ANAND INSTITUTE OF HIGHER TECHNOLOGY
OLD MAHABALIPURAM ROAD,
KALASALINGAM NAGAR, KAZHIPATTUR – 603103

CUSTOMER CHURN PREDICTION WITH DATA ANALYTICS USING COGNOS

PHASE – 5

NAME : ROSHINI M

REG No. : 310121104086

BRANCH : COMPUTER SCIENCE & ENGINEERING

YEAR/SEM : III / V

IBM DATA ANALYTICS WITH COGNOS

TEAM NAME : Proj_229798_Team_1

PROJECT : 3101-Customer Churn Prediction

TEAM MEMBERS :

1. Nokitha V M
2. Mounika V
3. Yasmeen U
4. Roshini

LEADER : Nokitha V M

Customer Churn Prediction

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.

Data Collection and Preprocessing :

1.Identify Key Data :

Dive into the essential data points required to effectively predict customer churn. Understand the importance of data quality and data cleansing techniques

2.Data Preprocessing :

Explore the steps involved in preparing the data for analysis. Learn about techniques such as data normalization, feature engineering, and handling missing values

3.Feature Selection :

Discover various methods to identify the most relevant features for predicting churn. Understand the impact of feature selection on model performance.

Exploratory Data Analysis :

Visualizing Data:

Learn how to effectively visualize and analyze data to gain insights into customer behavior and potential churn indicators. Use charts and graphs to tell the story.

Sentiment Analysis :

Uncover the power of sentiment analysis in predicting customer churn. Learn how to extract meaningful information from customer feedback and reviews.

Customer Journey Mapping :

Map out the customer journey to identify critical touchpoints and moments of truth. Understand the key stages where customers are at risk of churning.

Design Thinking:

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

NumPy :

NumPy, which stands for "Numerical Python," provides support for working with large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays efficiently. NumPy is a core library in the Python data.

- ❖ Data Loading
- ❖ Data Preparation
- ❖ Data Exploration
- ❖ Numerical Operations

Pandas DataFrame :

A pandas DataFrame is a two-dimensional, labelled data structure commonly used in data analysis and manipulation within the Python programming language. It is one of the fundamental data structures provided by the pandas library.

- ❖ Data Import
- ❖ Data Exploration
- ❖ Data Manipulation
- ❖ Time Series Analysis

Matplotlib :

Matplotlib is a widely-used Python library for creating static and interactive visualizations and plots. It is a powerful tool for data visualization and is particularly popular in the fields of data analysis, scientific computing, and machine learning.

- ❖ Data Visualization
- ❖ Exploratory Data Analysis(EDA)
- ❖ Performance Metrics
- ❖ Customization and Interactivity

Jupyter Notebook Interface

Machine Learning Algorithms in Jupyter Notebook :

Showcasing the implementation of KNN, CNN, and Gradient Descent algorithms in Jupyter Notebook.

Executing Code in Jupyter Notebook :

Witness the step-by-step execution of code snippets within the Jupyter Notebook environment.

Visualizing Data in Jupyter Notebook :

Demonstrating the power of data visualization using Jupyter Notebook's interactive capabilities.

Using Machine Learning Algorithms :

- KNN, CNN, and Gradient Descent are some of the most popular machine learning algorithms used for data analysis. In this section, we will discuss how these algorithms are useful to transportation efficiency data's analysis and their results.
- KNN can help in real-time traffic management and route optimization, CNN can provide automated insights from visual data for better traffic monitoring, and Gradient Descent plays a crucial role in training predictive models for various transportation-related applications.

- The results of these algorithms can lead to more efficient transportation systems, reduced congestion, lower costs, and improved overall transportation experiences.

DEVELOPMENT PART

- ❖ Start building the customer churn prediction using IBM Cognos for visualization. Define the analysis objectives and collect customer data from the source shared.
- ❖ Process and clean the collected data to ensure its quality and accuracy.
- ❖ This process involves collecting, cleaning, transforming, reduction of null values, visualization, scalability, efficiency and structuring raw data to make it suitable for analysis.
- ❖ Both qualitative and quantitative customer data are usually needed to start building an effective churn prediction model. To ensure that predictions aren't being made by arbitrary human guesses, these models are often built by a data scientist using machine learning.

DATA COLLECTION:

CUSTOMER CHURN PREDICTION is done by using the Dataset of "Telco Customer Churn" provided by the dataset site www.Kaggle.com

Dataset Link:

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

DATASET AND ITS DETAILS :

TITLE: CUSTOMER CHURN PREDICTION

LIBRARIES: sklearn, Matplotlib, pandas, seaborn, and NumPy

Context:

The dataset “CUSTOMER CHURN PREDICTION” on Kaggle is a collection of data related to the a Machine Learning Model That Can Predict Customers Who Will Leave The Company "Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs."

Content:

Each row represents a customer, each column contains customer’s attributes described on the column Metadata.

SIGNIFICANCE OF LOADING AND PREPROCESSING THE DATASET:

- Data Loading is defined as copying data from one electronic file or database into another. Data loading implies converting from one format into another; for example, from one type of production database into a decision support database from a different vendor.
- Data preprocessing is essential before its actual use. Data preprocessing is the concept of changing the raw data into a clean data set. The dataset is preprocessed in order to check missing values, noisy data, and other inconsistencies before executing it to the algorithm.

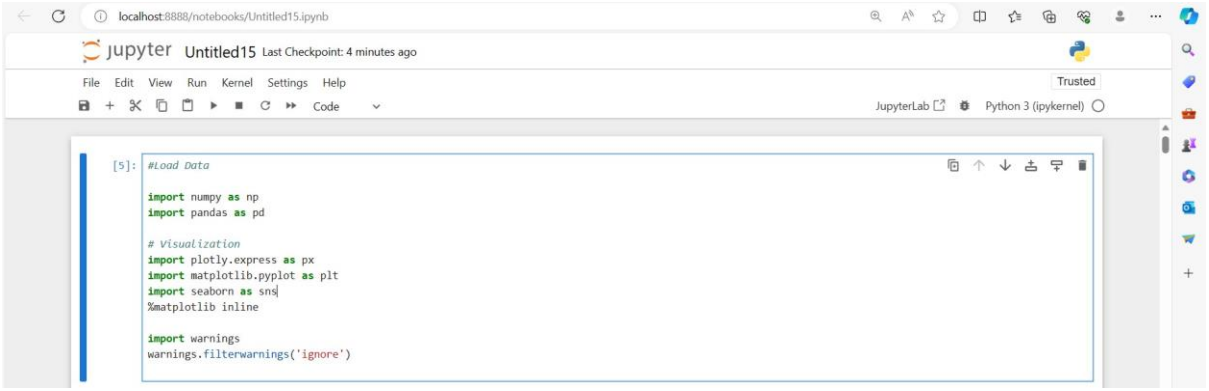
IMPORT AND LOAD THE DATASET :

Use Pandas to read the dataset file you downloaded into a DataFrame:

CODING AND ITS OUTPUT:

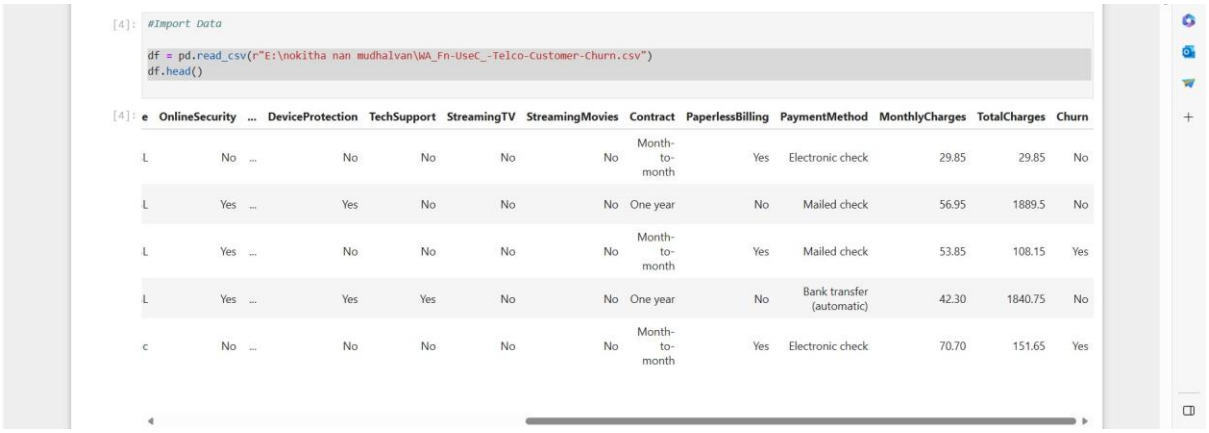
LOAD DATA:

```
import numpy as np
import pandas as pd

# Visualization
import plotly.express as px
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```



IMPORT DATA:

df = pd.read_csv(r"E:\nokitha nan mudhalvan\WA_Fn-UseC_-Telco-Customer-Churn.csv")
df.head()

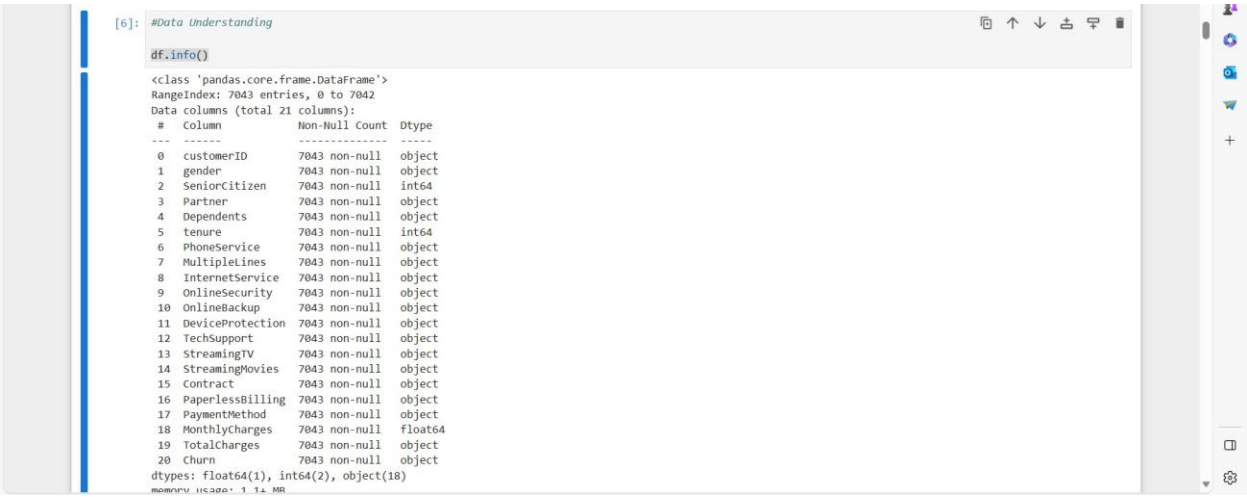


[4]:

e	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
L	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
L	Yes	...	Yes	No	No	No	One year	No	Mailed check	56.95	1889.5	No
L	Yes	...	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
L	Yes	...	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
c	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

DATA UNDERSTANDING:

```
df.info()
```



```
[6]: #Data Understanding

     df.info()

     <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: 1.14 MB
```

Check the Duplicate:

```
print(df.duplicated().value_counts())
```

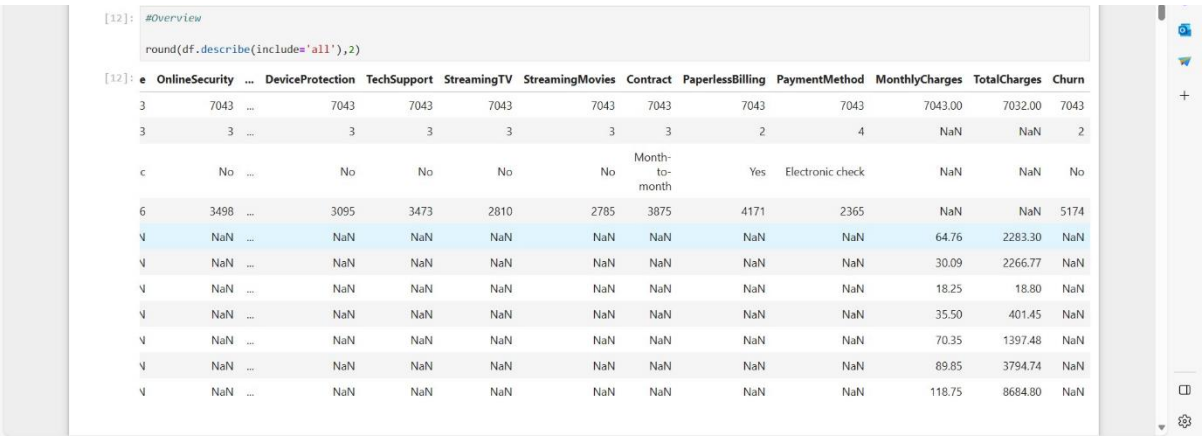
Check the missing values:

```
df.isnull().values.any()
```

```
[8]: #Check the Duplicate

     print(df.duplicated().value_counts())

     False    7043
     Name: count, dtype: int64

[9]: #Check the missing Values

     df.isnull().values.any()

[9]: True
```

Overview:

```
round(df.describe(include='all'),2)
```



```
[12]: #Overview

      round(df.describe(include='all'),2)
```

[12]:

e	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
3	7043	...	7043	7043	7043	7043	7043	7043	7043	7043.00	7032.00	7043
3	3	...	3	3	3	3	3	2	4	NaN	NaN	2
c	No	...	No	No	No	No	Month-to-month	Yes	Electronic check	NaN	NaN	No
6	3498	...	3095	3473	2810	2785	3875	4171	2365	NaN	NaN	5174
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	64.76	2283.30	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	30.09	2266.77	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	18.25	18.80	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	35.50	401.45	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	70.35	1397.48	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	89.85	3794.74	NaN
N	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	118.75	8684.80	NaN

Target value:

 # Count the oocurance of unique values in the 'Churn' Column

