip install imbalanced-learn
    * Importing imblearn, counter, smote
    * Splitting the data into train and test
    * Applying smote to handle the imbalanced data.

1. Logistic Regression

Observations:

* + Logistic Regression Accuracy : 0.797936893203883
  + Logistic Regression f1 score : 81.24%
  + Logistic Regression Precision score : 77.28%
  + Logistic Regression recall score : 85.63%

1. Decision Tree Classifier

Observations:

* + Decision Tree Classifier Accuracy : 0.77912621359223
  + Decision Tree Classifier f1 score : 79.37%
  + Decision Tree Classifier Precision score : 75.68%
  + Decision Tree Classifier recall score : 83.43%

1. Random Forest Classifier

Observations:

* + Random Forest Classifier Accuracy : 0.787621359223301
  + Random Forest Classifier f1 score : 79.37%
  + Random Forest Classifier Precision score : 73.81%
  + Random Forest Classifier recall score : 87.27%

1. K Nearest Neighbours:

Observations:

* + KNN Accuracy : 0.7961165048543689
  + KNN f1 score : 81.00%
  + KNN Precision score : 73.44%
  + KNN recall score : 90.29%

1. Support Vector Classifier

Observations:

* + Support Vector Classifier Accuracy : 0.7967233009708737
  + Support vector Classifier f1 score : 80.80%
  + Support vector Classifier Precision score : 77.73%
  + Support vector Classifier recall score : 84.13%

1. Ada Boost Classifier

Observations:

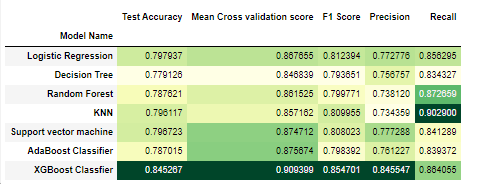
* + AdaBoostClassifier Accuracy : 0.7870145631067961
  + AdaBoostClassifier f1 score : 79.84%
  + AdaBoostClassifier Precision score : 76.12%
  + AdaBoostClassifier recall score : 83.94%

1. XGBoost Classifier

Observations:

* + XGBClassifier Accuracy : 0.8452669902912622
  + XGBClassifier f1 score : 85.47%
  + XGBClassifier Precision score : 84.55%
  + XGBClassifier recall score : 86.41%

1. Creating a table that consists of all the accuracy scores, mean cross validation score, f1 score, precision score, recall score.



We see that 'XGBoost classifier' gives 83.8% accuracy followed by KNN and Support vector machine which gives 80% accuracy.

**5.** **Conclusion**

\* CONCLUSION wrt Customer Information

1. Customer churning for SeniorCitizen customers is pretty low.

2. Customers without a partner or dependent are more likely to churn.

\* CONCLUSION wrt Services Subscribed by the Customer

1. For PhoneService, despite having no phone service, more customers were retained as compared to the number of customers who dropped the services.

2. Customers have mostly dropped services when there was no TECHSUPPORT, Device Protection, online backup and online security.

3. A high number of customers have displayed their resistance towards the use of Fiber optic cables for providing the InternetService. On the contrary, from the above graph, customers prefer using DSL for their InternetService!

\* CONCLUSION wrt Payment Information

1. Customers with contract of one year and more are less likelyy to churn.Customer churning for a Month-to-Month based Contract is quite high.

2. PaperlessBilling displays a high number of customers being churned out.

3. Customers clearly resented the Electronic check PaymentMethod. Company definitely needs to either drop Electronic check method or make it hassle-free and user-friendly.