PREDICTION OF CUSTOMER CHURN IN A TELECOMMUNICATIONS COMPANY

Table of Contents

Objective

The primary goal of this project is to develop a predictive model to identify customers at risk of churning in a telecommunications company. The project aims to enable the company to take proactive measures to retain customers.

1.Introduction

Customer churn is the percentage of customers that stopped using a company's product or service during a certain time frame. Customer churn is one of the most important metrics for a growing business to evaluate as it is much less expensive to retain existing customers than it is to acquire new customers. Customers in the telecom industry can choose from a variety of service providers and actively switch from one to the next. The telecommunications business has an annual churn rate of 15-25 percent in this highly competitive market.Customer attrition is one of the biggest expenditures of any organization. Customer churn otherwise known as customer attrition or customer turnover is the percentage of customers that stopped using your company's product or service within a specified timeframe. For instance, if you began the year with 500 customers but later ended with 480 customers, the percentage of customers that left would be 4%. If we could figure out why a customer leaves and when they leave with reasonable accuracy, it would immensely help the organization to strategize their retention initiatives manifold.

Customer churn is extremely costly for companies. Individualized customer retention is demanding because most companies have a large number of customers and cannot afford to devote much time to each of them. The costs would be too great, outweighing the additional revenue. However, if a corporation could forecast which customers are likely to leave ahead of time, it could concentrate customer retention efforts only on these "high risk" clients.

In this project, we aim to find the likelihood of a customer leaving the organization, the key indicators of churn as well as the retention strategies that can be implemented to avert this problem.

2. Methodology

2.1 DataSet Collection

The data set used in this project is Telco Customer Churn, involving 7043 records and 21 attributes. The entire data set consists Customers who left within the last month – the column is called Churn.Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies.Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges.Demographic info about customers – gender, age range, and if they have partners and dependents.

```
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
```

Figure 1 : Dataset Description

2.2 Data Modeling

The data consists initially of a large number of entries which also has repetitive data and some information that is not usable and suitable for our goal, which we will optimize by handling null values and outlier detection. Hence, the final selected dataset had a much smaller size compared to the due to the removal of non-usable and redundant data from the original dataset entries and irrelevant data.

2.3 Data Preprocessing

2.3.1 Handling Missing Values

The dataset has only one column that has missing values i.e Total Charges Column.I replaced the missing values with the mean.

2.3.2 Encoding Categorical Variables

This step is the key to achieve a high accuracy. I used **Target guided ordinal encoding**. Target-guided ordinal encoding is a powerful technique for encoding categorical variables, especially when building predictive models for classification tasks. Ordering the categories according to the target means assigning a number to the category, but this numbering, this ordering, is informed by the mean of the target within the category. Briefly, I calculated the mean of the target for each label/category, then I ordered the labels according to these mean from smallest to biggest, and I numbered them accordingly.

2.4 Exploratory Data Analysis

After data preprocessing, in order to clearly understand the nature of our data, an exploratory analysis was conducted. EDA is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations.