```
churn_counts =  df.Churn.value_counts(normalize=True)
churn_counts
```

```
[15]: #Target Value

      # Count the oocurance of unique values in the 'Churn' Column

      churn_counts =  df.Churn.value_counts(normalize=True)
      churn_counts
```

```
[15]: Churn
      No     0.73463
      Yes    0.26537
      Name: proportion, dtype: float64
```

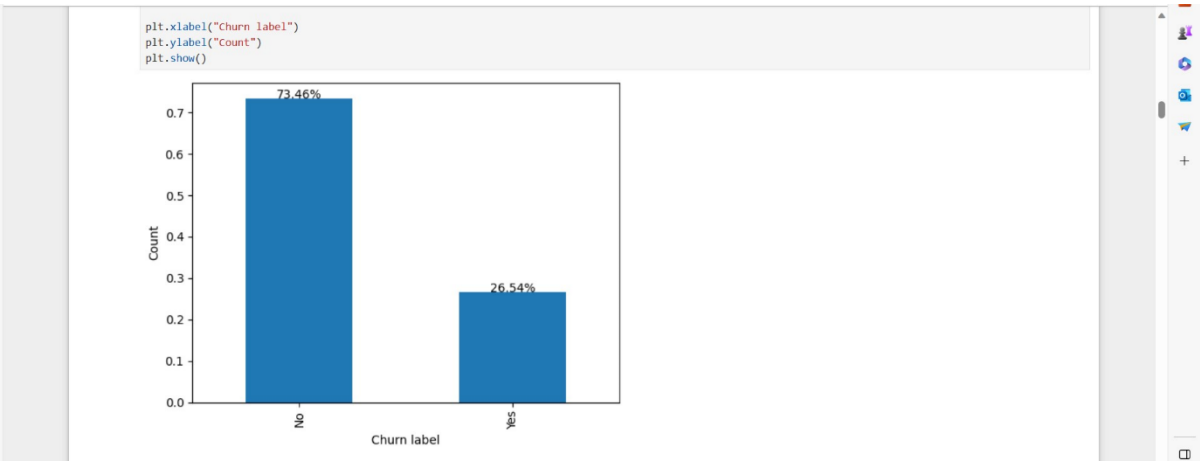
 # Calculate the percentage of 'Yes' and 'No' label

```
total_count = churn_counts.sum()

percentage_yes = (churn_counts['Yes']/ total_count) * 100
percentage_no = (churn_counts['No']/ total_count) *100
```

 # Plot the target value

```
ax = churn_counts.plot(kind='bar')
```

```
# Annotate the bars with percentages

for i, count in enumerate(churn_counts):
    percentage = percentage_yes if i == 1 else percentage_no
    ax.annotate(f'{percentage:.2f}%', xy=(i, count), ha='center')


plt.xlabel("Churn label")
plt.ylabel("Count")
plt.show()
```



EDA - Exploratory Data Analysis on each feature:

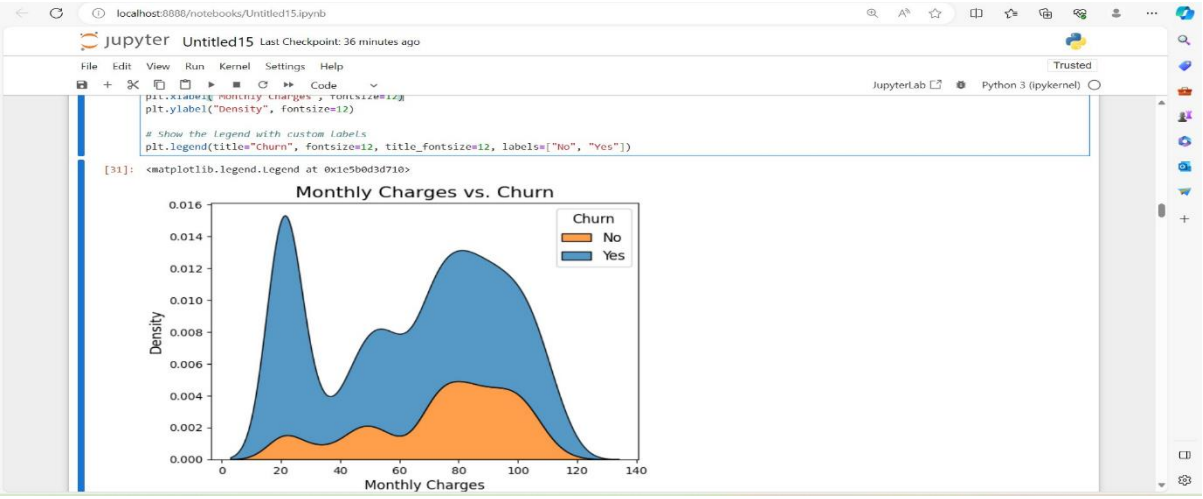
It is observed that the dataset exhibits a significant class imbalance, with a larger amount of data representing non-churners.

Is there a correlation between churn and factors such as monthly charges and total charges?

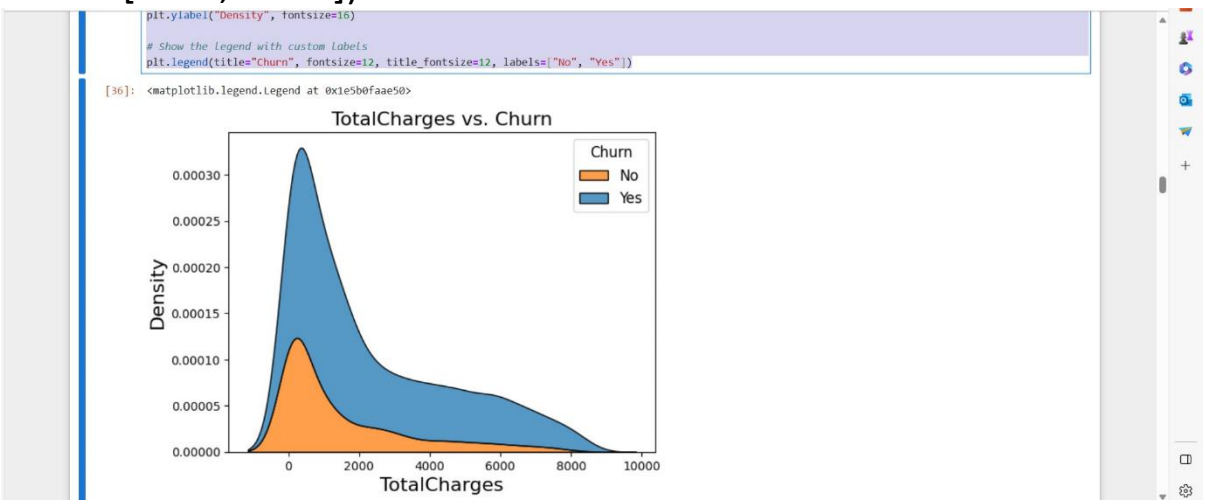
```
df[['MonthlyCharges','TotalCharges']]
```

```
df.groupby(['MonthlyCharges','TotalCharges'])['Churn'].size()
```

```
sns.kdeplot(data=df,
x="MonthlyCharges",hue="Churn",multiple="stack")

# Customize the plot appearance

plt.title("Monthly Charges vs. Churn", fontsize=16)

plt.xlabel("Monthly Charges", fontsize=12)

plt.ylabel("Density", fontsize=12)

# Show the legend with custom labels

plt.legend(title="Churn", fontsize=12, title_fontsize=12, labels=["No",
"Yes"])
```



- Note: it's noticeable that as monthly charges increase within the range of 60 to 120, the density also rises. This trend indicates a higher rate of churn as monthly charges increase.

- sns.kdeplot(data=df, x="TotalCharges",hue="Churn",multiple="stack")
- # Customize the plot appearance
- plt.title("TotalCharges vs. Churn", fontsize=16)
- plt.xlabel("TotalCharges", fontsize=16)
- plt.ylabel("Density", fontsize=16)
- # Show the legend with custom labels
- plt.legend(title="Churn", fontsize=12, title_fontsize=12, labels=["No", "Yes"])
- 

- Note: High churn rates are associated with lower total charges, with the highest churning occurring in the 0-2000 total charges range.

2. How does the length of a customer's tenure with the company influence their likelihood of churning?

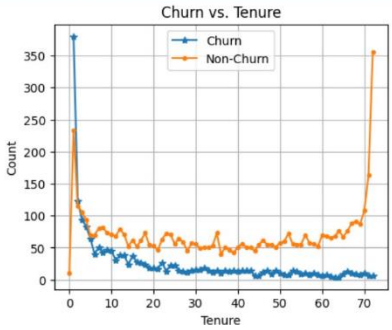
grouped_data = df.groupby(['tenure', 'Churn']).size().reset_index(name='count')

grouped_data[30:40]

Separate churn and non-churn counts

```
churn_data = grouped_data[grouped_data['Churn'] == 'Yes']

non_churn_data = grouped_data[grouped_data['Churn'] == 'No']

# Create a line chart for churn and non-churn counts

plt.figure(figsize=(5, 4))

plt.plot(churn_data['tenure'], churn_data['count'], label='Churn',
marker='*')

plt.plot(non_churn_data['tenure'], non_churn_data['count'],
label='Non-Churn', marker='.')


plt.xlabel('Tenure')

plt.ylabel('Count')

plt.title('Churn vs. Tenure')

plt.legend()

plt.grid(True)

plt.show()
```

```
plt.plot(churn_data['tenure'], churn_data['count'], label='Churn', marker='*')
plt.plot(non_churn_data['tenure'], non_churn_data['count'], label='Non-Churn', marker='.')

plt.xlabel('Tenure')
plt.ylabel('Count')
plt.title('Churn vs. Tenure')
plt.legend()
plt.grid(True)
plt.show()
```



Note:

Customers with the low tenure **"0-10”** has the highest rate of churning,these range can be crucial for business decisions.

In general the **Non-Churn** line has small fluctuation and remains **relatively stable**, with longer tenure tend to stay with the company.

The Churn line decreases or remains relatively stable as tenure increases, it suggests that **customer loyalty** increases with longer tenure

3. connection between gender, partner status, and churn

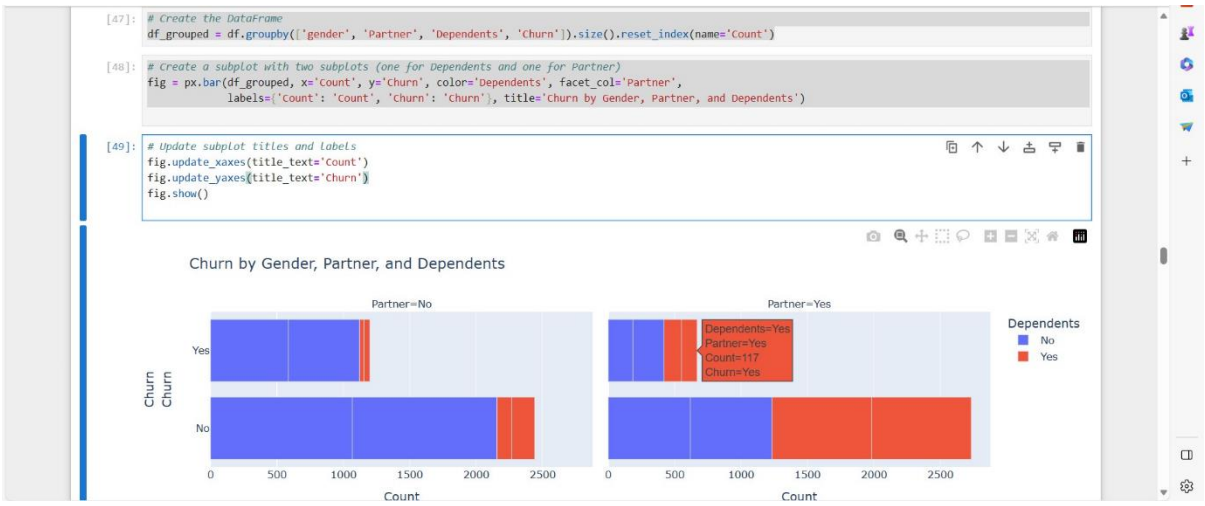
df.gender.value_counts(normalize=True)

Create the DataFrame

df_grouped = df.groupby(['gender', 'Partner', 'Dependents', 'Churn']).size().reset_index(name='Count')

Create a subplot with two subplots (one for Dependents and one for Partner)

```
fig = px.bar(df_grouped, x='Count', y='Churn', color='Dependents',
facet_col='Partner',
              labels={'Count': 'Count', 'Churn': 'Churn'}, title='Churn by
Gender, Partner, and Dependents')

# Update subplot titles and labels

fig.update_xaxes(title_text='Count')

fig.update_yaxes(title_text='Churn')

fig.show()
```



Note:

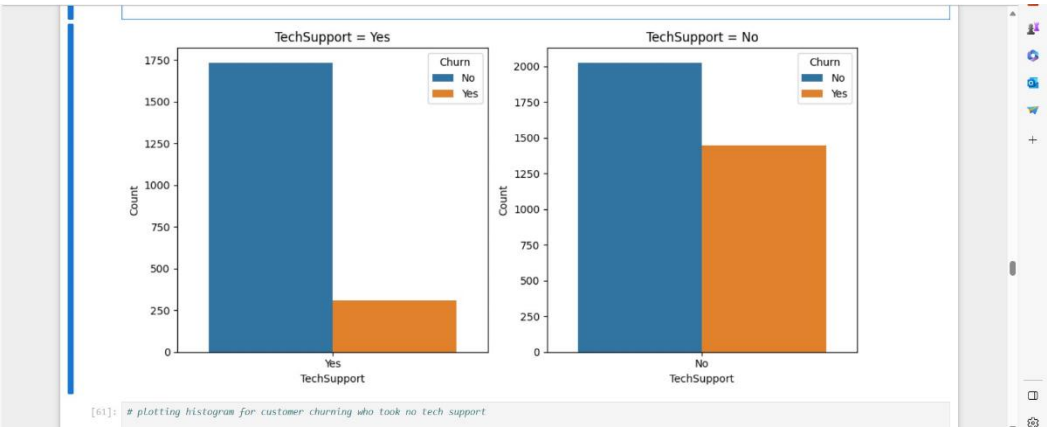
Regardless of gender, individuals who lack a partner or dependents are at a higher risk of churning.

5.Does the availability of technical support play a role in influencing customer churn? and does the duration of being customer

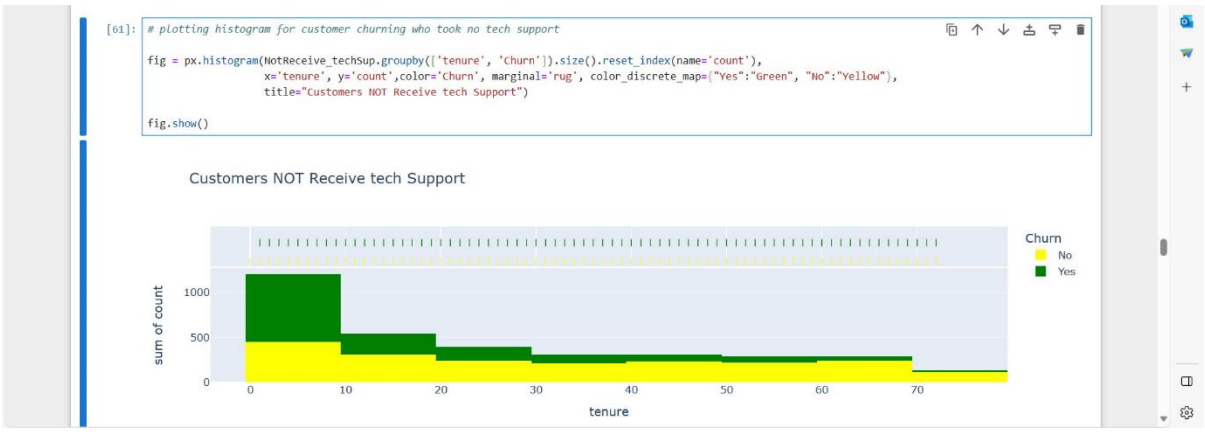
```
# Create subplots to compare the distribution

fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Plot the distribution of 'TechSupport' for 'Yes' category

sns.countplot(data=Receive_techSup, x='TechSupport',
hue='Churn', ax=axes[0])

axes[0].set_title('TechSupport = Yes')

axes[0].set_xlabel('TechSupport')

axes[0].set_ylabel('Count')

