PHASE-3

71772117301 – DEEPIKAA.V

COLLEGE CODE : 7177

CUSTOMER CHURN PREDICTION USING DATA ANALYTICS WITH COGNOS

INTRODUCTION

The project at hand revolves around harnessing the capabilities of IBM Cognos to address a critical business challenge – predicting customer churn and enhancing customer retention strategies. The overarching goal is to empower businesses with the means to reduce customer attrition by gaining deep insights into the underlying patterns and drivers of customer departures.

DATASET DETAILS:

Data set from kaggle:

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

Content

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

The data set includes information about:

- 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

LIBRARIES

```
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

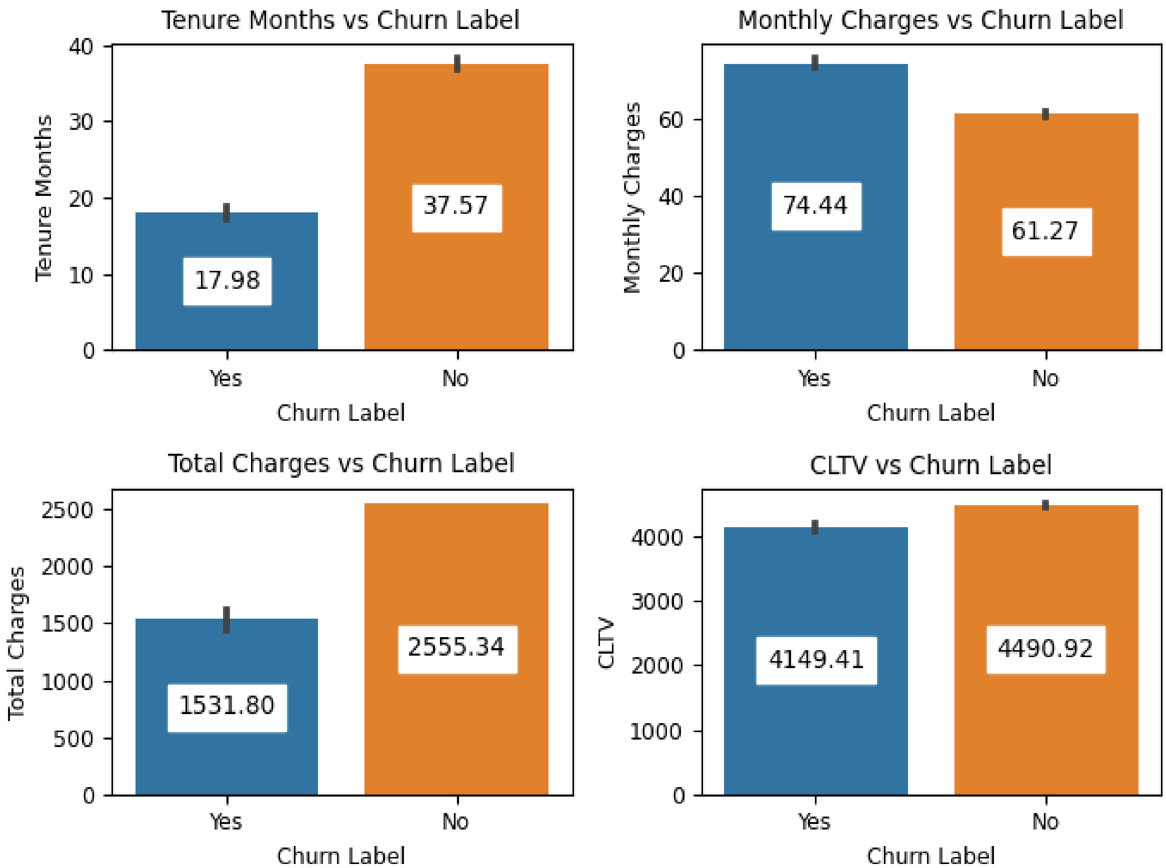
ANALYSIS OBJECTIVES:

1. **Descriptive Analysis**:

    - Understand the distribution of customers based on 'SeniorCitizen', 'gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService', and 'OnlineSecurity'.

Calculate the average 'tenure' for customers in different categories of the aforementioned fields and see if certain categories have notably shorter or longer tenures.