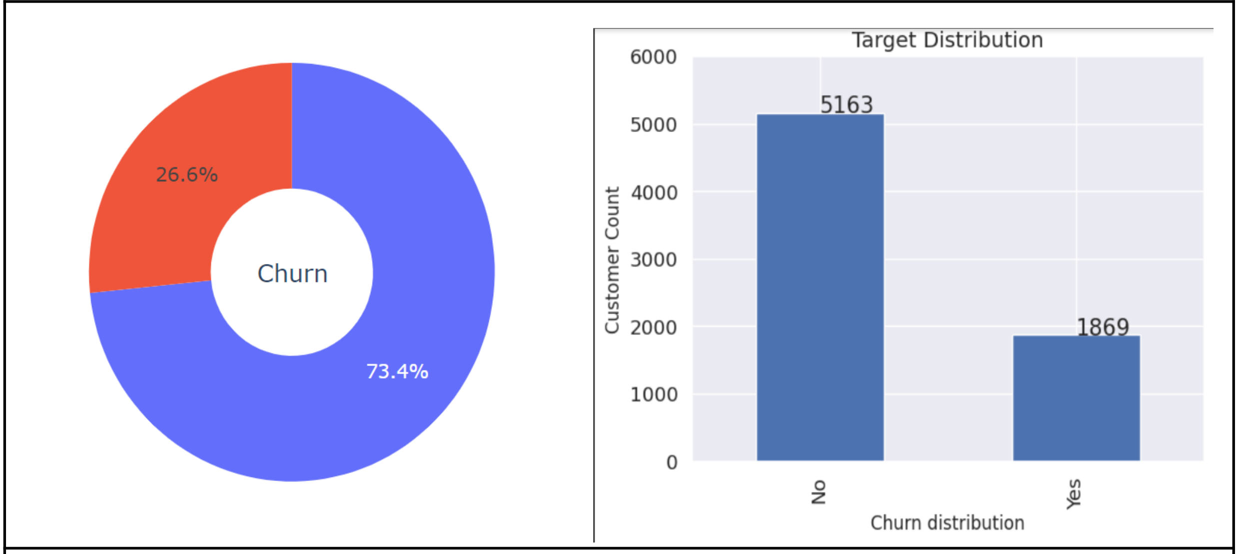

Figure 2 : Churn Distributions

Figure 2 depicts that the percentage of No is more i.e 73.4% and Yes is 26.6%.

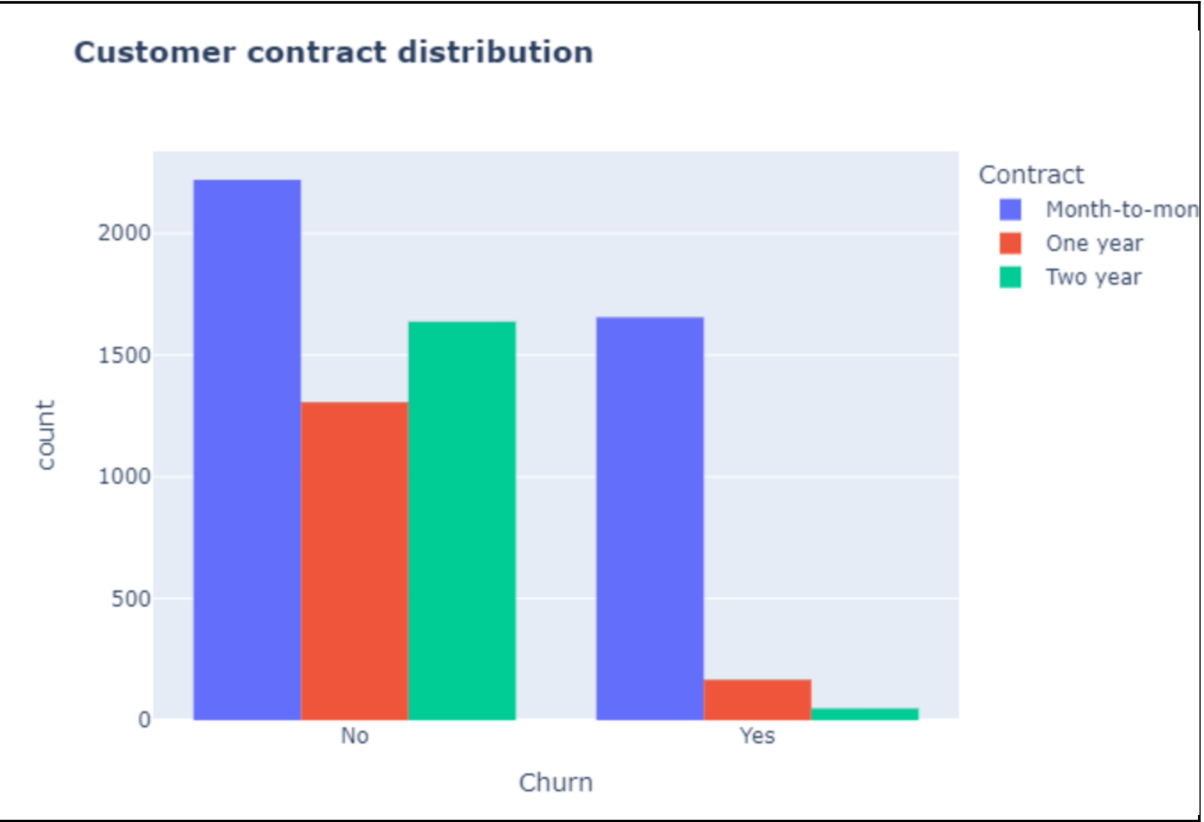

Figure 3 : Customer Contract Distribution

Figure 3: 75% of customers with Month-to-Month Contract opted to move out as compared to 13% of customers with One Year Contract and 3% with Two Year Contract.