# Plot the distribution of 'TechSupport' for 'No' category

sns.countplot(data=NotReceive_techSup, x='TechSupport',
hue='Churn', ax=axes[1])

axes[1].set_title('TechSupport = No')

axes[1].set_xlabel('TechSupport')

axes[1].set_ylabel('Count')

# Adjust layout

plt.tight_layout()

plt.show()
```

Note:

Customers who do not receive tech support are more likely to churn.



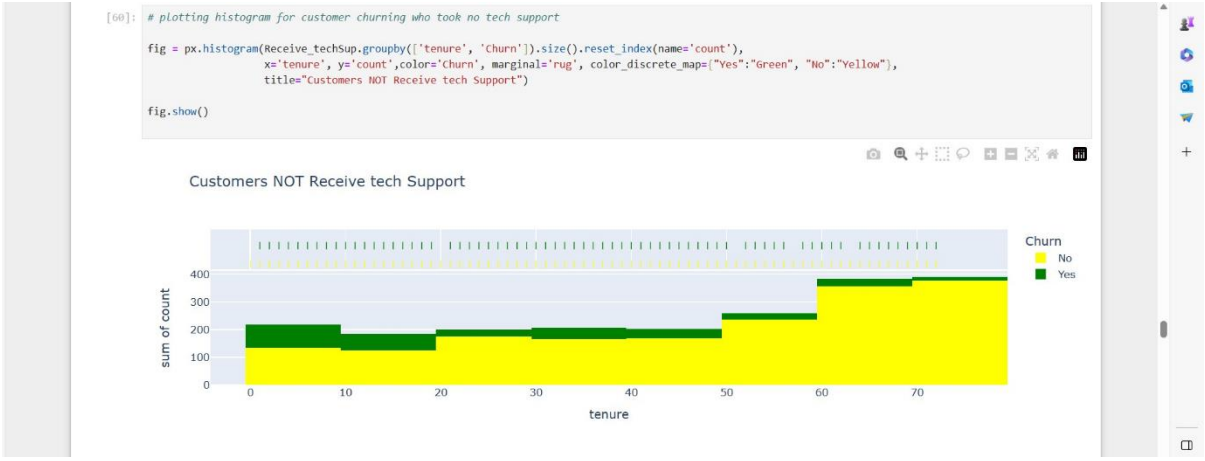

```
# plotting histogram for customer churning who took no tech support

fig = px.histogram(NotReceive_techSup.groupby(['tenure',
'Churn']).size().reset_index(name='count'),

                   x='tenure', y='count',color='Churn', marginal='rug',
color_discrete_map={"Yes":"Green", "No":"Yellow"},

                   title="Customers NOT Receive tech Support")

fig.show()
```



plotting histogram for customer churning who took no tech support

fig = px.histogram(Receive_techSup.groupby(['tenure', 'Churn']).size().reset_index(name='count'),
x='tenure', y='count',color='Churn', marginal='rug', color_discrete_map={"Yes":"Green", "No":"Yellow"},
title="Customers NOT Receive tech Support")

fig.show()



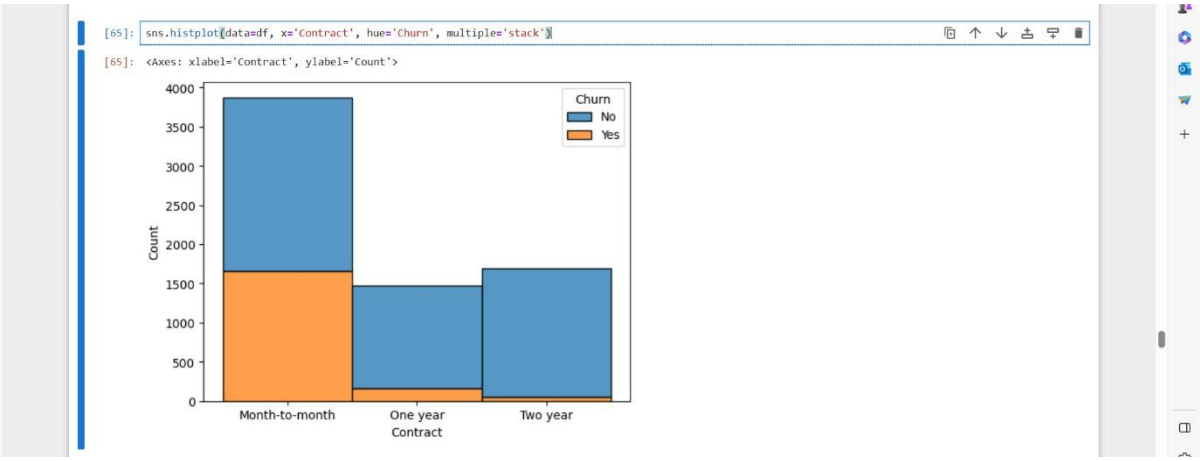
Note:

The data indicates that churn rates are highest within the first year of service, especially among customers without tech support. Conversely, longer tenure is associated with increased customer loyalty.

5 .Which aspect of the contract has the most significant impact on the business?

df.Contract.value_counts(normalize =True)

Contract_condition =
df.groupby(['Churn','Contract']).size().reset_index(name='count')
Contract_condition
sns.histplot(data=df, x='Contract', hue='Churn', multiple='stack')

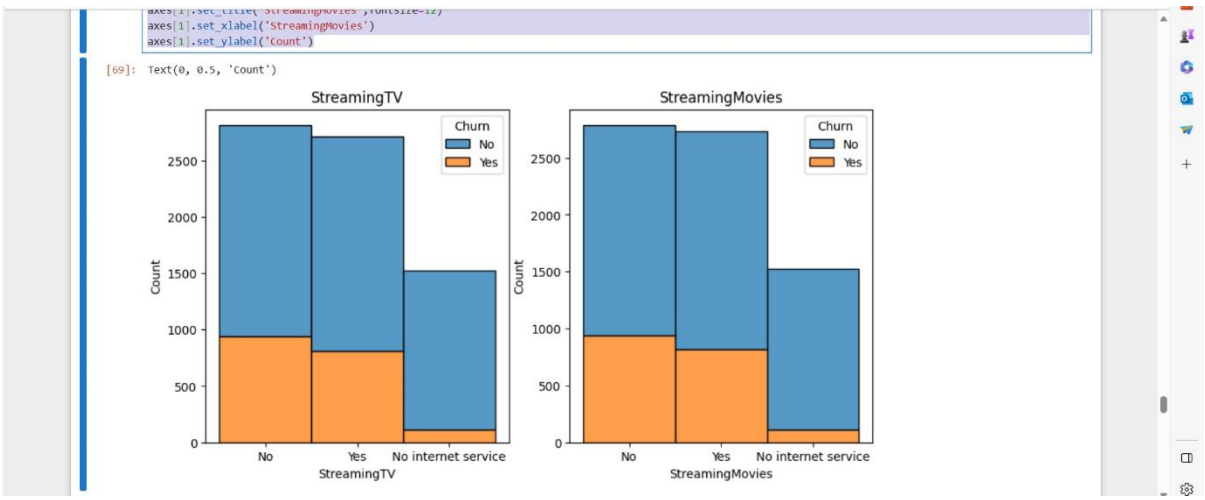


Note:
It is evident that customers with month-to-month contracts have the highest churn rates.

6 .How does the quality of service differ for customers who have opted for streaming services?

df.StreamingTV.value_counts()

```
Streming = df.groupby
(['Churn','StreamingTV']).size().reset_index(name='count')
Streming


ig, axes = plt.subplots(1,2,figsize = (10,5))

# Plot the distribution of 'StreamingTV'
sns.histplot(data=df, x='StreamingTV', hue='Churn',
multiple='stack',ax=axes[0])
axes[0].set_title('StreamingTV',fontsize=12)
axes[0].set_xlabel('StreamingTV')
axes[0].set_ylabel('Count')

# Plot the distribution of 'StreamingMovies'
sns.histplot(data=df, x='StreamingMovies', hue='Churn',
multiple='stack',ax=axes[1])
axes[1].set_title('StreamingMovies',fontsize=12)
axes[1].set_xlabel('StreamingMovies')
axes[1].set_ylabel('Count')
```

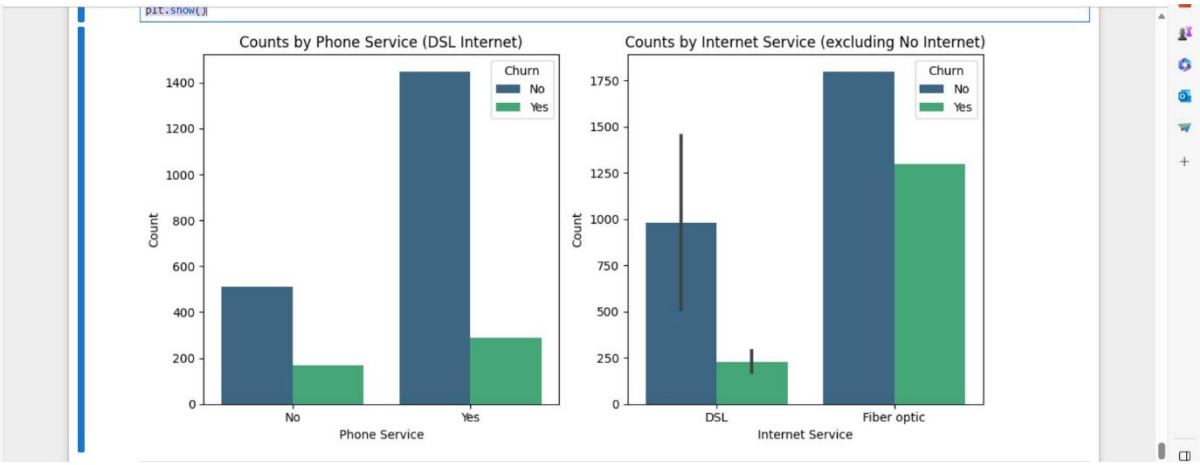


Note: Churn rates are similar for both the 'Yes' and 'No' groups in terms of whether customers are connected to StreamingTV and StreamingMovies.

7. Given that the dataset pertains to the telecom industry, what insights can we uncover regarding phone and internet services?

```
df.InternetService.value_counts(normalize=True)

df.PhoneService.value_counts(normalize=True)

# Group the data by "InternetService", "phoneService" and "Churn" and count the number of records

service_counts = df.groupby(['InternetService','PhoneService', 'Churn']).size().reset_index(name='Count')

service_counts

plt.figure(figsize=(10, 5))

# Create two subplots

plt.subplot(121)
# 1 row, 2 columns, the first plot for Phone Service

sns.barplot(data=service_counts[service_counts['InternetService'] == 'DSL'], x='PhoneService', y='Count', hue='Churn', palette='viridis')
plt.xlabel('Phone Service')
plt.ylabel('Count')
plt.title('Counts by Phone Service (DSL Internet)')

plt.subplot(122)
```

```
# 1 row, 2 columns, the second plot for Internet Service

sns.barplot(data=service_counts[service_counts['InternetService'] !=
'No'], x='InternetService', y='Count', hue='Churn', palette='viridis')
plt.xlabel('Internet Service')
plt.ylabel('Count')
plt.title('Counts by Internet Service (excluding No Internet)')

