CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING (C-DAC)

CHENNAI, TAMILNADU

PROJECT REPORT ON

"Telecom Customer Churn"

SUBMITTED TOWARDS THE

PG-DBDA September -2023

By Group

AKSHAY KULKARNI **PRN : 230960825003**

PARTH REDDY **PRN : 230960825005**

Under The Guidance Of

Mr. Harikrishnan Sir.

TABLE OF CONTENTS

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Mr. Harikrishnan sir**, C-DAC ACTS, Chennai for her constant guidance and helpful suggestion for preparing this project **TELECOM CUSTOMER CHURN PREDICTION**. We express our deep gratitude towards her for his inspiration, personal involvement, and constructive criticism that she provided us along with technical guidance during the course of this project.

We take this opportunity to thank the Head of the department **Dr. Sumithra** for providing us with such a great infrastructure and environment for our overall development.

Also our warm thanks to C-DAC ACTS, Chennai which provides us with this opportunity to carry out this prestigious Project and to enhance our learning in various technical fields.

Akshay Mukund Kulkarni (230960825003)

Parth Kashiling Reddy (230960825005)

Chapter 1: Introduction

1.1 About the Project

Customer churn is defined as when customers or subscribers discontinue doing business with a firm or service.

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.

Individualized customer retention is tough because most firms have a large number of customers and can't 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 focus customer retention efforts only on these "high risk" clients. The ultimate goal is to expand its coverage area and retrieve more customers loyalty. The core to succeed in this market lies in the customer itself.

Customer churn is a critical metric because it is much less expensive to retain existing customers than it is to acquire new customers

To reduce customer churn, telecom companies need to predict which customers are at high risk of churn.

To detect early signs of potential churn, one must first develop a holistic view of the customers and their interactions across numerous channels, including store/branch visits, product purchase histories, customer service calls, Web-based transactions, and social media interactions, to mention a few.

As a result, by addressing churn, these businesses may not only preserve their market position, but also grow and thrive. More customers they have in their network, the lower the cost of initiation and the larger the profit. As a result, the company's key focus for success is reducing client attrition and implementing effective retention strategy.

1.2 Project Objective

I will explore the data and try to answer some questions like:

- What's the % of Churn Customers and customers that keep in with the active services?
- Is there any patterns in Churn Customers based on the gender?
- Is there any patterns/preference in Churn Customers based on the type of service provided?
- What's the most profitable service types?
- Which features and services are most profitable?

1.3 Problem Statement:

In the telecom industry, customers are able to choose from a pool of companies to cater their needs regarding communication and internet.

Customers are very critical about the kind of services they receive and judge the enitre company based on a single experience! These communication services have become so recurrent and inseparable from the daily routine that a 30 minute maintenance break kicks in anxiety in the users highlighting our taken-for-granted attitude towards these services! Coupled with high customer acquisition costs, churn analysis becomes very pivotal! Churn rate is a metric that describes the number of customers that cancelled or did not renew their subscription with the company.

Thus, higher the churn rate, more customers stop buying from your business, directly affecting the revenue! Hence, based on the insights gained from the churn analysis, companies can build strategies, target segments, improve the quality of the services being provided to improve the customer experience, thus cultivating trust with the customers.

That is why building predictive models and creating reports of churn analysis becomes key that paves the way for growth!

1.4. About Dataset

There are 17 categorical features:

- **Customer:** Customer ID unique for each customer
- **gender:** Whether the customer is a male or a female
- **SeniorCitizen:** Whether the customer is a senior citizen or not (1, 0)
- **Partner:** Whether the customer has a partner or not (Yes, No)
- **Dependent:** Whether the customer has dependents or not (Yes, No)
- **PhoneService:** Whether the customer has a phone service or not (Yes, No)
- **MultipeLines:** Whether the customer has multiple lines or not (Yes, No, No phone service)
- **InternetService:** Customer's internet service provider (DSL, Fiber optic, No)
- **OnlineSecurity:** Whether the customer has online security or not (Yes, No, No internet service)
- **OnlineBackup:** Whether the customer has an online backup or not (Yes, No, No internet service)
- **DeviceProtection:** Whether the customer has device protection or not (Yes, No, No internet service)
- **TechSupport:** Whether the customer has tech support or not (Yes, No, No internet service)
- **StreamingTV:** Whether the customer has streaming TV or not (Yes, No, No internet service)
- **StreamingMovies:** Whether the customer has streaming movies or not (Yes, No, No internet service)