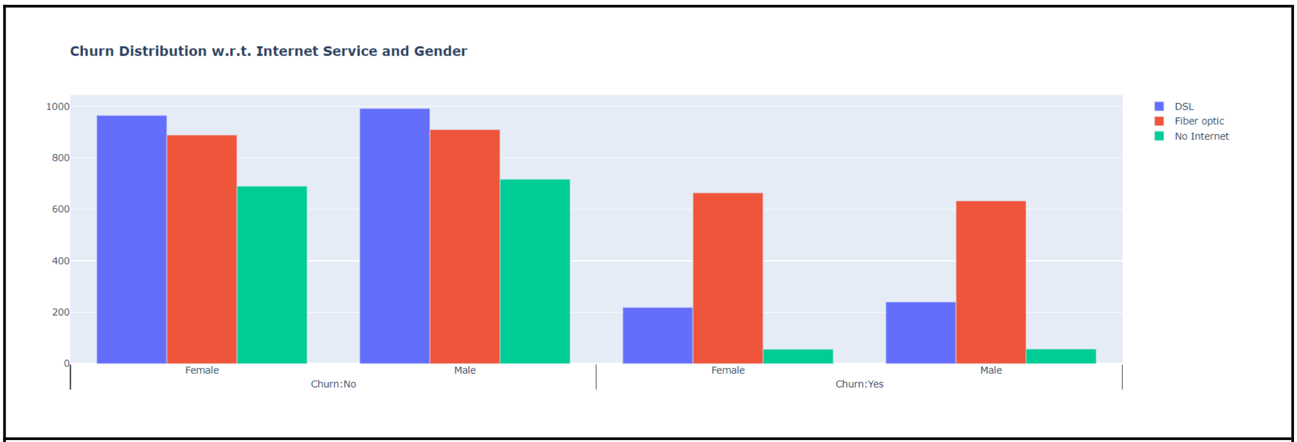

Figure 4 : Churn Distribution between Internet Service and Gender.

Figure 4 depicts that Several customers choose the Fiber optic service and it's also evident that the customers who use Fiber optic have high churn rate, this might suggest a dissatisfaction with this type of internet service. Customers having DSL service are majority in number and have less churn rate compared to Fiber optic service.