plt.tight_layout()
plt.show()
```



Note:

The plot suggests that the customers with phone service are loyal and have not churned.(the blue bar is significantly taller than green).

For customers without phone service, the blue bar for non-churn is still higher than the green bar. This indicate that even among those without phone service, a significant portion has not churned, and the absence of phone service is associated with a lower churn rate.

PROBLEM DEFINITION

Customer churn is a crucial metric impacting a company's profitability and growth. Losing existing customers leads to revenue decline and incurs costs associated with acquiring new customers to replace the lost ones. High customer churn can also damage a company's reputation and hinder its long-term sustainability.

Required steps:

1 .univariate Analysis:

Univariate data requires to analyze each variable separately. Data is gathered for the purpose of answering a question.

2. Bivariate Analysis:

Bivariate analysis is stated to be an analysis of any concurrent relation between two variables or attributes.

3. Time series analysis:

Time Series models are based on data where we observe predictors and churn simultaneously as they occur period to period.

4. Dashboard creation:

A customer churn dashboard is a visual representation of key metrics and data related to customer churn within a business.

5. heatmap:

A Churn Heatmap is a visual summary of every account journey of your churned accounts - from customer acquisition to customer exit - scoring the company's performance at each step with a color code.

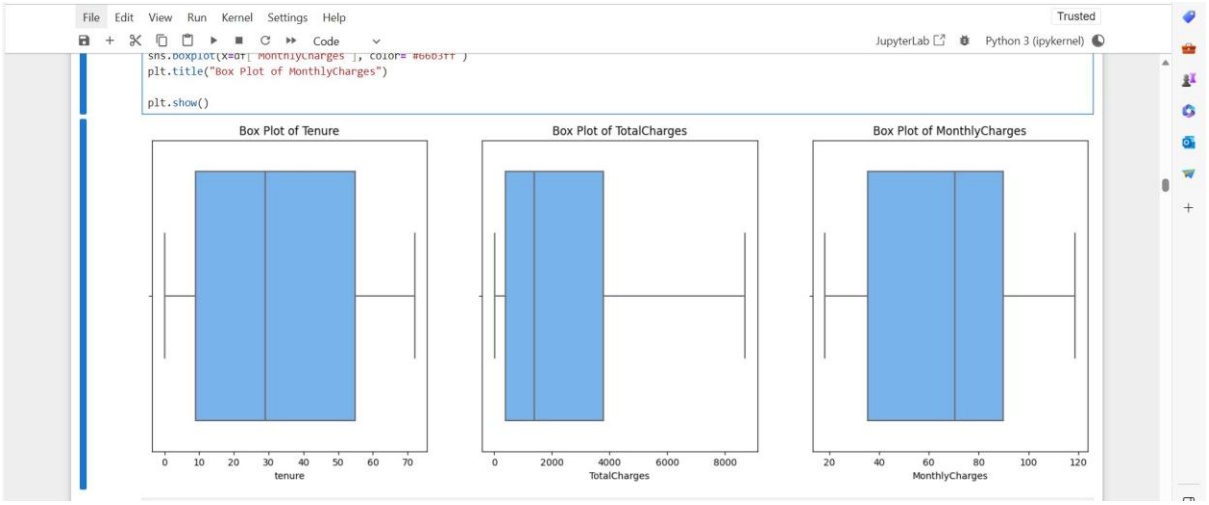
6. Cohort analysis:

The examination of different groups of users based on how they interact with your product.

Use machine learning algorithms to build a predictive model that identifies potential churners based on historical data and relevant features.

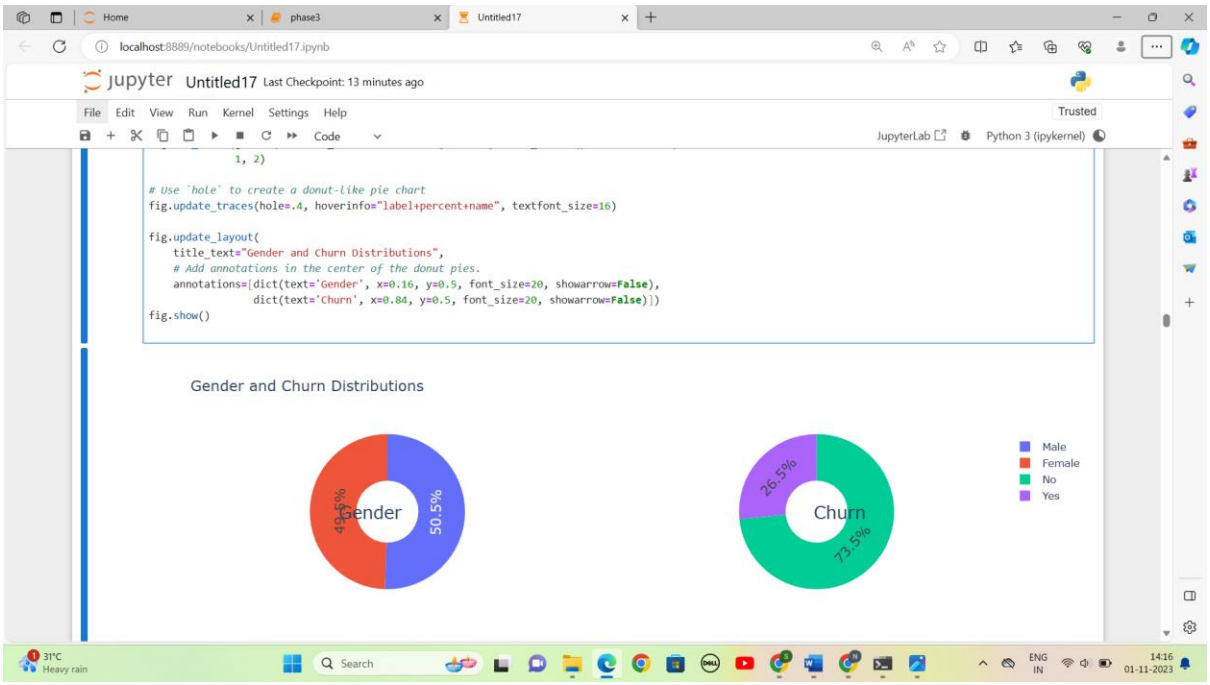
```
# Create separate box plots for 'tenure', 'TotalCharges', and 'MonthlyCharges'
plt.figure(figsize=(18, 6))

# Box Plot of 'tenure'
plt.subplot(131)  # 1 row, 3 columns, plot 1
sns.boxplot(x=df['tenure'], color='#66b3ff')
plt.title("Box Plot of Tenure")

# Box Plot of 'TotalCharges'
plt.subplot(132)  # 1 row, 3 columns, plot 2
```

```
sns.boxplot(x=df['TotalCharges'], color='#66b3ff')
plt.title("Box Plot of TotalCharges")

# Box Plot of 'MonthlyCharges'
plt.subplot(133)  # 1 row, 3 columns, plot 3
sns.boxplot(x=df['MonthlyCharges'], color='#66b3ff')
plt.title("Box Plot of MonthlyCharges")

plt.show()
```



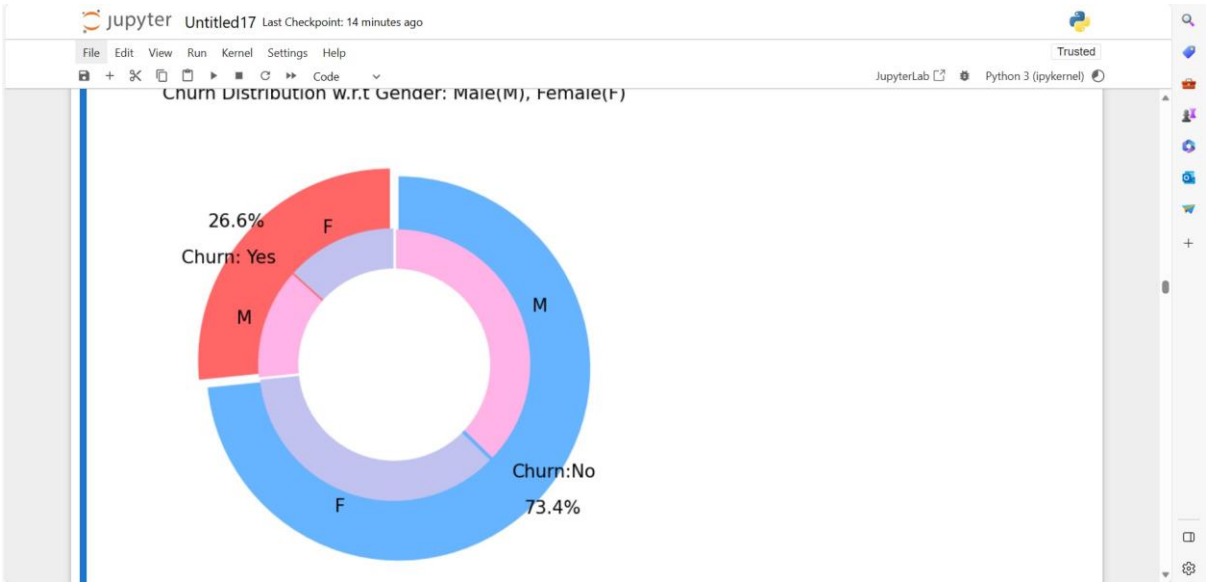

```
g_labels = ['Male', 'Female']
c_labels = ['No', 'Yes']

# Create subplots: use 'domain' type for Pie subplot
fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'},
{'type':'domain'}]])
```

```
fig.add_trace(go.Pie(labels=g_labels,
values=df['gender'].value_counts(), name="Gender"),
              1, 1)
fig.add_trace(go.Pie(labels=c_labels,
values=df['Churn'].value_counts(), name="Churn"),
              1, 2)


# Use `hole` to create a donut-like pie chart
fig.update_traces(hole=.4, hoverinfo="label+percent+name",
textfont_size=16)


fig.update_layout(
    title_text="Gender and Churn Distributions",
    # Add annotations in the center of the donut pies.
    annotations=[dict(text='Gender', x=0.16, y=0.5, font_size=20,
showarrow=False),
                 dict(text='Churn', x=0.84, y=0.5, font_size=20,
showarrow=False)])
fig.show()
```



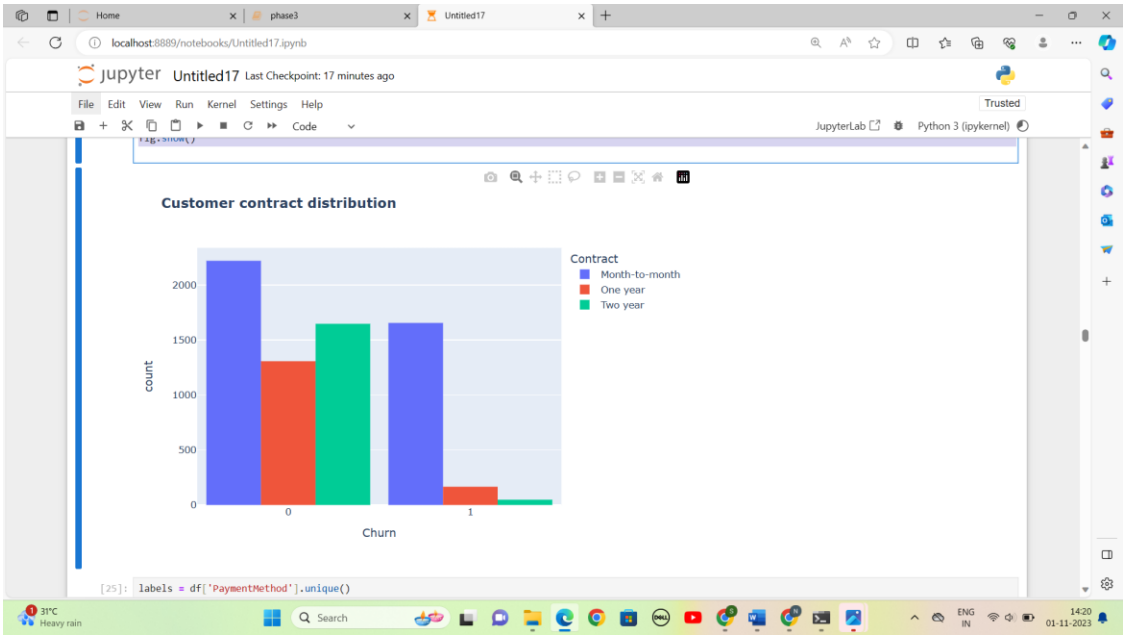

```
plt.figure(figsize=(6, 6))
labels =["Churn: Yes","Churn:No"]
values = [1869,5163]
labels_gender = ["F","M","F","M"]
sizes_gender = [939,930 , 2544,2619]
colors = ['#ff6666', '#66b3ff']
colors_gender = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']
explode = (0.3,0.3)
explode_gender = (0.1,0.1,0.1,0.1)
textprops = {"fontsize":15}
#Plot
plt.pie(values,
labels=labels,autopct='%1.1f%%',pctdistance=1.08,
```

```
labeldistance=0.8,colors=colors, startangle=90,frame=True,
explode=explode,radius=10, textprops =textprops, counterclock =
True, )

plt.pie(sizes_gender,labels=labels_gender,colors=colors_gender,s
tartangle=90, explode=explode_gender,radius=7, textprops
=textprops, counterclock = True, )

#Draw circle

centre_circle = plt.Circle((0,0),5,color='black',
fc='white',linewidth=0)

fig = plt.gcf()

fig.gca().add_artist(centre_circle)


plt.title('Churn Distribution w.r.t Gender: Male(M), Female(F)',
fontsize=15, y=1.1)


# show plot


plt.axis('equal')

plt.tight_layout()

plt.show()
```



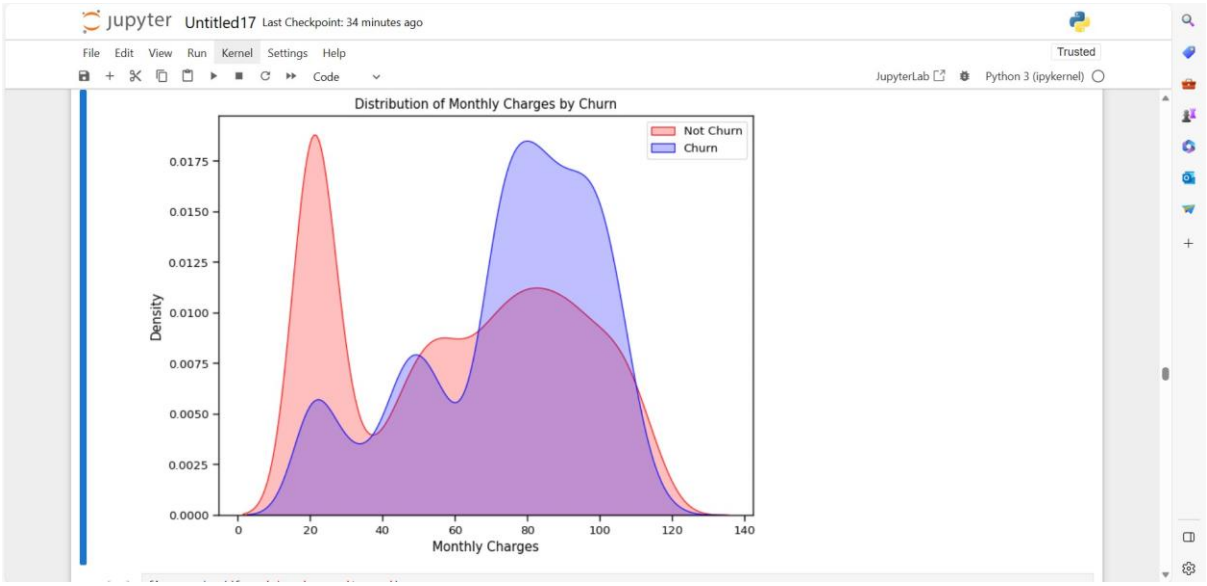

```
fig = px.histogram(df, x="Churn", color="Contract",
barmode="group", title="<b>Customer contract distribution<b>")
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```

```
labels = df['PaymentMethod'].unique()
values = df['PaymentMethod'].value_counts()
```




```
sns.set_context("paper", font_scale=1.1)
plt.figure(figsize=(8, 6))

# Line plot for customers who do not churn (Churn = 0)
sns.kdeplot(df.MonthlyCharges[df['Churn'] == 0], color='red',
label='Not Churn', shade=True)

# Line plot for customers who churn (Churn = 1)
sns.kdeplot(df.MonthlyCharges[df['Churn'] == 1], color='blue',
label='Churn', shade=True)

plt.xlabel('Monthly Charges')
plt.ylabel('Density')
plt.title('Distribution of Monthly Charges by Churn')
```

```
plt.legend()

plt.show()
```