- **Contract:** The contract term of the customer (Month-to-month, One year, Two years)
- **PaperlessBilling:** The contract term of the customer (Month-to-month, One year, Two years)
- **PaymentMethod:** The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
  Next, there are 3 numerical features:
- **Tenure:** Number of months the customer has stayed with the company
- **MonthlyCharges:** The amount charged to the customer monthly
- **TotalCharges:** The total amount charged to the customer

- **Prediction feature:**

  Churn: Whether the customer churned or not (Yes or No)

These features can also be subdivided into:

- **Demographic customer information:**

  gender , SeniorCitizen , Partner , Dependents

- **Services that each customer has signed up for:**

  PhoneService , MultipleLines , InternetService , OnlineSecurity , OnlineBackup , DeviceProtection , TechSupport , StreamingTV , StreamingMovies,

- **Customer account information:**

  tenure , Contract , PaperlessBilling , PaymentMethod , MonthlyCharges , TotalCharges

Chapter 2: Literature Survey

Kavita et al. suggested that customers who had been purchased the expensive services and were senior citizen are more likely to churn in telecom industry. Their result describes that the future buying strongly depends on the history of previously bought services. They used a lot of machine learning algorithms and analyzed them and concluded that logistic regression performs well for customer churn prediction

Sayedur Rahman et al. focused of using concept on impact learning in which data is trained by considering impacts of features. This research also included a comparison of Impact Learning, Logistic Regression, and Artificial Neural Networks. Maximum accuracy of 81.06% is achieved by using impact learning.

With the help of ensemble techniques like Random Forest, Extreme Gradient Boosting (XGBoost), and Adaptive Boosting (AdaBoost) and different supervised machine learning algorithms like Decision Tree, Support Vector Machine (SVM), Logistic Regression, and Artificial Neural Networks (ANN), Prasanth Senthan at el. was able to compare and attain an accuracy of 82.01%.

Kassem et al. has given two approaches for the churn prediction problem. Identifying factors affecting churn. Identifying customers who will churn for sure. They began gathering data through practical inquiries in order to identify the influencing aspects, and then they used machine learning methods like logistic regression and naive bayes to assess the results. For second approach they had performed sentiment analysis to get polarity of the User Generated Contents (UGC)

Damandeep Singh et al. compared different algorithms which are used to predict customer churn and found out Gradient boost model can provide the best accuracy of 78.05% while Random Forest Classifier can give an accuracy of 77.38%. He compared 9 different algorithms to formulate the importance of feature selection

Latifah Almuqren et al. used a different approach to Social Media Mining in this proposed method we use real-time data to analyse customers' sentiments with real-time data. Social media being the number one way to communicate and reach out to people this method provides a new approach and can convey the ground reality of the service that is being provided. Telecom companies can keep their social image protected and can also satisfy their customers' needs in this way.

