CUSTOMER CHURN PREDICTION

Problem Definition and Design Thinking

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem.
Please think on a design and present in form of a document.

Project Definition: The project involves using IBM Cognos to predict customer churn and identify factors influencing customer retention. The goal is to help businesses reduce customer attrition by understanding the patterns and reasons behind customers leaving. This project includes defining analysis objectives, collecting customer data, designing relevant visualizations in IBM Cognos, and building a predictive model.

Design Thinking:

1. Analysis Objectives: Define the specific objectives of predicting customer churn, such as identifying potential churners and understanding the key factors contributing to churn.
2. Data Collection: Determine the sources and methods for collecting customer data, including customer demographics, usage behavior, and historical interactions.
3. Visualization Strategy: Plan how to visualize the insights using IBM Cognos, showcasing factors affecting churn and retention rates.
4. Predictive Modeling: Decide on the machine learning algorithms and features to use for predicting customer churn.

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

Creating a design to **analyse customer churn** and implementing it involves several steps, from defining the problem to implementing a solution. Here's a detailed guide on how to go about it:

Step 1: Problem Definition and Understanding

1. Define the Objective: Clearly state the goal of your customer churn analysis, such as predictingwhich customers are likely to churn.

2. Understand Churn: Gain a deep understanding of what customer churn means for your business,what factors contribute to it, and why it's important to address.

3. Gather Stakeholder Requirements: Collaborate with stakeholders (e.g., product managers, marketing teams) to understand their requirements and expectations regarding the churn analysis.

Step 2: Data Collection and Preparation

1. Identify Data Sources: Determine the sources of data relevant to customer churn, such ascustomer profiles, usage patterns, transaction history, etc.

2. Data Cleaning and Integration: Clean the data to remove errors, duplicates, and irrelevant entries.Integrate data from various sources into a cohesive dataset.

3. Feature Engineering: Identify relevant features (attributes) that might impact customer churn (e.g., customer demographics, usage patterns, satisfaction scores) and engineer new features if needed.

Step 3: Exploratory Data Analysis (EDA)

1. Descriptive Statistics: Analyze basic statistics of the data to understand its distribution andcharacteristics.

2. Visualizations: Create various plots and visualizations (e.g., histograms, scatter plots, correlationmatrices) to identify patterns and relationships in the data.

3. Identify Key Factors: Use EDA to identify the factors most likely to influence customer churn.

Step 4: Model Selection and Development

1. Model Selection: Choose appropriate machine learning models for the churn prediction task (e.g.,logistic regression, decision trees, random forests, neural networks).

2. Train-Test Split: Divide the dataset into training and testing sets to evaluate the model's performance.

3. Model Training: Train the chosen model using the training data, optimizing hyperparameters forbest performance.

Step 5: Model Evaluation

1. Performance Metrics: Evaluate the model's performance using relevant metrics (e.g., accuracy,precision, recall, F1-score) to assess its predictive capabilities.

2. Fine-Tuning: If needed, fine-tune the model or experiment with different algorithms to improve performance.

Step 6: Implementation and Deployment

1. Model Integration: Integrate the trained and validated model into the organization's existing systems or platforms.

2. Real-Time Prediction: Set up a mechanism for real-time prediction using the deployed model to predict customer churn as new data becomes available.

3. Monitoring and Maintenance: Continuously monitor the model's performance in the live environment, retraining it periodically to ensure it remains accurate and effective.

Step 7: Interpretation and Actionable Insights

1. Interpret Results: Interpret the model's predictions and understand the factors contributing tocustomer churn as identified by the model.

2.Recommendations: Generate actionable insights and recommendations based on the model'sfindings to mitigate churn (e.g., targeted marketing strategies, product improvements).

ANALYSIS IMPLEMENTATION:

```
#import all relevant libraries

import numpy as np import
pandas as pd
import matplotlib.pyplot as pltimport
seaborn as sns
import missingno as msnoimport
warnings
warnings.filterwarnings("ignore")
from sklearn.experimental import enable_iterative_imputerfrom
sklearn.impute import IterativeImputer
from sklearn.preprocessing import LabelEncoder
```

```
#loading the dataset
data=pd.read_csv("C:\Users\HP\Downloads\Telco-Customer-Churn.csv”)
```

Data Collection:

- We will collect customer churn prediction data from reputable sources pertinent to our industry, such as internal databases, customer records, and transaction histories.
- The primary data source for this project will be our company's customer database, encompassing historical customer interactions, subscription details, and churn-related information.
- Daily data updates will be drawn from our internal customer records, allowing us to maintain a real-time understanding of churn.
- Subsequently, this data will be merged with additional information, such as demographic data, to enrich our analysis and develop a more comprehensive customer churn prediction model.

```
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	Device
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No