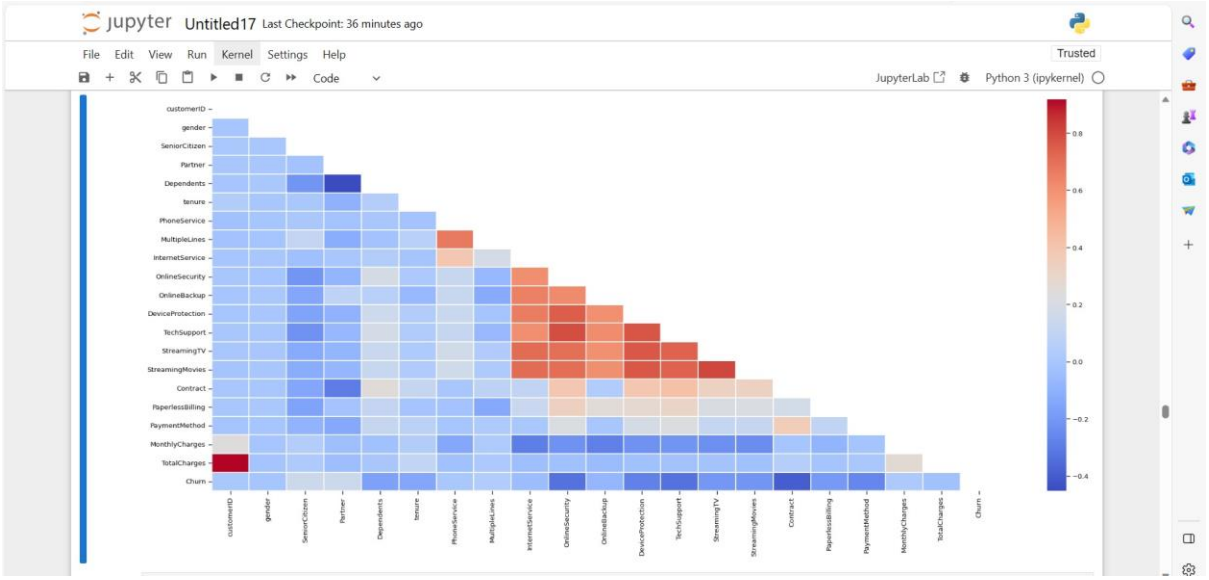



```
fig = px.box(df, x='Churn', y = 'tenure')

# Update yaxis properties
fig.update_yaxes(title_text='Tenure (Months)', row=1, col=1)
# Update xaxis properties
fig.update_xaxes(title_text='Churn', row=1, col=1)

# Update size and title
fig.update_layout(autosize=True, width=750, height=600,
    title_font=dict(size=25, family='Courier'),
```

```
    title='<b>Tenure vs Churn</b>',
)

fig.show()
```


```
# Correlation between all variables

plt.figure(figsize=(25, 10))

corr = df.apply(lambda x: pd.factorize(x)[0]).corr()

mask = np.triu(np.ones_like(corr, dtype=bool))
```

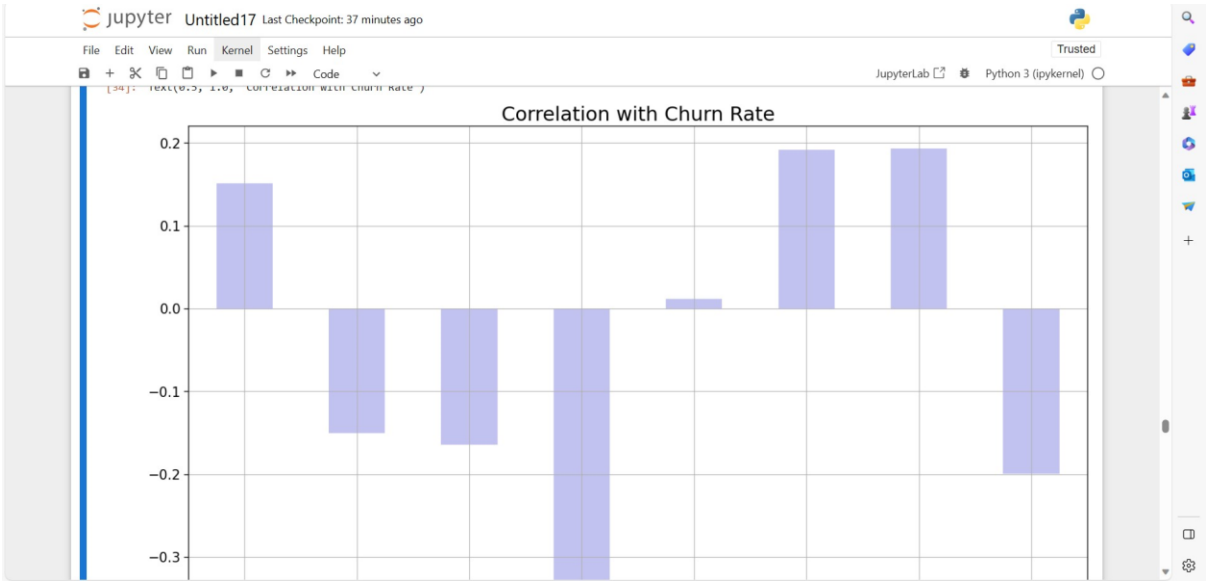
```
ax = sns.heatmap(corr, mask=mask, xticklabels=corr.columns,
yticklabels=corr.columns, annot=True, linewidths=.2,
cmap='coolwarm')
```




```
# Correlation between churn and selected boolean and numeric
variables

plt.figure

ds_corr = df[['SeniorCitizen', 'Partner', 'Dependents',

        'tenure', 'PhoneService', 'PaperlessBilling',

         'MonthlyCharges', 'TotalCharges']]


correlations = ds_corr.corrwith(df.Churn)

correlations = correlations[correlations!=1]

correlations.plot.bar(

        figsize = (18, 10),
```

```
        fontsize = 15,
        color = '#c2c2f0',
        rot = 45, grid = True)
```

```
plt.title('Correlation with Churn Rate',
horizontalalignment="center", fontstyle = "normal", fontsize = "22",
fontfamily = "sans")
```

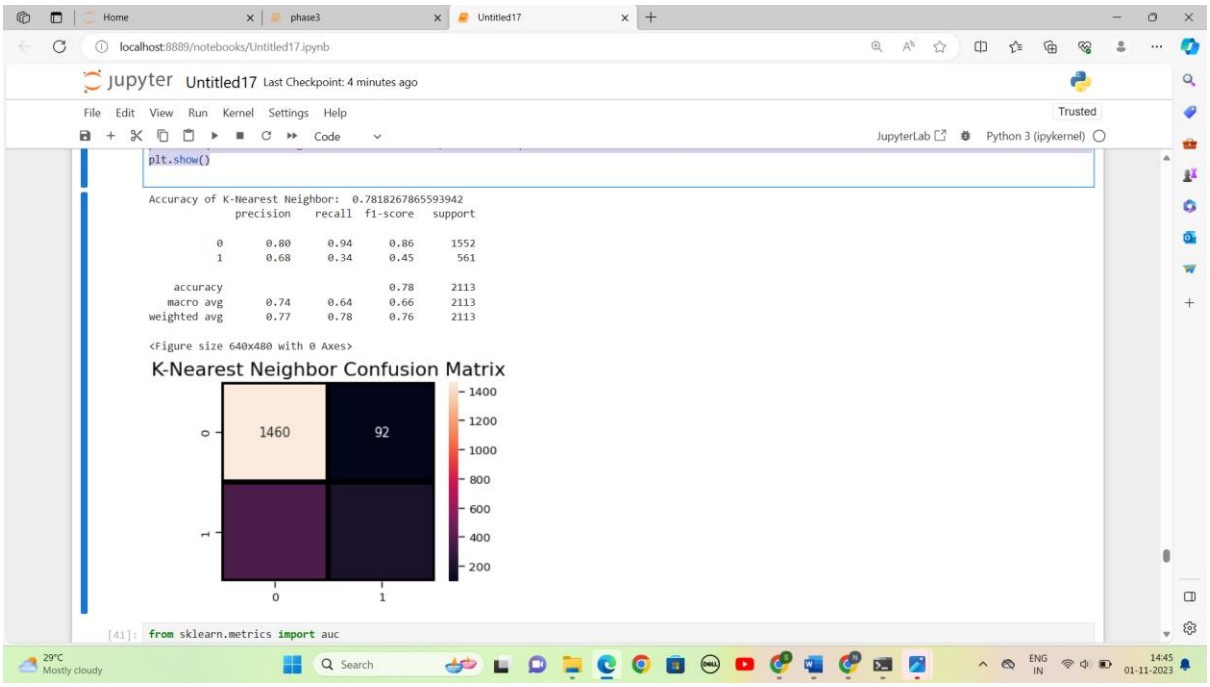



```
knn_model = KNeighborsClassifier(n_neighbors = 10)
```

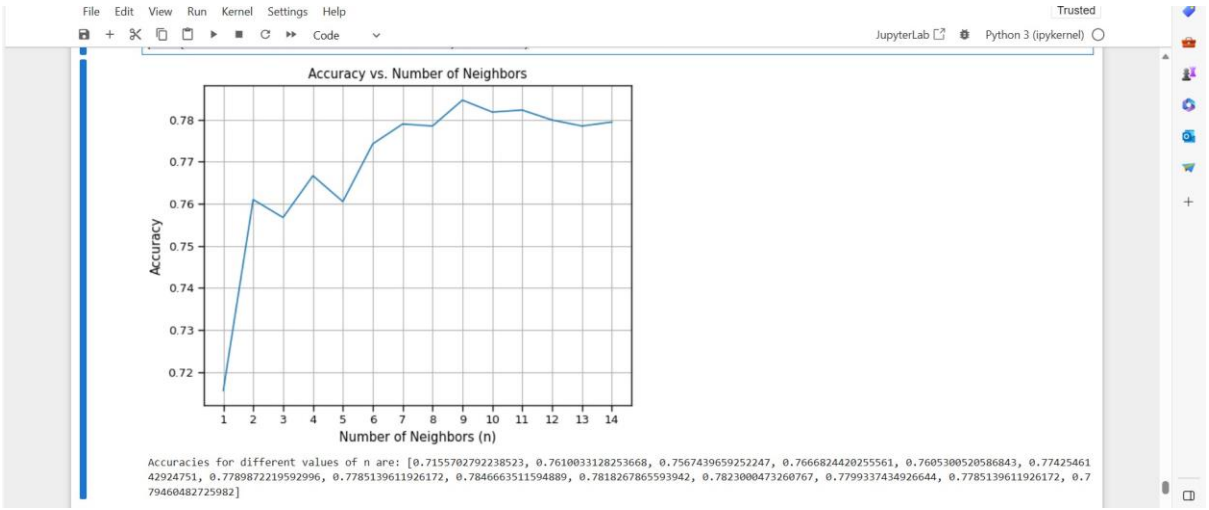
```
knn_model.fit(X_train,y_train)
```

```
# Evaluate model
accuracy_knn = knn_model.score(X_test,y_test)
print("Accuracy of K-Nearest Neighbor: ", accuracy_knn)
```

```
# Classification report
knn_prediction = knn_model.predict(X_test)
print(classification_report(y_test, knn_prediction))

plt.figure(14)
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, knn_prediction),
            annot=True, fmt = "d", linecolor="k", linewidths=3)

plt.title("K-Nearest Neighbor Confusion Matrix", fontsize=16)
plt.show()
```

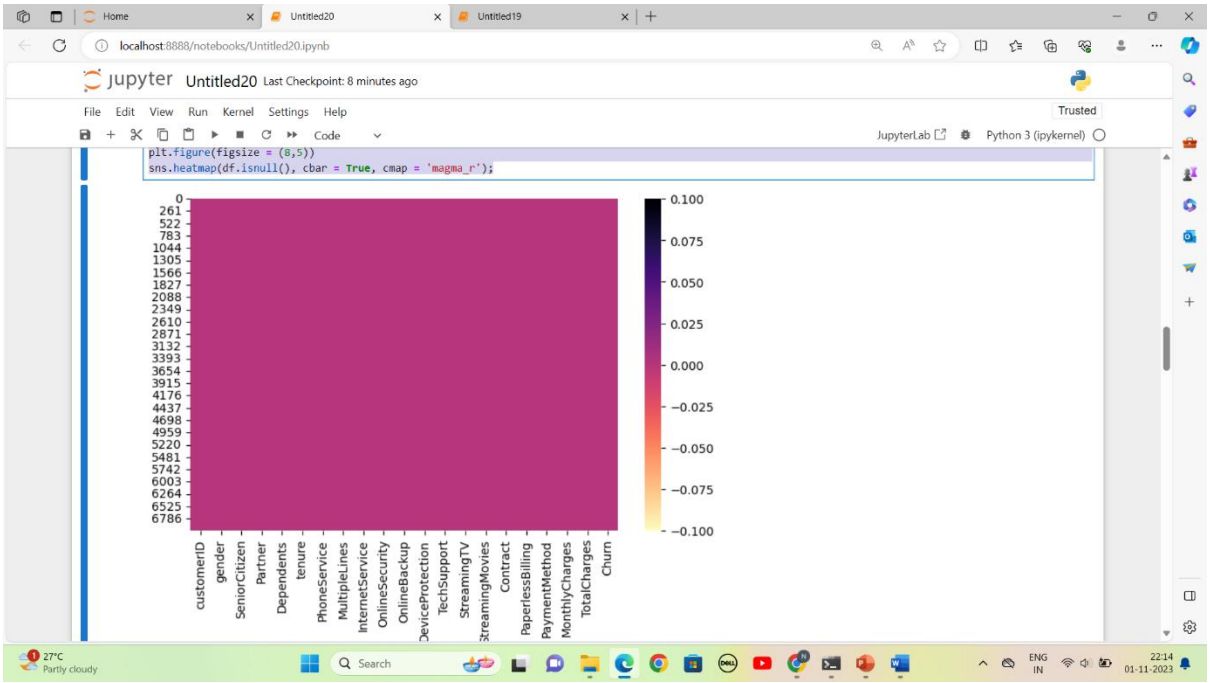


```
a_index = list(range(1, 15))
accuracies = []  # Initialize an empty list to store accuracy values

for i in a_index:
    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train, y_train)
    prediction = model.predict(X_test)
    accuracy = metrics.accuracy_score(y_test, prediction)
    accuracies.append(accuracy)

# Plot the accuracy values
import matplotlib.pyplot as plt

plt.plot(a_index, accuracies)
plt.xticks(a_index)
plt.xlabel('Number of Neighbors (n)')
plt.ylabel('Accuracy')
plt.title('Accuracy vs. Number of Neighbors')
plt.grid(True)
plt.show()

print('Accuracies for different values of n are:', accuracies)
```



visualizing the missing values in 'TotalCharges'

```
plt.figure(figsize = (8,5))
sns.heatmap(df.isnull(), cbar = True, cmap = 'magma_r');
```