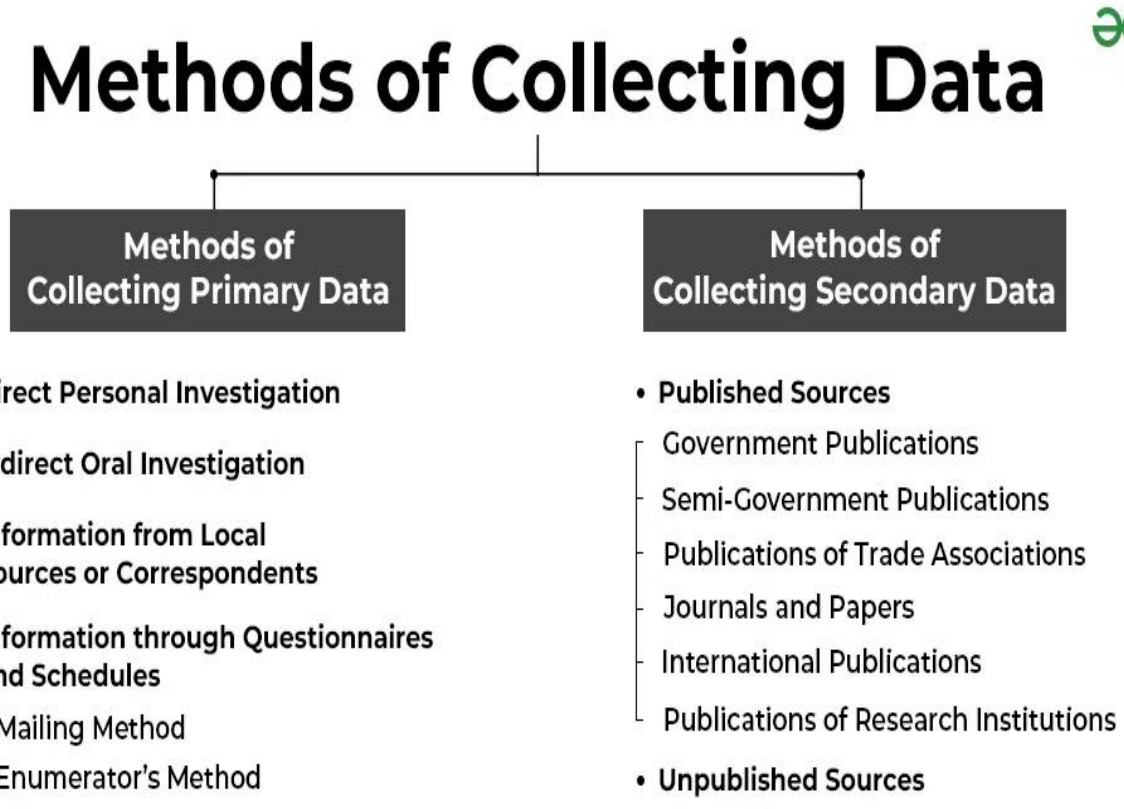
Chapter 3: Feature Engineering

3.1 Data Collection

Data collection is the process of collecting and evaluating information or data from multiple sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities. It is an essential phase in all types of research, analysis, and decision-making, including that done in the social sciences, business, and healthcare.

Accurate data collection is necessary to make informed business decisions, ensure quality assurance, and keep research integrity.

3.1.1. Methods of Collecting Data



For our project, we have collected the dataset from Kaggle's Official Website.

Link:- https://www.kaggle.com/datasets/blastchar/telco-customer-churn/data

3.2 Data Preprocessing

Data preprocessing is an important step in the data mining process. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

3.2.1 Data Cleaning:

This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates. Various techniques can be used for data cleaning, such as imputation, removal, and transformation.

```
data['TotalCharges'] = pd.to_numeric(data.TotalCharges, errors='coerce')
data.isnull().sum()
```

```
[17]: customerID          0
      gender              0
      SeniorCitizen       0
      Partner             0
      Dependents          0
      tenure              0
      PhoneService        0
      MultipleLines       0
      InternetService     0
      OnlineSecurity      0
      OnlineBackup        0
      DeviceProtection    0
      TechSupport         0
      StreamingTV         0
      StreamingMovies     0
      Contract            0
      PaperlessBilling    0
      PaymentMethod       0
      MonthlyCharges      0
      TotalCharges       11
      Churn               0
      dtype: int64
```

In our Dataset We find the there are 11 Null Values are there in Total Charges Column.

To solve the problem of missing values in TotalCharges column, we decided to fill it with the **Mean** of TotalCharges values.

```
[22]: new_data=data.fillna(data["TotalCharges"].mean())

[23]: new_data.isnull().sum()

[23]: customerID          0
      gender              0
      SeniorCitizen       0
      Partner             0
      Dependents          0
      tenure              0
      PhoneService        0
      MultipleLines       0
      InternetService     0
      OnlineSecurity      0
      OnlineBackup        0
      DeviceProtection    0
      TechSupport         0
      StreamingTV         0
      StreamingMovies     0
      Contract            0
      PaperlessBilling    0
      PaymentMethod       0
      MonthlyCharges      0
      TotalCharges        0
      Churn               0
      dtype: int64
```

3.3 Feature Scaling

Feature scaling is a data preprocessing technique used to transform the values of features or variables in a dataset to a similar scale. The purpose is to ensure that all features contribute equally to the model and to avoid the domination of features with larger values.

Feature scaling becomes necessary when dealing with datasets containing features that have different ranges, units of measurement, or orders of magnitude. In such cases, the variation in feature values can lead to biased model performance or difficulties during the learning process. There are several common techniques for feature scaling, including standardization, normalization, and min-max scaling.

By applying feature scaling, the dataset's features can be transformed to a more consistent scale, making it easier to build accurate and effective machine learning models.

- **What is Standardization?**

Standardization is another Feature scaling method where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero, and the resultant distribution has a unit standard deviation.