```
<class 'pandas.core.frame.DataFrame'>
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)
memory usage: 1.1+ MB
```

```
data.describe()
```

	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

```
data[[col for col in data.columns.difference(num_cols) if col != 'seniorcitizen']
].describe().T
```

	count	unique	top	freq
churn	7043	2	No	5174
contract	7043	3	Month-to-month	3875
dependents	7043	2	No	4933
deviceprotection	7043	3	No	3095
gender	7043	2	Male	3555
internetservice	7043	3	Fiber optic	3096
multiplelines	7043	3	No	3390
onlinebackup	7043	3	No	3088
onlinesecurity	7043	3	No	3498
paperlessbilling	7043	2	Yes	4171
partner	7043	2	No	3641
paymentmethod	7043	4	Electronic check	2365
phoneservice	7043	2	Yes	6361
streamingmovies	7043	3	No	2785
streamingtv	7043	3	No	2810

Data Preprocessing

- Data cleaning and preprocessing constitute crucial phases in the preparation of data for analysis.
- During this stage, we will address data issues including duplicate entries, inconsistent f ormats, the treatment of missing values, and the conversion of categorical variables into numerical formats.

```
data.dtypes
```

```
customerID           object
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges         object
Churn                object
dtype: object
```

```
data.isnull().sum()
```

```
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
```

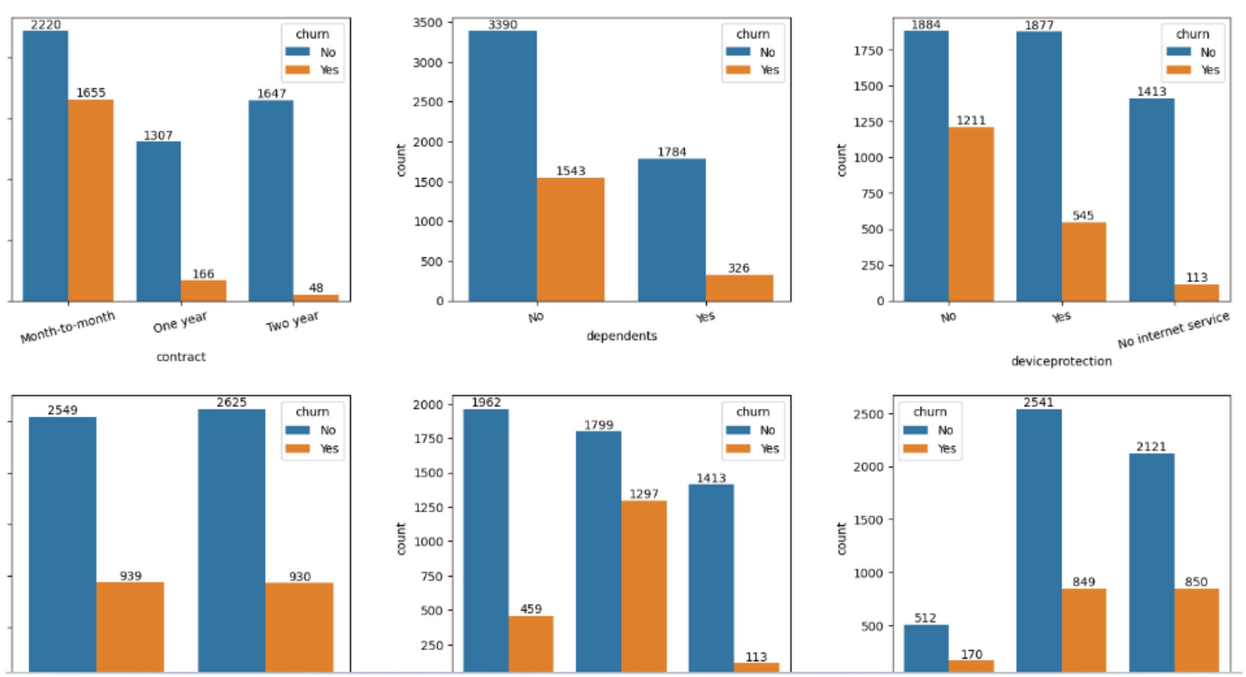
Data Exploration

- Engage in exploratory data analysis (EDA) to grasp the dataset's characteristics, pattern s, and relationships.
- In this phase, the objective is to delve into the data to comprehend its nuances. EDA enta ils computing statistical summaries, creating data visualizations, and recognizing patter ns and anomalies.
- Key areas of exploration encompass customer demographics, historical usage patterns, a nd the impact of various features on churn predictions.
- Utilize visualizations to gain insights into the distribution of key features and the identif ication of influential factors affecting customer churn.

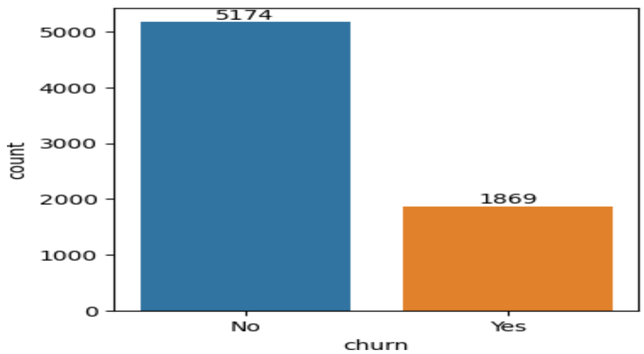
```
#data cleaning  data transformation  data reduction
#drop irrelevant variables
data=data.drop(['CustomerId'],axis=1)
#identifying and treating missing values
data.isnull().sum()
data=data.fillna(0)