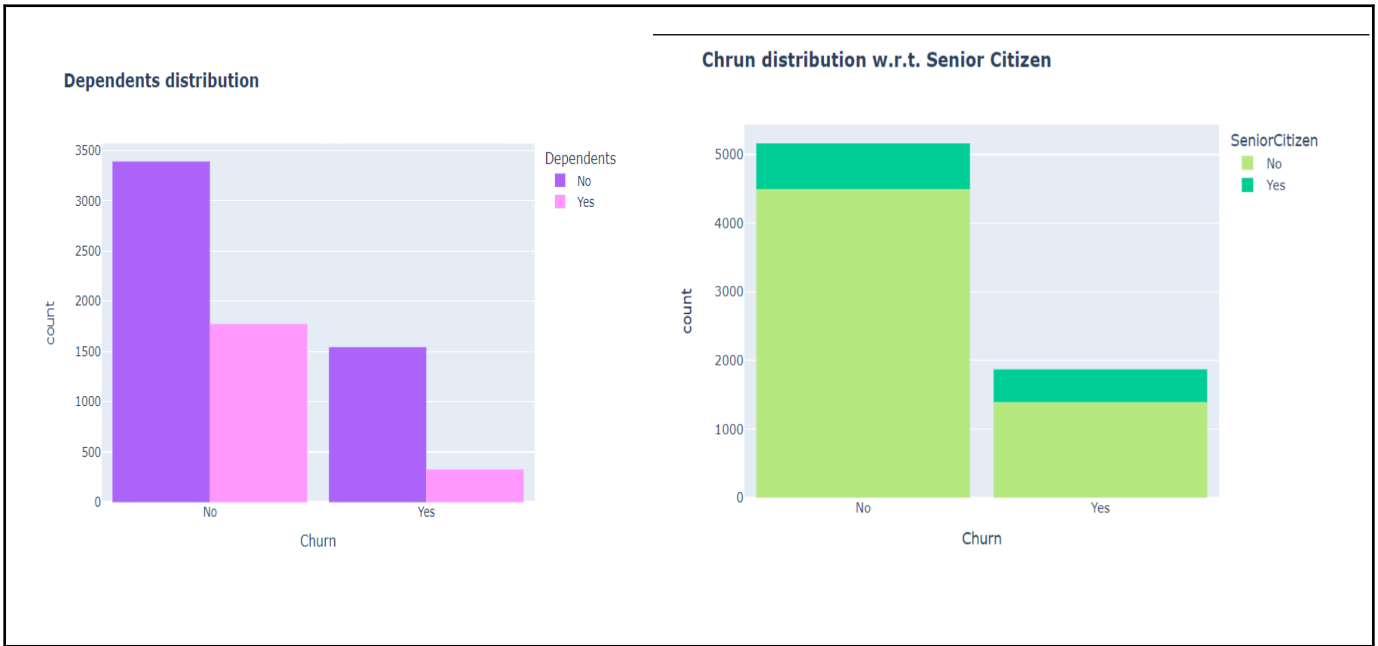

Figure 5 : Dependents Distribution and Senior Citizen.

Figure 5 depicts that Customers without dependents are more likely to churn.

Most of the senior citizens churn, the number of senior citizens are very less in the all customer base.

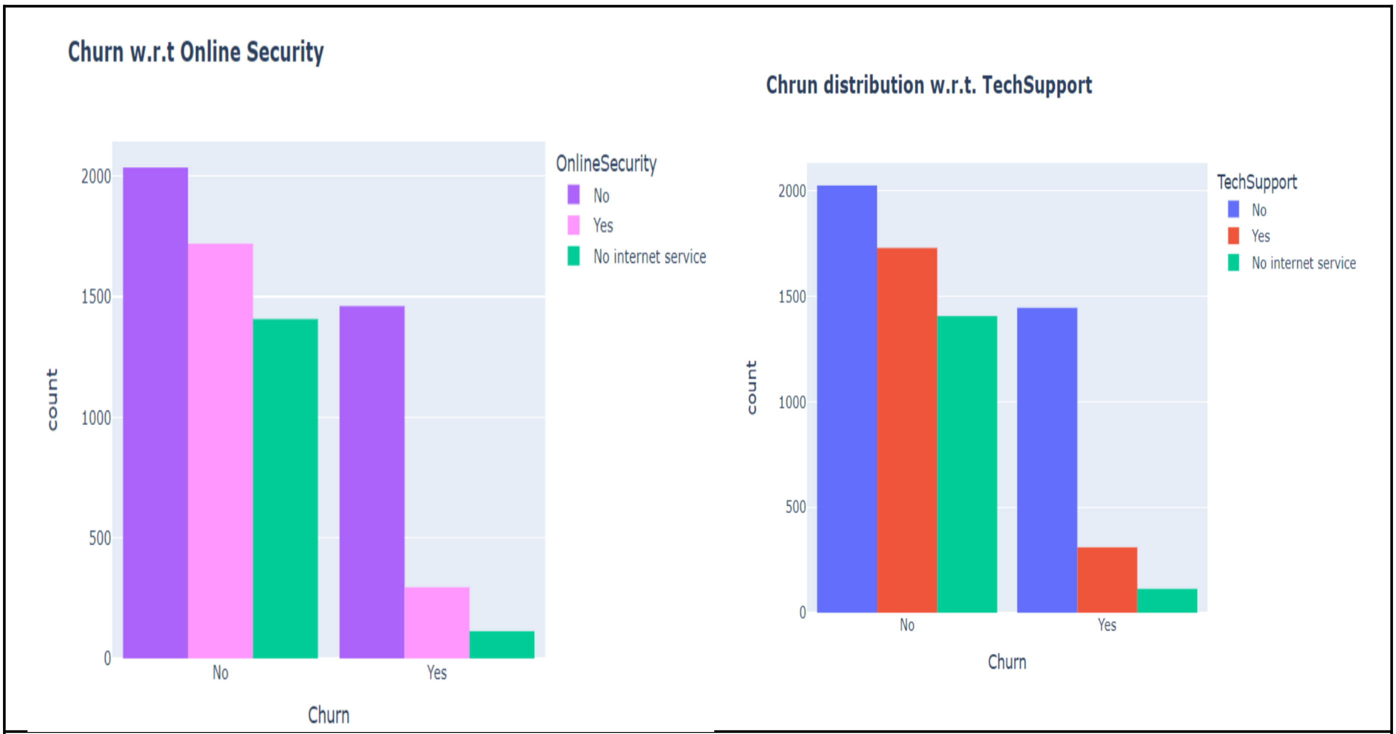

Figure 6 : Online Security and Tech Support.

Figure 6 depicts most customers churn due to lack of online security customers with no TechSupport are most likely to migrate to another service provider.