3.4 Feature Extraction

3.3.1. Label Encoding

When working with categorical data in machine learning, it is essential to convert these variables into a numerical format that algorithms can understand. Two commonly used techniques for encoding categorical variables are one-hot and label encoding. Choosing the appropriate encoding method can significantly impact the performance of a ML model. In this article, we will explore the differences between one-hot encoding vs label encoding, their use cases, and how to implement them using the powerful Scikit-Learn library in Python.

- **What is Categorical Encoding?**

Typically, any structured dataset includes multiple columns – a combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text.

Categorical encoding is the process of converting categorical columns to numerical columns so that a machine learning algorithm understands it. It is a process of converting categories to numbers.

- **Different Approaches to Categorical Encoding**

So, how should we handle categorical variables? As it turns out, there are multiple ways of handling Categorical variables. In this article, I will discuss the two most widely used techniques:

- Label Encoding
- One-Hot Encoding

- **What is Label Encoding?**

Label Encoding is a popular encoding technique for handling categorical variables. A unique integer or alphabetical ordering represents each label.

Input :-

Importing Label Encoder

```
from sklearn.preprocessing import LabelEncoder
```

Here we will be encoding the categorical data of object data type to numeric as 0, 1, 2.

```
def object_to_int(new_data1):
    if new_data1.dtype=='object':
        new_data1 = LabelEncoder().fit_transform(new_data1)
    return new_data1
```

```
df = new_data1.apply(lambda x: object_to_int(x))
df.head()
```

Output :-

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport
0	0	0	1	0	1	0	1	0	0	2	0	0
1	1	0	0	0	34	1	0	0	2	0	2	0
2	1	0	0	0	2	1	0	0	2	2	0	0
3	1	0	0	0	45	0	1	0	2	0	2	2
4	0	0	0	0	2	1	0	1	0	0	0	0

Here, we can see that we encoded the categorical data of object data type to numeric as 0, 1, 2 by using Label Encoding Technique.

3.3.2. Handling Imbalanced Dataset

Resampling data is one of the most commonly preferred approaches to deal with an imbalanced dataset. There are broadly three types of methods for this

i) Under sampling
ii) Oversampling.
iii) SMOTE (Synthetic Minority Over-sampling Technique)

In most cases, oversampling is preferred over under sampling techniques. The reason being, in under sampling we tend to remove instances from data that may be carrying some important information. *In this project, we specifically use Synthetic Minority Over-sampling Technique (SMOTE).*

- **Synthetic Minority Oversampling Technique (SMOTE):**

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together.

SMOTE is specifically designed to tackle imbalanced datasets by generating synthetic samples for the minority class. By mitigating bias and capturing important features of the minority class, SMOTE contributes to more accurate predictions and better model performance.

- **Before handling the distribution of classes:**

```
[41]: y.value_counts()
```

```
[41]: Churn
      0    5163
      1    1869
      Name: count, dtype: int64
```

- **After handling**

```
from imblearn.over_sampling import SMOTE
```

```
smote = SMOTE(random_state=0)
X_resample, y_resample = smote.fit_resample(X,y)
```

```
y_resample.value_counts()
```

```
Churn
0    5163
1    5163
Name: count, dtype: int64
```

Chapter 4: Development and Coding

4.1 Technology Used

We have used multiple technologies for multiple Modules.