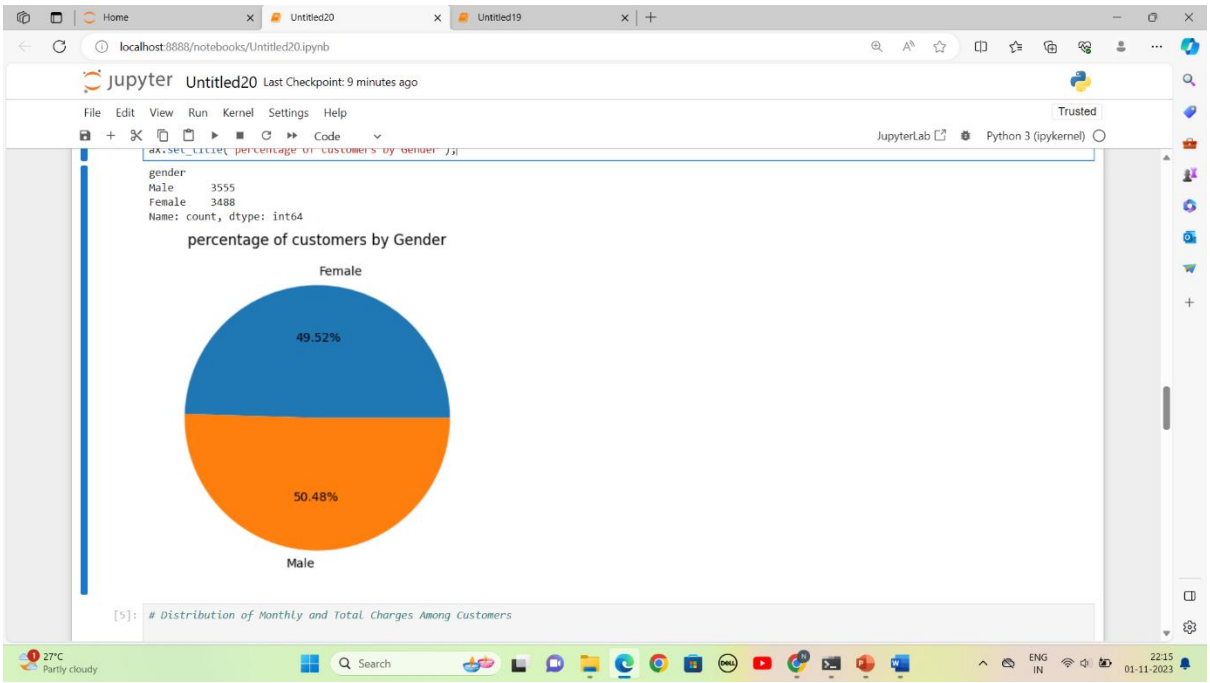
#Data Analysis

#Univariate Analysis

percentage of customers by Gender

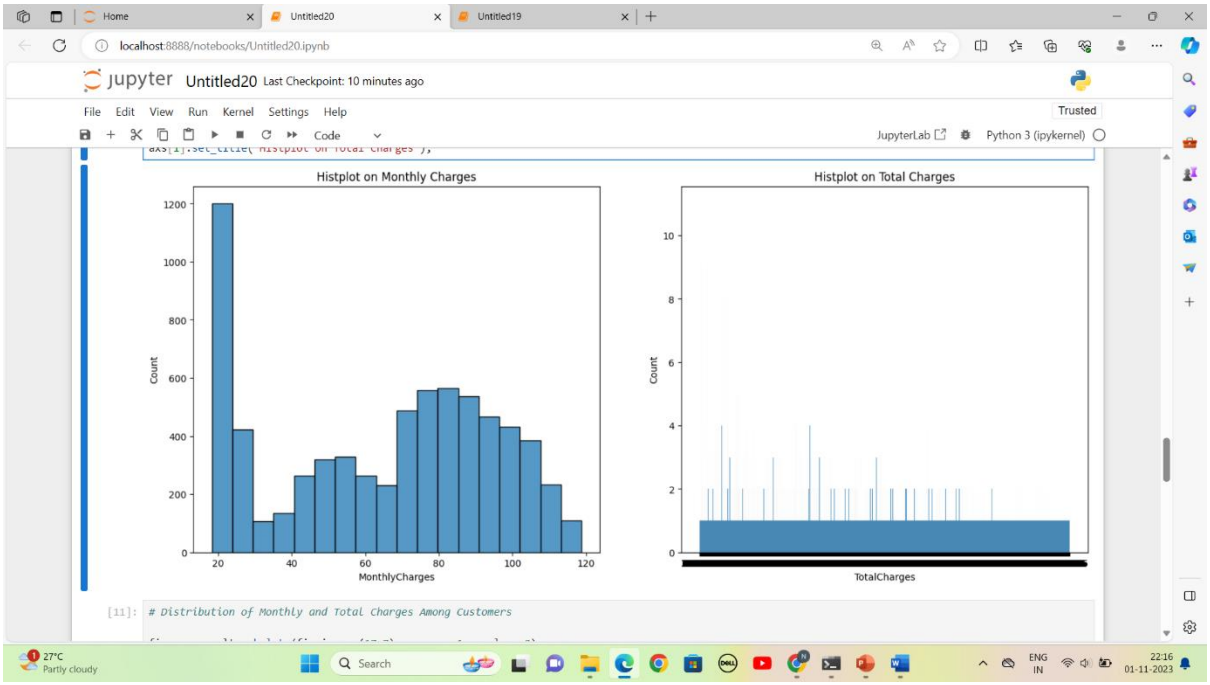
```
print(df['gender'].value_counts())
```

```
fig, ax = plt.subplots(figsize = (5,5))
count = Counter(df['gender'])
ax.pie(count.values(), labels = count.keys(), autopct = lambda p:f'{p:.2f}%')
ax.set_title('percentage of customers by Gender');
```

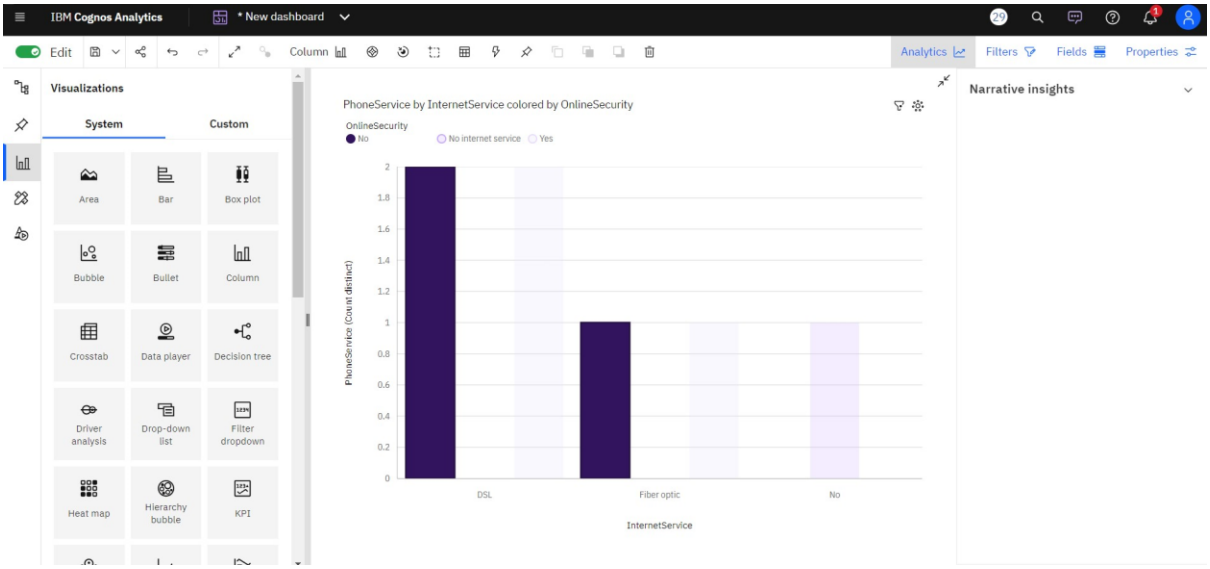


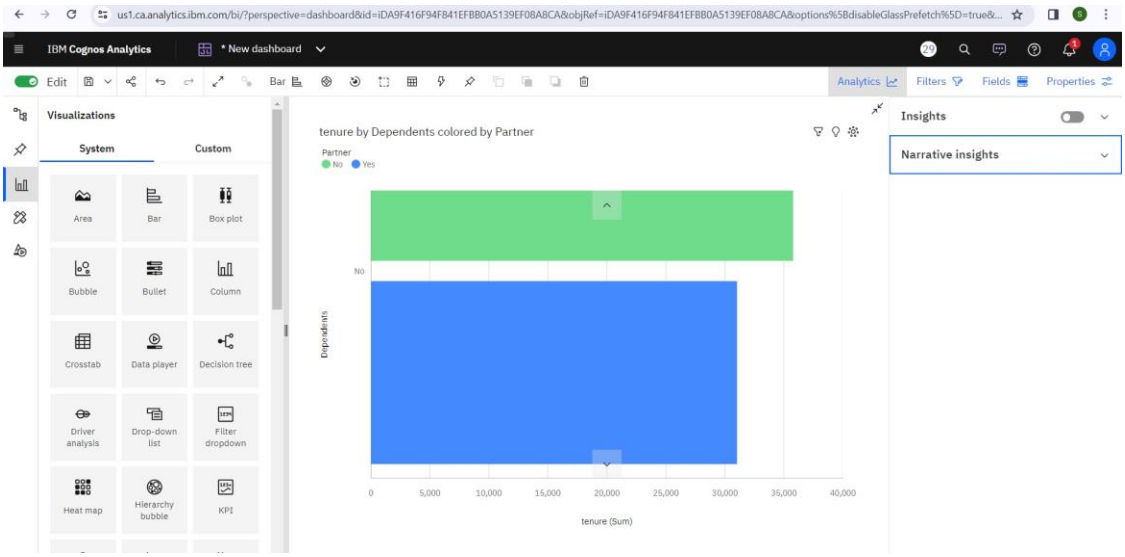
Distribution of Monthly and Total Charges Among Customers

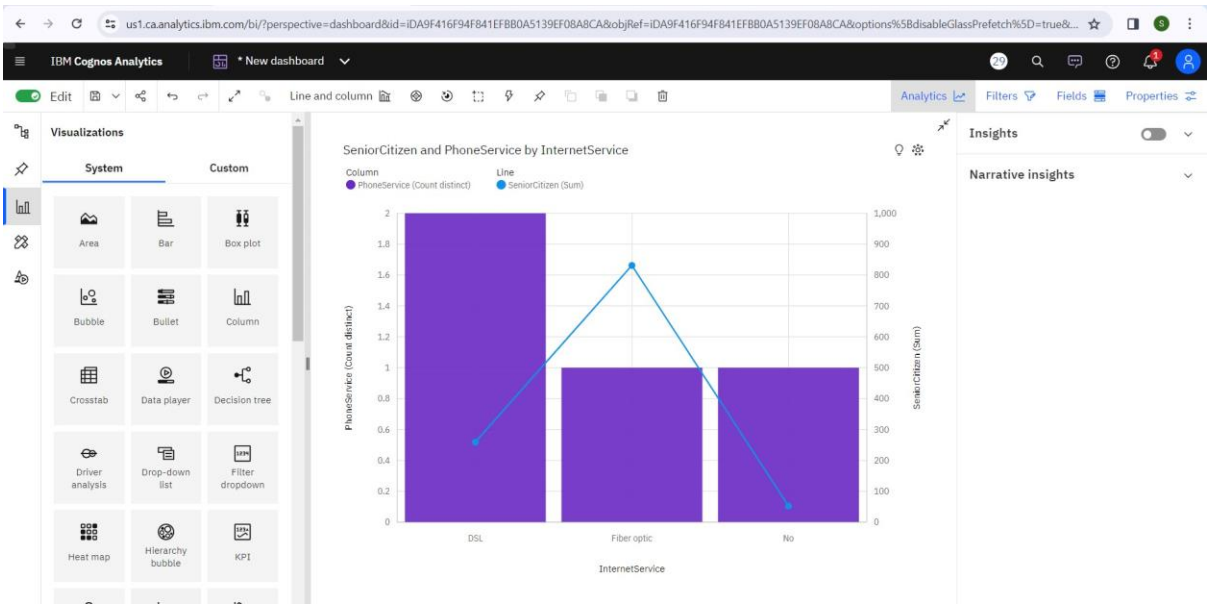
```
fig, axs = plt.subplots(figsize = (17,7), nrows = 1, ncols = 2)

sns.histplot(x = 'MonthlyCharges', data = df, ax = axs[0])
axs[0].set_title('Histplot on Monthly Charges')

sns.histplot(x = 'TotalCharges', data = df, ax = axs[1])
axs[1].set_title('Histplot on Total Charges');
```