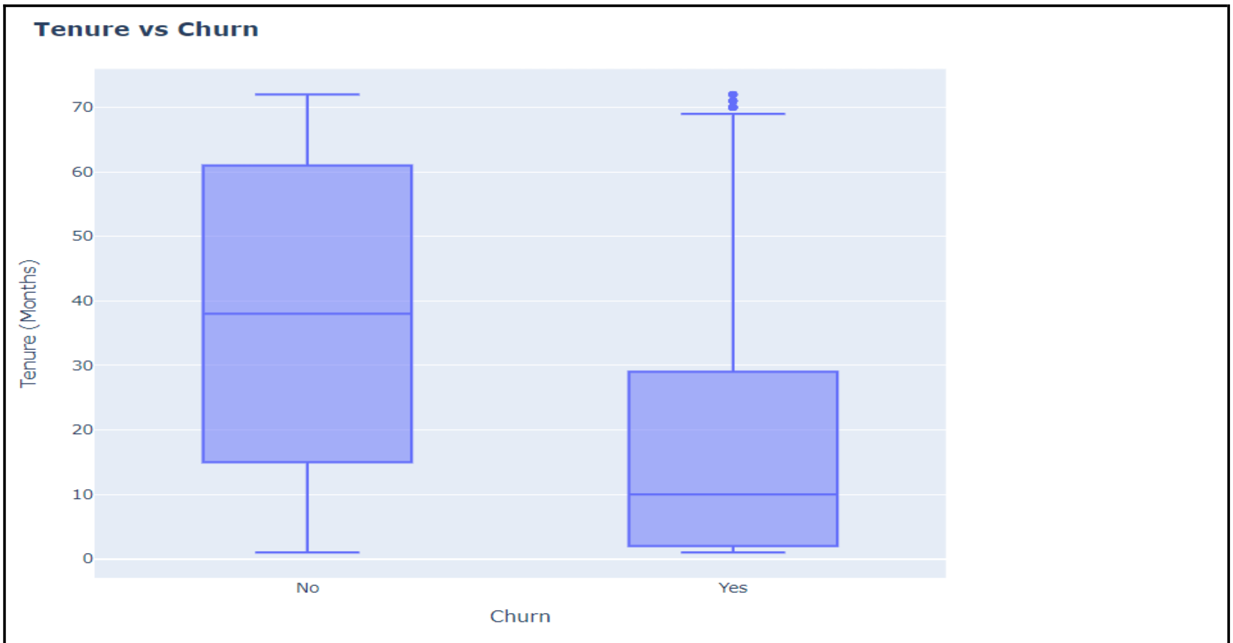

Figure 7 : Tenure Vs Churn.

Figure 7 depicts Customers with higher Monthly Charges are also more likely to churn.New customers are more likely to churn.