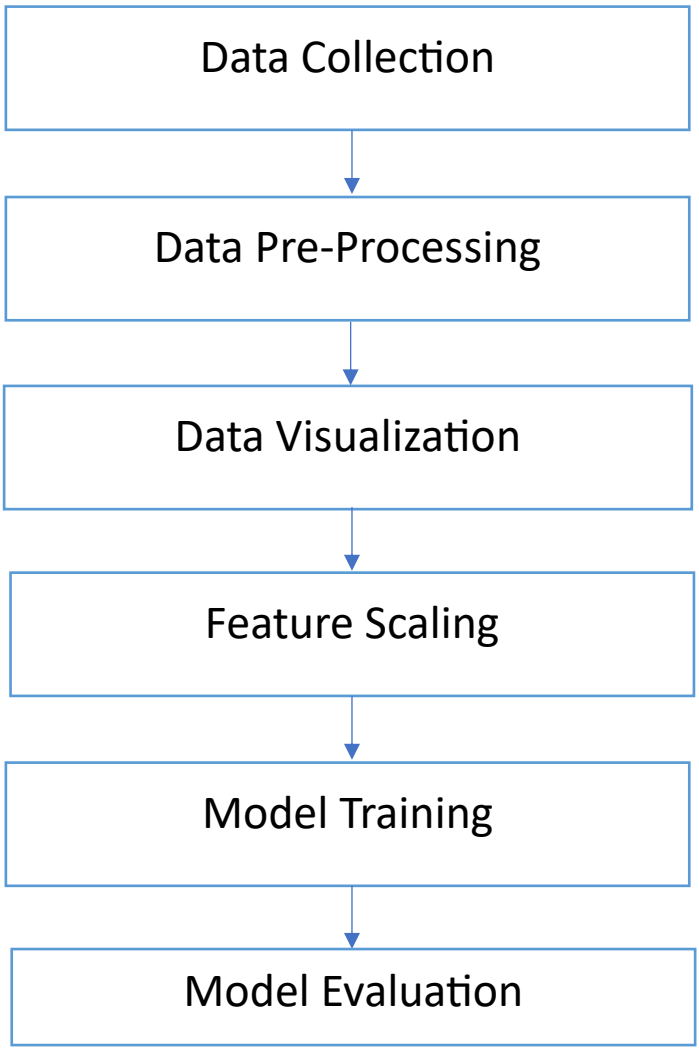
Programming Language	Python
Python Libraries	NumPy, Pandas, missingno, warning
Data Visualization	Matplotlib, Seaborn, Scipy, Tableau
Machine Learning Algorithms	1. Logistic Regression 2. Decision Tree 3. Random Forest 4. Support Vector Machine (SVM) 5. XG-Boost 6. Artificial Neural Network
Imbalanced Dataset Handling Technique	SMOTE (Synthetic Minority Over-sampling Technique)
Hyperparameter Turning	GridSearchCV
Machine Learning Library	Scikit-learn

4.2. Machine Learning:

Machine Learning is a branch of artificial intelligence that develops algorithms by learning the hidden patterns of the datasets used it to make predictions on new similar type data, without being explicitly programmed for each task.

- **Types of Machine Learning Algorithm :**
    - Linear regression
    - Logistic regression
    - Decision tree
    - SVM algorithm
    - Naive Bayes algorithm
    - KNN algorithm
    - K-means
    - Random forest algorithm
    - Dimensionality reduction algorithms
    - Gradient boosting algorithm and AdaBoosting algorithm

4.3. Architecture Diagram



- **Data Collection: -** We have collected the dataset from Kaggle's Official Website**.**

- **Data Pre-processing: -**

    1. Firstly , we have checked is there any Null Value is present or not.
    2. We have checked Duplicate values.
    3. We have checked Data Types of all Attributes.
    4. Then we checked Statistical Summary of our dataset.

- **Data Visualization: -** After Data pre-processing, we Visualized the dataset by using Matplotlib and Seaborn Libraries

- **Feature Scaling: -**

    - In Feature Scaling we have use Label Encoder for encoding the categorical data into numerical Data**.**

    - For Handling Imbalanced Data, We have used the Synthetic Minority Over-Sampling Technique (SMOTE).

    - For standardization, we have used Standard Scalar technique.

- **Model Training: -**

    - For Model training we have used **Six** Algorithms as follow: -

i. Logistic Regression.

```
lr_model = LogisticRegression()
lr_model.fit(X_train,y_train)
lr_pred= lr_model.predict(X_test)
accuracy_lr = lr_model.score(X_test,y_test)

print(classification_report(y_test, lr_pred))
print("-----------------------------------------------------")
print("Logistic Regression accuracy is :",accuracy_lr)
print("-----------------------------------------------------")
print('Confusion matrix for Logistic Regression :','\n',confusion_matrix(y_test,lr_pred))
```

ii. Decision Tree Classifier.

```
dt_model = DecisionTreeClassifier()
dt_model.fit(X_train,y_train)
predictdt_y = dt_model.predict(X_test)
accuracy_dt = dt_model.score(X_test,y_test)

print(classification_report(y_test, predictdt_y))
print("-----------------------------------------------------")
print("Decision Tree accuracy is :",accuracy_dt)
print("-----------------------------------------------------")
print('Confusion matrix for Decision Tree :','\n',confusion_matrix(y_test,predictdt_y))
```

iii. Random Forest.

```
rf_model = RandomForestClassifier(n_estimators=500, random_state=50, max_features=None, max_leaf_nodes=30)

rf_model.fit(X_train, y_train)
prediction_test = rf_model.predict(X_test)
rf_accuracy=metrics.accuracy_score(y_test, prediction_test)

print(classification_report(y_test, prediction_test))
print("----------------------------------------------------")
print("Random Forest accuracy is :",rf_accuracy)
print("----------------------------------------------------")
print('Confusion matrix for Random Forest :','\n',confusion_matrix(y_test,prediction_test))
```