data.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Y
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	N

```
num_cols = ['tenure','monthlycharges', 'totalcharges']

label="Churn"

plt.figure(figsize = (15, 26))

for i, col in enumerate(data.columns.difference(num_cols)[1:]):plt.subplot(6, 3, i+1)

    ax = sns.countplot(data, x = col, hue = label)

    ax.bar_label(ax.containers[0])

    ax.bar_label(ax.containers[1]) plt.xticks(rotation

    = 15)

plt.tight_layout()

plt.show()
```



```
plt.figure(figsize = (4,4))
ax = sns.countplot(data, x = label)
ax.bar_label(ax.containers[0])
plt.show()
```



PREDICTIVE MODEL:

LOGISTIC REGRESSION

```
In [1]: y = df_dummies['Churn'].values
        X = df_dummies.drop(columns = ['Churn'])

        # Scaling all the variables to a range of 0 to 1
        from sklearn.preprocessing import MinMaxScaler
        features = X.columns.values
        scaler = MinMaxScaler(feature_range = (0,1))
        scaler.fit(X)
        X = pd.DataFrame(scaler.transform(X))
        X.columns = features
```

It is important to scale the variables in logistic regression so that all of them are within arange of 0 to 1. This helped me improve the accuracy from 79.7% to 80.7%.

```
In [2]: from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

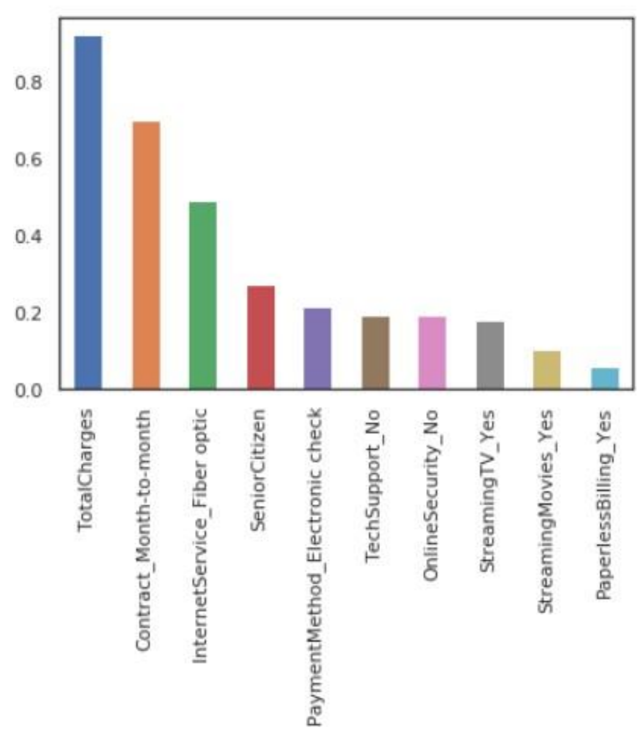
In [ ]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
result = model.fit(X_train, y_train)
```

In [3]:
```
from sklearn import metrics
prediction_test = model.predict(X_test)
# Print the prediction accuracy
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.8075829383886256
```

In [4]:
```
# To get the weights of all the variables
weights = pd.Series(model.coef_[0],
                 index=X.columns.values)
print (weights.sort_values(ascending = False)[:10].plot(kind='bar'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



```
In [5]: print(weights.sort_values(ascending = False)[-10:].plot(kind='bar'))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```