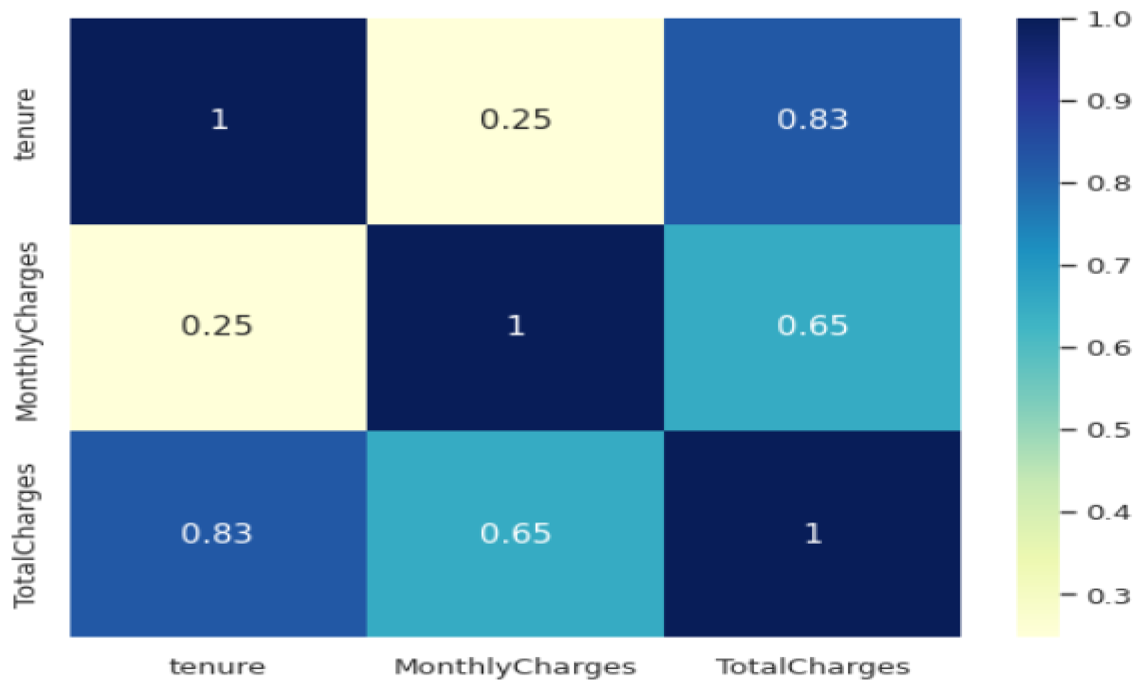

Figure 8 : Correlation Matrix

Figure 8 : Total charges are correlated with Tenure and Monthly charges. Customers who stay for a longer tenure will have less chances of churning.

2.5 Outlier Detection

In the data preprocessing stage, outlier detection is an essential step that aids in locating and managing data points that differ considerably from the bulk of the dataset.Outliers can adversely impact the performance of machine learning models, and addressing them appropriately is essential for accurate predictions.The presence of outliers in a classification or regression dataset can result in a poor fit and lower predictive modeling performance, therefore we should see there are outliers in the data.

```
detect_outliers(['tenure', 'MonthlyCharges', 'TotalCharges']) #No outlier

*** tenure outlier points***
 Series([], Name: tenure, dtype: int64)

*** MonthlyCharges outlier points***
 Series([], Name: MonthlyCharges, dtype: float64)

*** TotalCharges outlier points***
 Series([], Name: TotalCharges, dtype: float64)
```

Figure 9 : Outlier Detection.

3 . Machine Learning Techniques

3.1.1 Logistic Regression

Logistic Regression is a binary classification algorithm used in machine learning. It employs the sigmoid function to map a linear combination of input features and weights plus a bias term to a probability between 0 and 1. The model is trained by maximizing the likelihood of observed labels using the log loss cost function. Optimization techniques like Gradient Descent adjust weights and bias iteratively to minimize the cost. The resulting decision boundary separates instances of one class from another in the feature space. Logistic Regression is valued for its simplicity, interpretability, and efficiency. Despite being linear, it performs well in scenarios with approximately linear decision boundaries. For more complex relationships, advanced models like support vector machines or deep neural networks may be preferred.

3.1.2 Random Forest Regression

Random forest is known as a supervised model learning strategy for regression, classification and other relevant works that is made up by arranging countless decision trees. It is considered as one of the flexible techniques which can produce a result most of the time even without any hyper parameter optimization. It is simply diverted. The forest ensembles decision trees, often trained by a bagging method which combines learning models to increase overall result.

3.1.3 KNN

K-Nearest Neighbors (KNN) is an instance-based machine learning algorithm for classification and regression. It makes predictions based on the majority class (for classification) or average target values (for regression) of the K nearest neighbors in the feature space. The algorithm doesn't have a training phase, memorizing the entire dataset. The choice of distance metric and the value of K are critical hyperparameters. KNN is suitable for scenarios with locally homogeneous data but may struggle with high-dimensional or noisy datasets. It is a "lazy learner" as it defers computation to prediction time. KNN finds applications in recommendation systems and pattern recognition, especially when decision boundaries are complex.

3.1.4 Decision Tree

Decision tree is a type of supervised machine learning algorithm used for both classification and regression tasks. It is a type of tree-like graphical representation of different choices and their possible consequences. It is used for predicting outcomes for a given data set by learning decision rules from the data features. It works by breaking down the data into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes representing the classification or decision. The topmost node in the tree is known as the root node. Each node in the tree is a "test" on an attribute, and each branch represents the outcome of the test.

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

3.1.5 AdaBoost

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm used for binary classification. It combines multiple weak learners, like simple decision stumps, in a sequential manner. Initially, all instances have equal weight, and misclassified instances gain higher weights in subsequent iterations. The final model is a weighted sum of the weak learners, with individual contributions based on accuracy. AdaBoost adapts by focusing on challenging instances, leading to improved performance and robustness. It is less prone to overfitting, handles outliers well, and is widely applied in tasks such as face detection and object recognition. The number of iterations is a crucial hyperparameter. In summary, AdaBoost creates a strong classifier by emphasizing difficult instances and combining the strengths of weak learners through weighted voting.