iv. Support Vector Machine (SVM).

```
svm_model=SVC(kernel='linear',random_state = 42)
svm_model.fit(X_train,y_train)
predict_y = svm_model.predict(X_test)
accuracy_svm = svm_model.score(X_test,y_test)

print(classification_report(y_test, predict_y))
print("----------------------------------------------------")
print("SVM accuracy is :",accuracy_svm)
print("----------------------------------------------------")
print('Confusion matrix for SVM :','\n',confusion_matrix(y_test,predict_y))
```

v. XGBoost.

```
xgb_model = XGBClassifier( max_depth=3,n_estimators=50,random_state=42)
xgb_model.fit(X_train, y_train)
pred_model_y= xgb_model.predict(X_test)
accuracy_xgb=metrics.accuracy_score(y_test, pred_model_y)

print(classification_report(y_test, pred_model_y))
print("----------------------------------------------------")
print("XGB accuracy is :",accuracy_xgb)
print("----------------------------------------------------")
print('Confusion matrix for XGB :','\n',confusion_matrix(y_test,pred_model_y))
```

vi. Artificial Neural Network.

```
model = keras.Sequential([
    # input layer
    keras.layers.Dense(19, input_shape=(19,), activation='relu'),
    keras.layers.Dense(15, activation='relu'),
    keras.layers.Dense(10,activation = 'relu'),
    # we use sigmoid for binary output
    # output layer
    keras.layers.Dense(1, activation='sigmoid')
]
)

model.compile(optimizer = 'adam',
              loss = 'binary_crossentropy',
              metrics = ['accuracy'])
# now we fit our model to training data
model.fit(X_train,y_train,epochs=50)

ann_evaluation = model.evaluate(X_test, y_test)
ann_accuracy = ann_evaluation[1]

# predict the churn values
ypred = model.predict(X_test)
print(ypred)
# unscaling the ypred values
ypred_lis = []
for i in ypred:
    if i>0.5:
        ypred_lis.append(1)
    else:
        ypred_lis.append(0)
print(ypred_lis)
print(classification_report(y_test,ypred_lis))
print("-----------------------------------------------------")
print("ANN accuracy is :",ypred_lis)
print("-----------------------------------------------------")
print('Confusion matrix for ANN :','\n',confusion_matrix(y_test,ypred_lis))
```

- **Model Evaluation:**

  After Model is Trained, we evaluated on the test dataset to determine its accuracy and performance using different techniques like classification report, F1 score, precision, recall etc.

M.L. Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	80%	83%	76%	79%
Decision Tree	79%	80%	79%	80%
Random Forest	82%	85%	78%	81%
Support Vector Machine	80%	83%	75%	79%
XGBoost	83%	85%	81%	83%
ANN	82%	84%	81%	82%

To improve model Accuracy, we have use Hyperparameter Tunning Technique.

- **Hyperparameter Tunning: -**

  Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters.

  - **Hyperparameter Tuning techniques: -**

    Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. The best strategies for Hyperparameter tuning are:

    1. GridSearchCV
    2. RandomizedSearchCV
    3. Bayesian Optimization

  We have used **GridSearchCV** Technique for our project.

- **GridSearchCV**

  Grid search can be considered as a "brute force" approach to hyperparameter optimization. We fit the model using all possible combinations after creating a grid of potential discrete hyperparameter values. We log each set's model performance and then choose the combination that produces the best results. This approach is called GridSearchCV, because it searches for the best set of hyperparameters from a grid of hyperparameters values.

  In our Project, we have used **GridSearchCV** techniques for two Models:

  1. Random Forest
  2. XGBoost

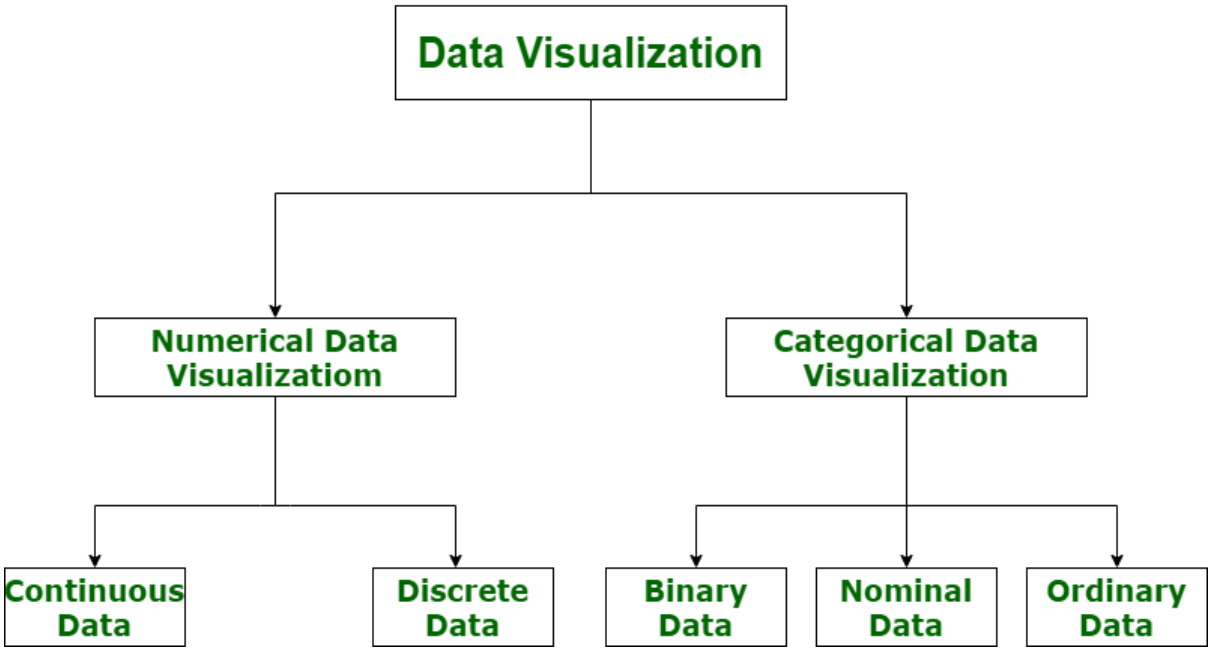
After Using GridSearchCV

M.L. Algorithm	Accuracy	Precision	Recall	F1-Score
Random Forest	84%	85%	84%	84%
XGBoost	84%	85%	83%	84%

Here we can clearly see that, After using Hyperparameter technique, Model accuracy is improve.

Chapter 5: Data Visualisation

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

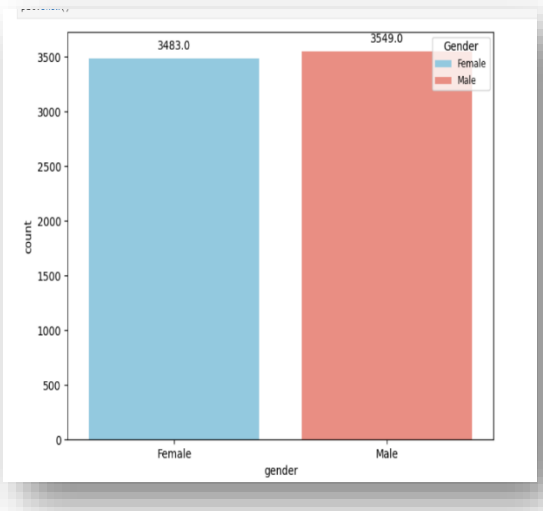


- **Data Visualization Tools:**

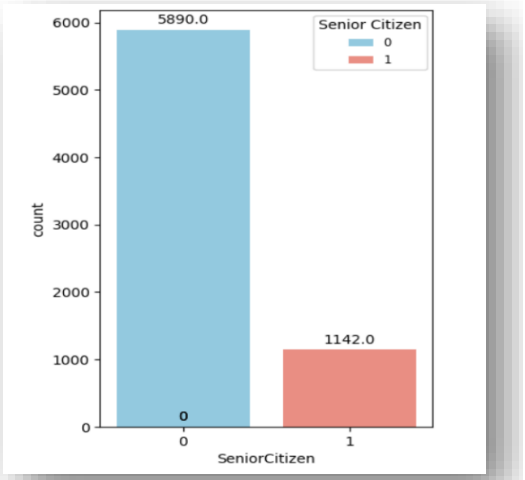
  The following are some Data Visualization Tools

  1. Tableau
  2. Looker
  3. Zoho Analytics
  4. Sisense
  5. IBM Cognos Analytics
  6. Qlik Sense
  7. Domo
  8. Microsoft Power BI
  9. Klipfolio
  10. SAP Analytics Cloud