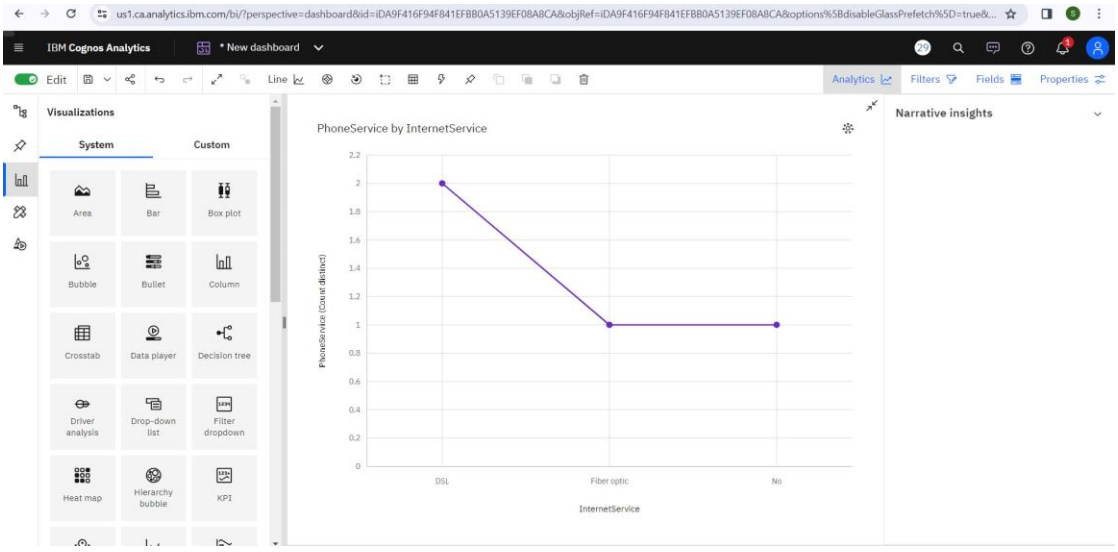


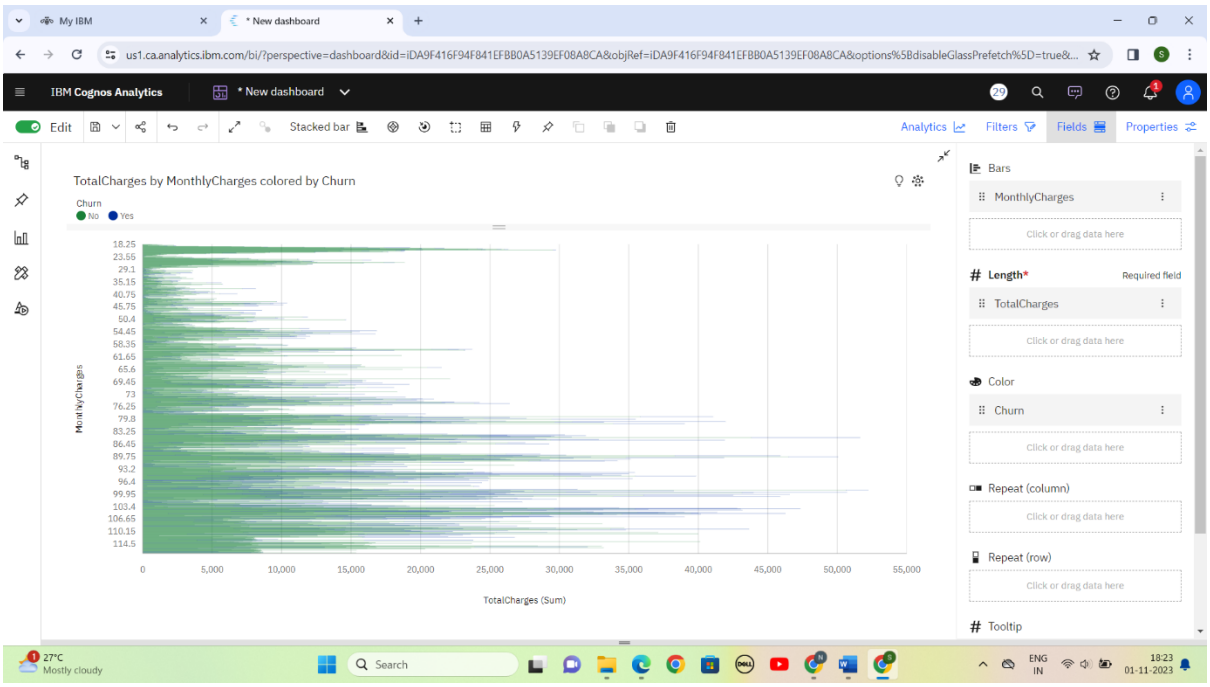
Create interactive dashboards and reports in IBM Cognos to visualize churn patterns, retention rates, and key factors influencing churn.


IBM Cognos Analytics
* New dashboard
Edit
Column
Analytics
Filters
Fields
Properties
Visualizations
System
Custom
Area
Bar
Box plot
Bubble
Bullet
Column
Crosstab
Data player
Decision tree
Driver analysis
Drop-down list
Filter dropdown
Heat map
Hierarchy bubble
KPI
PhoneService by InternetService colored by OnlineSecurity
OnlineSecurity
No
No internet service
Yes
PhoneService (Count distinct)
DSL
Fiber optic
No
InternetService
Narrative insights

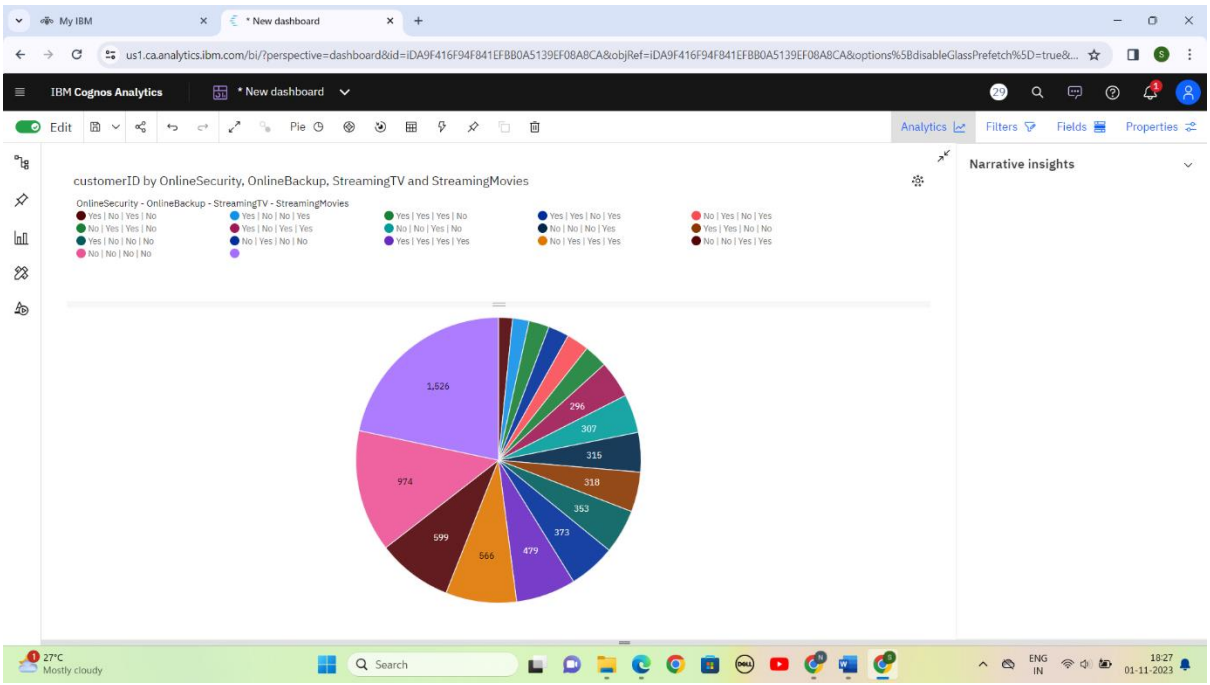

us1.ca.analytics.ibm.com/bi/?perspective=dashboard&id=iDA9F416F94F841EFBB0A5139EF08A8CA&objRef=iDA9F416F94F841EFBB0A5139EF08A8CA&options%5BdisableGlassPrefetch%5D=true&...
IBM Cognos Analytics
* New dashboard
Edit
Bar
Analytics
Filters
Fields
Properties
Visualizations
System
Custom
Area
Bar
Box plot
Bubble
Bullet
Column
Crosstab
Data player
Decision tree
Driver analysis
Drop-down list
Filter dropdown
Heat map
Hierarchy bubble
KPI
tenure by Dependents colored by Partner
Partner
No
Yes
Dependents
No
0
5,000
10,000
15,000
20,000
25,000
30,000
35,000
40,000
tenure (Sum)
Insights
Narrative insights

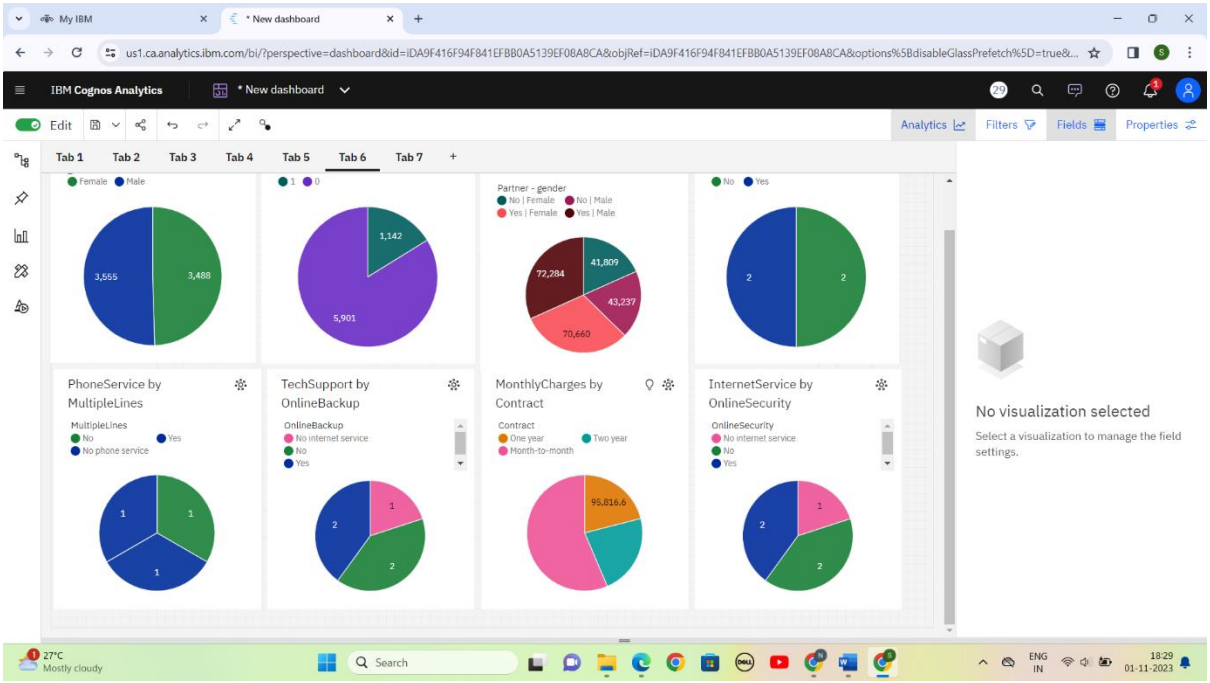

IBM Cognos Analytics
* New dashboard
Edit
Line and column
Analytics
Filters
Fields
Properties
Visualizations
System
Custom
Area
Bar
Box plot
Bubble
Bullet
Column
Crosstab
Data player
Decision tree
Driver analysis
Drop-down list
Filter dropdown
Heat map
Hierarchy bubble
KPI
SeniorCitizen and PhoneService by InternetService
Column
PhoneService (Count distinct)
Line
SeniorCitizen (Sum)
PhoneService (Count distinct)
SeniorCitizen (Sum)
DSL
Fiber optic
No
InternetService
Insights
Narrative insights

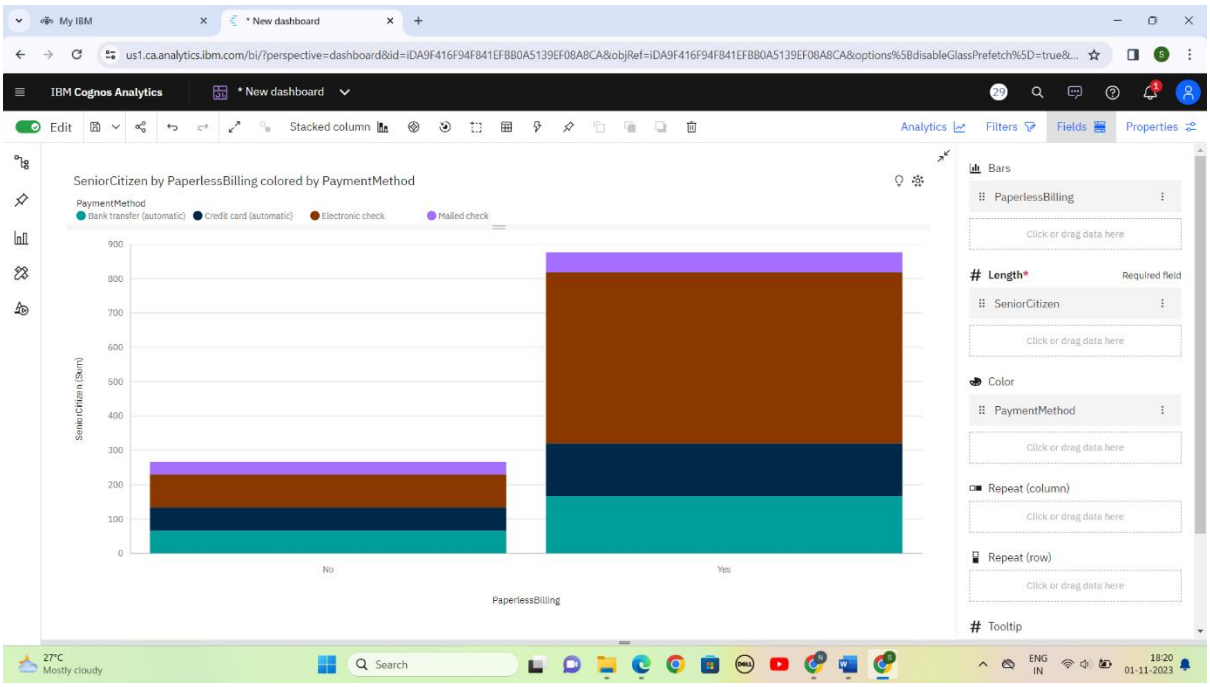

IBM Cognos Analytics
* New dashboard
Edit
Line
Analytics
Filters
Fields
Properties
Visualizations
System
Custom
Area
Bar
Box plot
Bubble
Bullet
Column
Crosstab
Data player
Decision tree
Driver analysis
Drop-down list
Filter dropdown
Heat map
Hierarchy bubble
KPI
PhoneService by InternetService
PhoneService (Count distinct)
DSL
Fiber optic
No
InternetService
Narrative insights