3.1.6 XGBoost

XGBoost is a distributed gradient boosting machine learning technique that has been optimized. A tree model designed to be highly efficient and flexible is good at classification and regression problems. The experimental results demonstrate that the proposed model is based on XGBoost's good performance in improving the running rate and the prediction accuracy. And we succeed at making better sales predictions compared to other machine learning methods. This technique ensembles decision trees that use the framework of gradient boosting. Algorithms based on decision trees are thought to be the most optimal for small to medium amounts of structured or tabular data. It is considered high in efficiency, flexibility, and portability.

3.1.7 Stacking

Stacking is one of the popular ensemble modeling techniques in machine learning. Various weak learners are ensembled in a parallel manner in such a way that by combining them with Meta learners, we can predict better predictions for the future.This ensemble technique works by applying input of combined multiple weak learners' predictions and Meta learners so that a better output prediction model can be achieved.In stacking, an algorithm takes the outputs of sub-models as input and attempts to learn how to best combine the input predictions to make a better output prediction.Stacking is also known as a stacked generalization and is an extended form of the Model Averaging Ensemble technique in which all sub-models equally participate as per their performance weights and build a new model with better predictions. This new model is stacked up on top of the others; this is the reason why it is named stacking.

4. Model Evaluation Metric

The metrics I used, Accuracy, F1 Score, and AUC Score,are most commonly used in machine learning to evaluate the performance of classification models.

Accuracy: Accuracy is the ratio of correctly predicted instances to the total instances. It is a commonly used metric for classification problems, but it may not be suitable for imbalanced datasets.

Formula for Accuracy:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

F1 Score:

The F1 Score is the harmonic mean of precision and recall.
Precision refers to the number of true positives divided by the total number of positive predictions

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision Formula

Recall, also known as the true positive rate (TPR), is the percentage of data samples that a machine learning model correctly identifies as belonging to a class of interest—the "positive class"—out of the total samples for that class.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall Formula

Formula for F1 Score:

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

AUC Score (Area Under the ROC Curve): The AUC Score represents the area under the Receiver Operating Characteristic (ROC) curve. It is particularly useful for binary classification problems and provides a measure of the classifier's ability to discriminate between positive and negative instances.The ROC curve is a plot of the true positive rate (sensitivity) against the false positive rate (1-specificity) for different threshold values.AUC values range from 0 to 1, where a higher value indicates better performance.These metrics are often used together to get a comprehensive view of a model's performance. It's important to note that the choice of evaluation metric depends on the specific characteristics of your dataset and the goals of your machine learning task.

5.Feature Importance

Random Forest is a popular ensemble learning method used for both classification and regression tasks. One of the advantages of Random Forest is its ability to provide an estimate of feature importance, which can be valuable for understanding the contribution of different features to the model's predictions.Random Forest algorithms offer importance scores based on the reduction in the criterion used to select split points, like Gini or entropy.