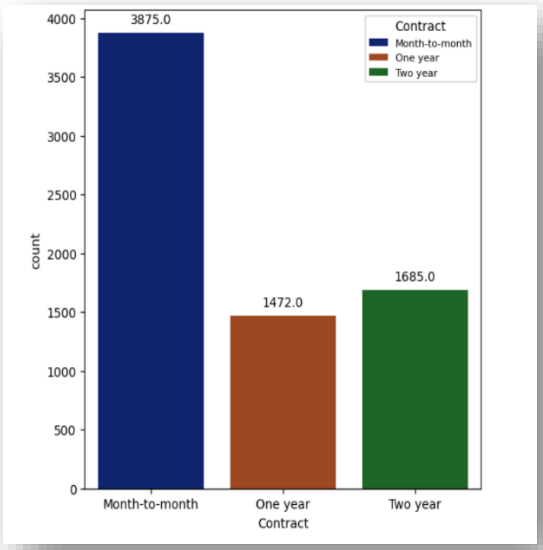
In our project, we had use Matplotlib library, Seaborn library and Tableau.



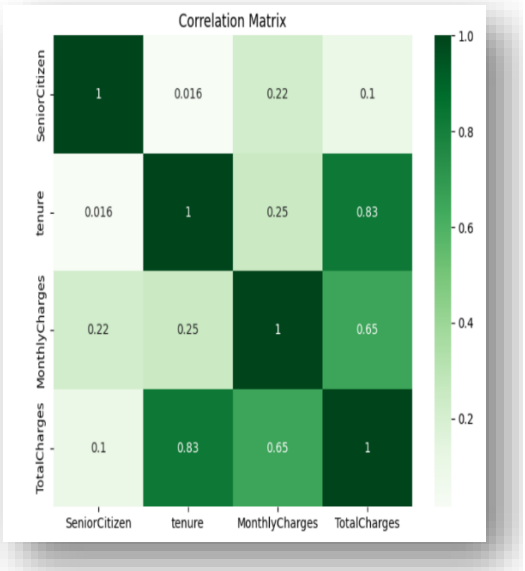
Gender wise Count



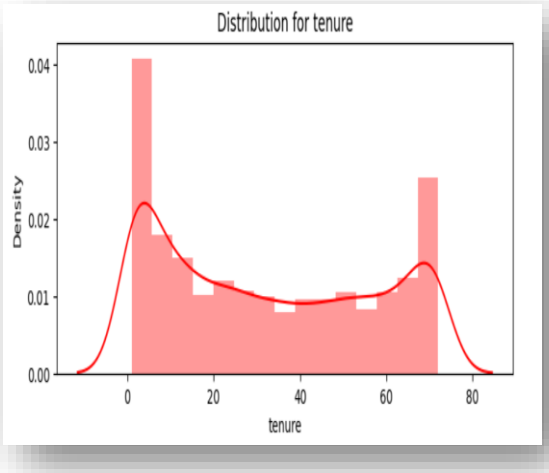
Senior Citizen Count



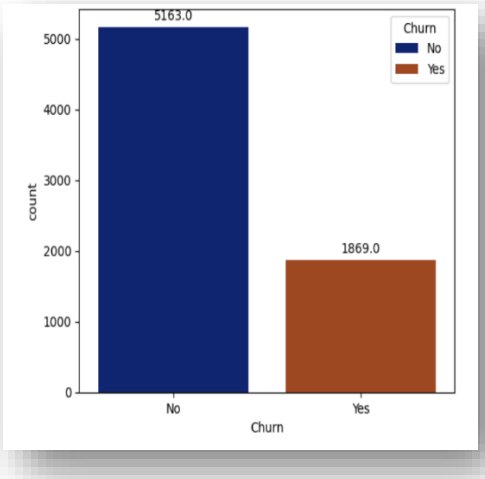
X`Contract Wise Count



Correlation Matrix



Histogram



Churn Count

Conclusion

In conclusion, this project aimed to predict customer churn in the telecom industry using a machine learning model. The dataset was cleaned by removing duplicates and imputing missing values. EDA was performed to analyse the distribution of the independent variables and the target variable. Six models were used to predict customer churn, and the XGBoost model had the highest accuracy and F1-score.

Hyperparameter tuning proved to be an effective strategy for improving the performance of the Random Forest and XGBoost model in the Telecom Churn Prediction task. By fine-tuning the models hyperparameters, we achieved significant enhancement in the model's ability to accurately predict churn. These improvements demonstrate the importance of hyperparameter optimization and highlights its potential to boost the performance of machine learning models in real world scenarios.