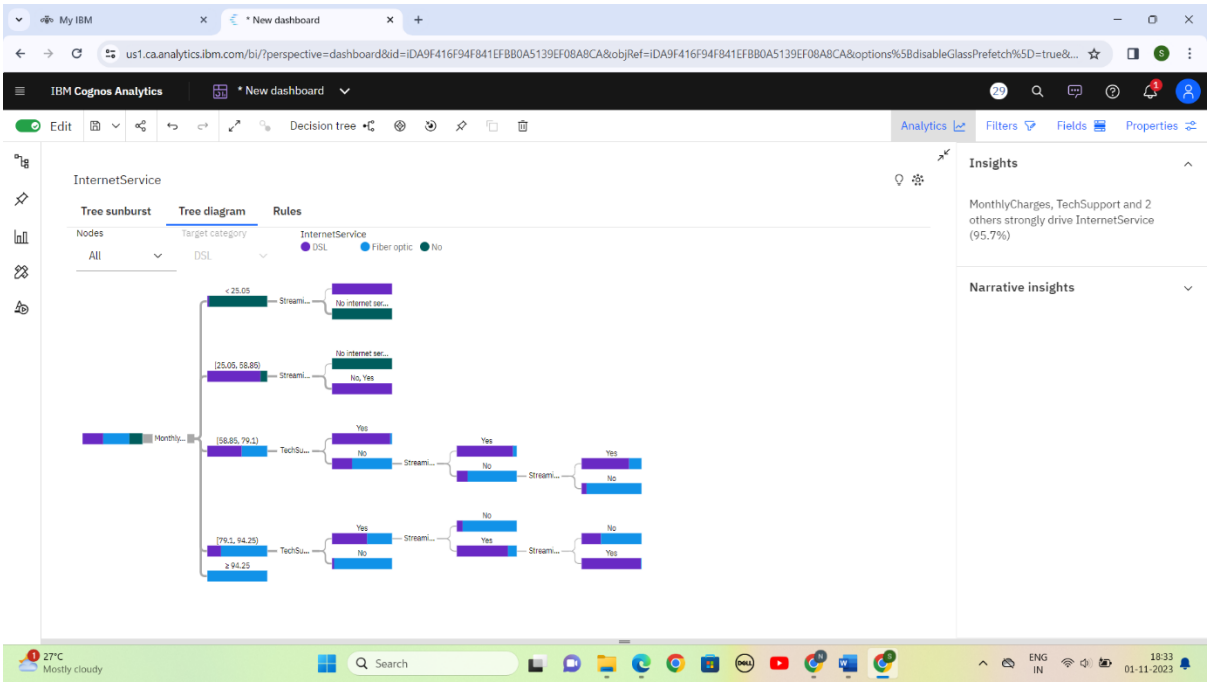

My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Stacked bar
Analytics
Filters
Fields
Properties
TotalCharges by MonthlyCharges colored by Churn
Churn
No
Yes
MonthlyCharges
TotalCharges (Sum)
Bars
MonthlyCharges
Length*
Required field
TotalCharges
Color
Churn
Repeat (column)
Repeat (row)
Tooltip
Search


My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Area
Analytics
Filters
Fields
Properties
TotalCharges by MonthlyCharges colored by SeniorCitizen
SeniorCitizen
0
1
TotalCharges (Sum)
MonthlyCharges
x-axis*
Required field
MonthlyCharges
Color
SeniorCitizen
y-axis*
Required field
TotalCharges
Repeat (column)
Repeat (row)
Tooltip
Search


My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Pie
Analytics
Filters
Fields
Properties
customerID by OnlineSecurity, OnlineBackup, StreamingTV and StreamingMovies
OnlineSecurity - OnlineBackup - StreamingTV - StreamingMovies
Narrative insights
1,526
296
307
315
318
353
373
479
566
599
974
Search


My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Analytics
Filters
Fields
Properties
Tab 1
Tab 2
Tab 3
Tab 4
Tab 5
Tab 6
Tab 7
Female
Male
3,555
3,488
1
0
1,142
5,901
Partner - gender
No | Female
No | Male
Yes | Female
Yes | Male
72,284
41,809
43,237
70,660
No
Yes
2
2
PhoneService by MultipleLines
MultipleLines
No
Yes
No phone service
1
1
1
TechSupport by OnlineBackup
OnlineBackup
No internet service
No
Yes
2
1
2
MonthlyCharges by Contract
Contract
One year
Two year
Month-to-month
95,816.6
InternetService by OnlineSecurity
OnlineSecurity
No internet service
No
Yes
2
1
2
No visualization selected
Select a visualization to manage the field settings.
Search


My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Stacked column
Analytics
Filters
Fields
Properties
SeniorCitizen by PaperlessBilling colored by PaymentMethod
PaymentMethod
Bank transfer (automatic)
Credit card (automatic)
Electronic check
Mailed check
SeniorCitizen (Sum)
PaperlessBilling
No
Yes
Bars
PaperlessBilling
Length*
Required field
SeniorCitizen
Color
PaymentMethod
Repeat (column)
Repeat (row)
Tooltip
Search


My IBM
* New dashboard
IBM Cognos Analytics
* New dashboard
Edit
Decision tree
Analytics
Filters
Fields
Properties
InternetService
Tree sunburst
Tree diagram
Rules
Nodes
All
Target category
DSL
InternetService
DSL
Fiber optic
No
Insights
MonthlyCharges, TechSupport and 2 others strongly drive InternetService (95.7%)
Narrative insights
Search

Insights and prediction model to help businesses reduce customer churn.

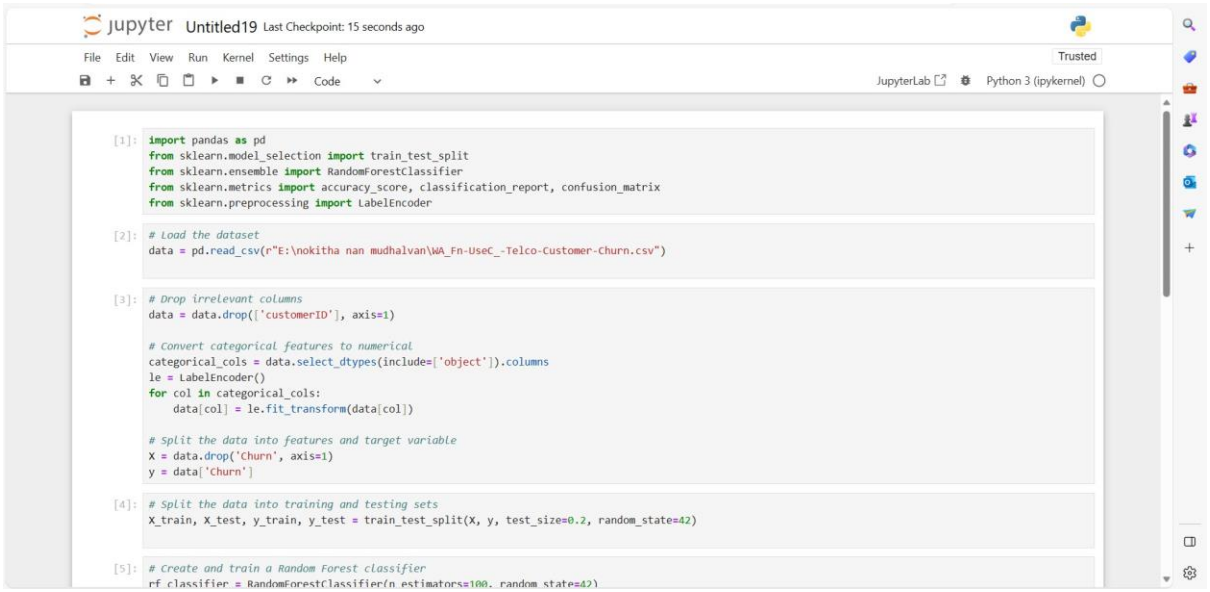
1. Import the necessary libraries.

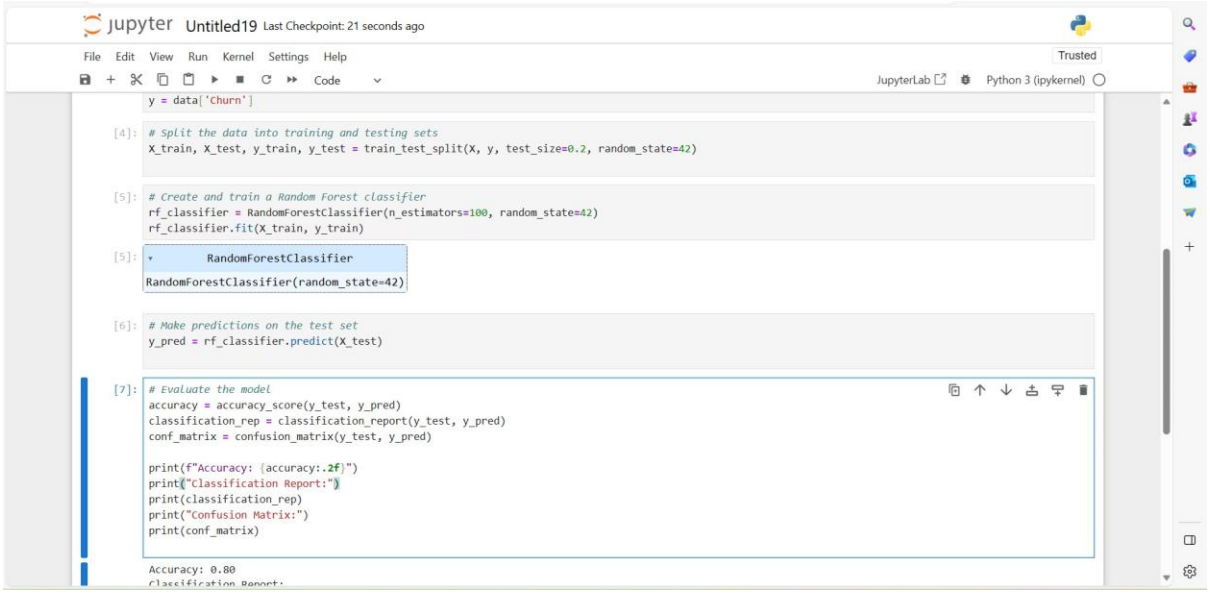
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder

# Load the dataset
data = pd.read_csv(r"E:\nokitha nan mudhalvan\WA_Fn-UseC_-Telco-Customer-Churn.csv")
# Drop irrelevant columns
data = data.drop(['customerID'], axis=1)

# Convert categorical features to numerical
categorical_cols = data.select_dtypes(include=['object']).columns
le = LabelEncoder()
for col in categorical_cols:
    data[col] = le.fit_transform(data[col])
```

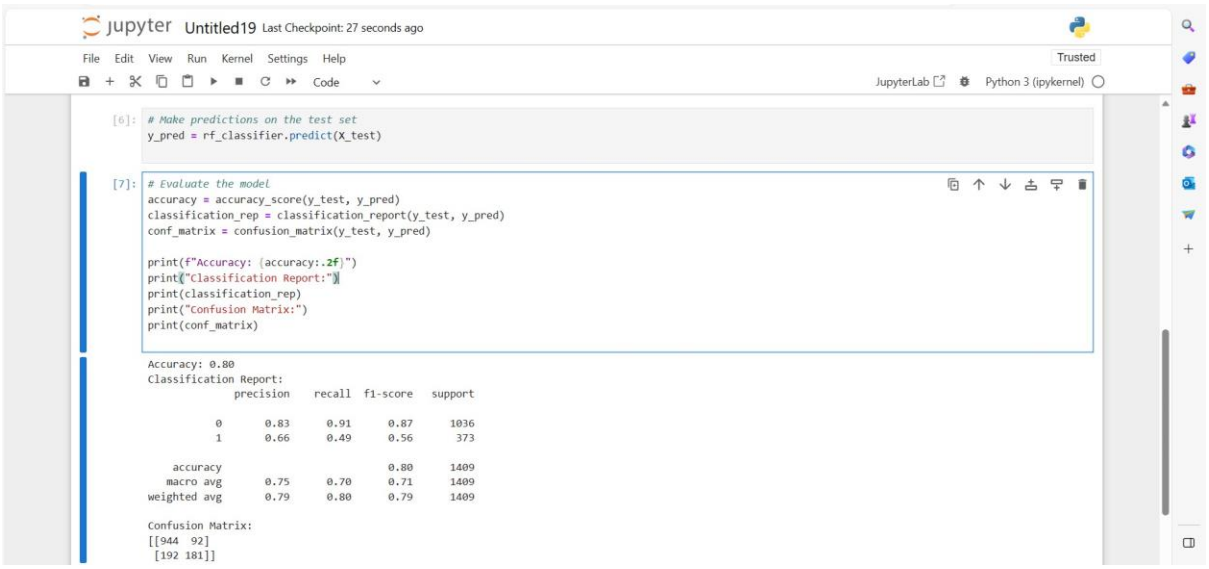
```
# Split the data into features and target variable
X = data.drop('Churn', axis=1)
y = data['Churn']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and train a Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)
rf_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(classification_rep)
```

print("Confusion Matrix:")

print(conf_matrix)







CONCLUSION:

In conclusion, the telecom customer churn prediction model plays a pivotal role in today's highly competitive telecommunications industry. By harnessing the power of data and machine learning, telecom companies can proactively identify and retain valuable customers while reducing customer attrition. This model allows businesses to make data-driven decisions, deploy targeted retention strategies, and allocate resources more effectively. It not only enhances customer satisfaction but also contributes to the long-term profitability and sustainability of telecom providers. By continuously improving and fine-tuning such models, telecom companies can stay ahead of the curve, adapt to changing customer dynamics, and build stronger, lasting relationships with their customer base.