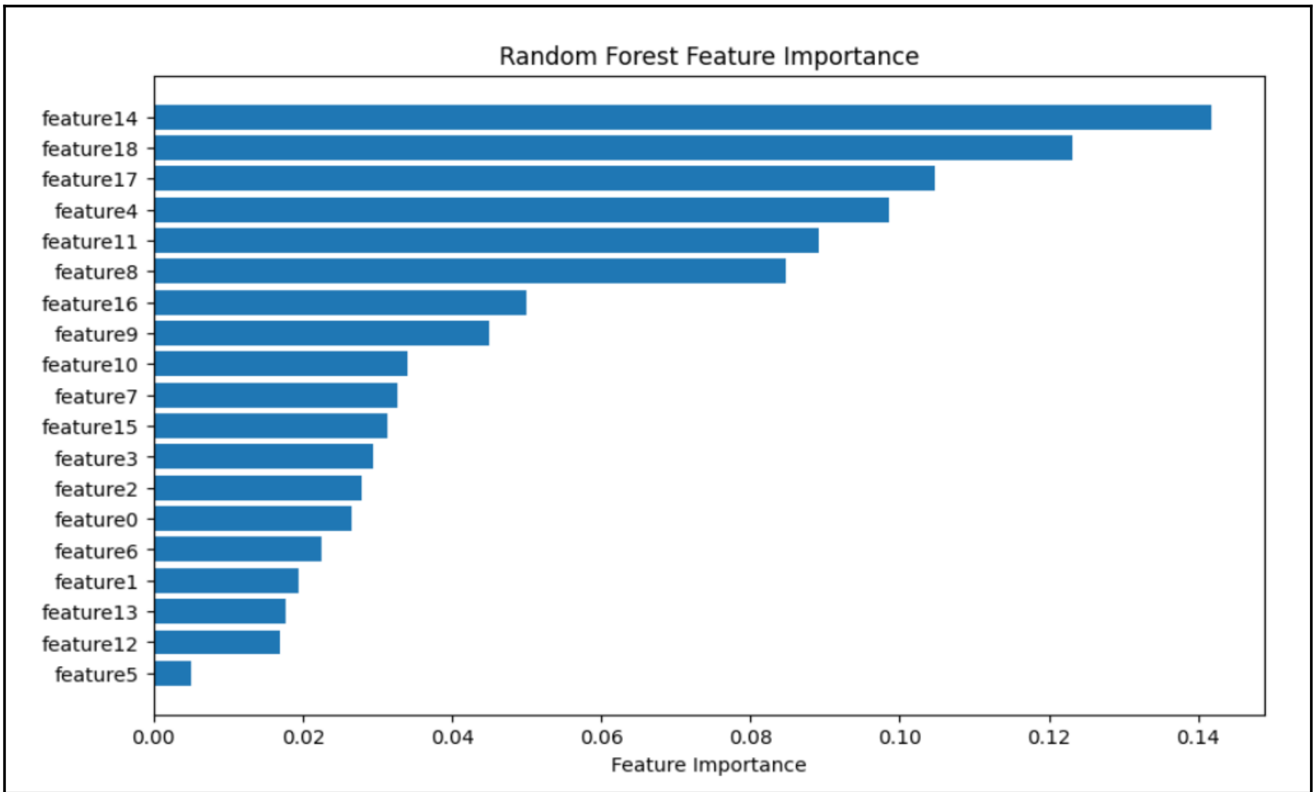

Figure 10 : Feature Importance using the Random Forest

6.Results and Discussion

	Model	Accuracy Mean	F1 Score Mean	AUC Score Mean
0	Logistic Regression	0.750561	0.769941	0.820525
1	Random Forest	0.857621	0.853320	0.934954
2	KNN	0.785674	0.798032	0.785845
3	Decision Tree	0.807586	0.806255	0.808546
4	Ada Boost	0.854506	0.855512	0.940120
5	XG Boost	0.832922	0.829593	0.921555
6	Stacking	0.865780	0.863096	0.946002

Figure 11 : Comparison of Various Models on Training Data Set.

Stacking model is the most stable and accurate model. As a result, Stacking is selected for the purpose of predicting Churn on the test data set.

```
48s [106] Stack.fit(X_train, y_train)
          y_pred = Stack.predict(X_test)

0s        print(classification_report(y_test,y_pred))

                    precision    recall  f1-score   support

                 0       0.87      0.90      0.89       505
                 1       0.90      0.87      0.89       530

          accuracy                           0.89      1035
         macro avg       0.89      0.89      0.89      1035
      weighted avg       0.89      0.89      0.89      1035

0s  [107] y_prob = Stack.predict_proba(X_test)
          roc_auc_score(y_test, y_prob[:,1],average='macro')

          0.9481337567719035
```

Figure 12 : Auc Score on Test Data set.

I achieved a 94.8% roc score and 85% accuracy on the test dataset.

7.Conclusion

Customer churn prediction is a crucial tool for businesses to proactively manage customer retention strategies. Utilizing data-driven models to forecast potential churn allows businesses to gain insights into customer behavior, anticipate trends, and tailor strategies to retain valuable customers. However, it is important to remember that no prediction model is perfect and that businesses should use a combination of methods to make the best decisions possible.

In this project, I successfully developed a predictive model for customer churn in a telecommunications company. I began by collecting and preprocessing data, addressing missing values and encoding categorical variables. Exploratory Data Analysis (EDA) provided insights into customer behavior.

The churn prediction model was built using machine learning algorithms such as Logistic regression, Random forests, KNN, Decision Tree, Ada Boost, XGBoost, Stacking.
Among all these models I got more accuracy on Stacking while training the dataset. As a result, Stacking is used for Predicting the Churn on the test data,I got a 94.8% roc score and 85% accuracy. We fine-tuned the models and evaluated their performance using key metrics like accuracy, precision, recall,F1-score and Auc Score..