```
styled_df = (
df.describe()
.drop("count", axis=0)
.style.background_gradient(axis=0, cmap="magma")
.set_properties(**{"text-align": "center"})
.set_table_styles([{"selector": "th", "props": [("background-color", "k")]}])
.set_caption("Summary Statistics")
)styled_df
```



2. **Data Cleaning & Preparation**:

- Handle missing values, if any, in the data fields.
- Convert categorical data fields like 'gender', 'Partner', etc., into a format suitable for analytical modeling (e.g., one-hot encoding).

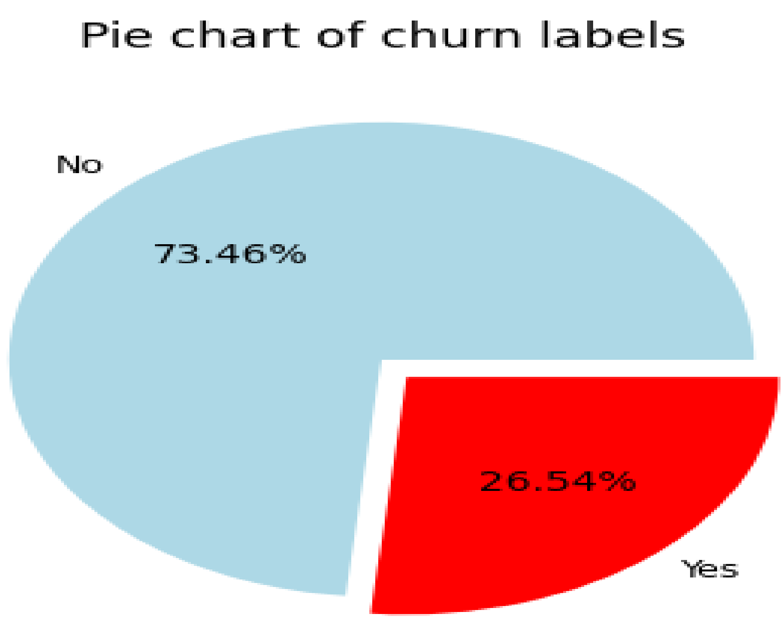
```
df['TotalCharges'] = pd.to_numeric(df.TotalCharges, errors='coerce')

df.isnull().sum()

df[df['tenure'] == 0].index

Out[13]:

Int64Index([488, 753, 936, 1082, 1340, 3331, 3826, 4380, 5218, 6670, 6754], dtype='int64'
```

3. **Churn Distribution Analysis**:

- Analyze the churn rate distribution among different categories within each field (e.g., churn rate among 'SeniorCitizen' vs. non-'SeniorCitizen').

Investigate if certain combinations of fields (e.g., a 'SeniorCitizen' with 'MultipleLines' service) have a higher propensity to churn.

```
reasons = df["Churn Reason"][df["Churn Reason"].notna()]

reasons = reasons.value_counts().to_frame()

reasons.index.name = "Churn Reason"

reasons.columns = ["counts"]

reasons = reasons.assign(percent=lambda x: x / reasons["counts"].sum())

formater = lambda x: f"{x:.2%}"

reasons["percent"] = reasons.percent.apply(formater)

reasons.reset_index(inplace=True)

reasons
```



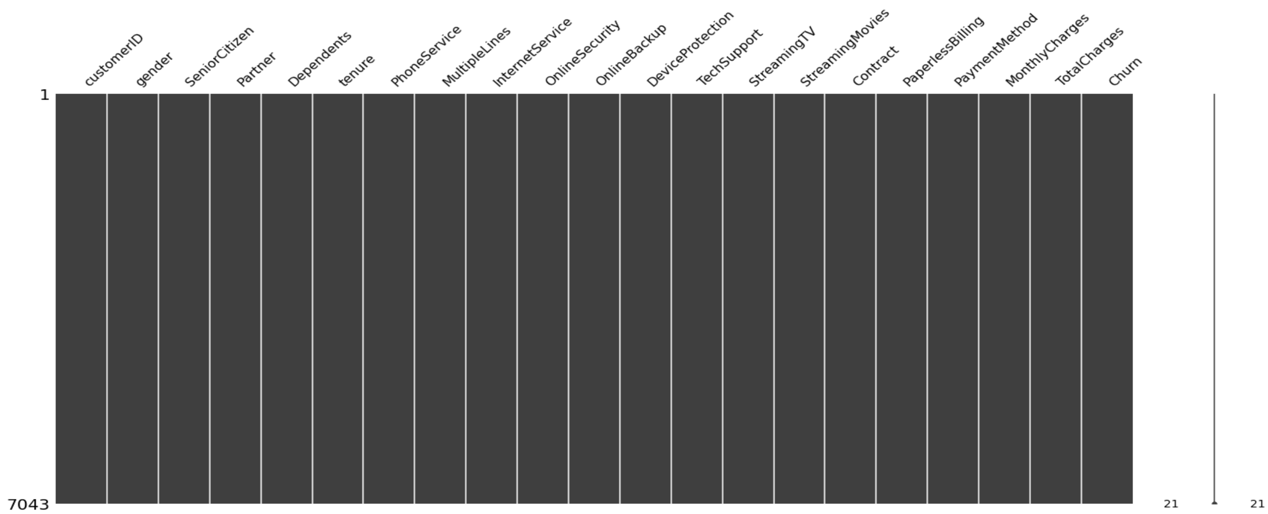
4. **Feature Importance & Correlation Analysis**:

- Identify which fields (or combination of fields) are most indicative or predictive of churn.
- Assess multicollinearity among predictors to ensure the stability and interpretability of the predictive model.

5. Visualize missing values

Visualize missing values as a matrix

msno.matrix(df);



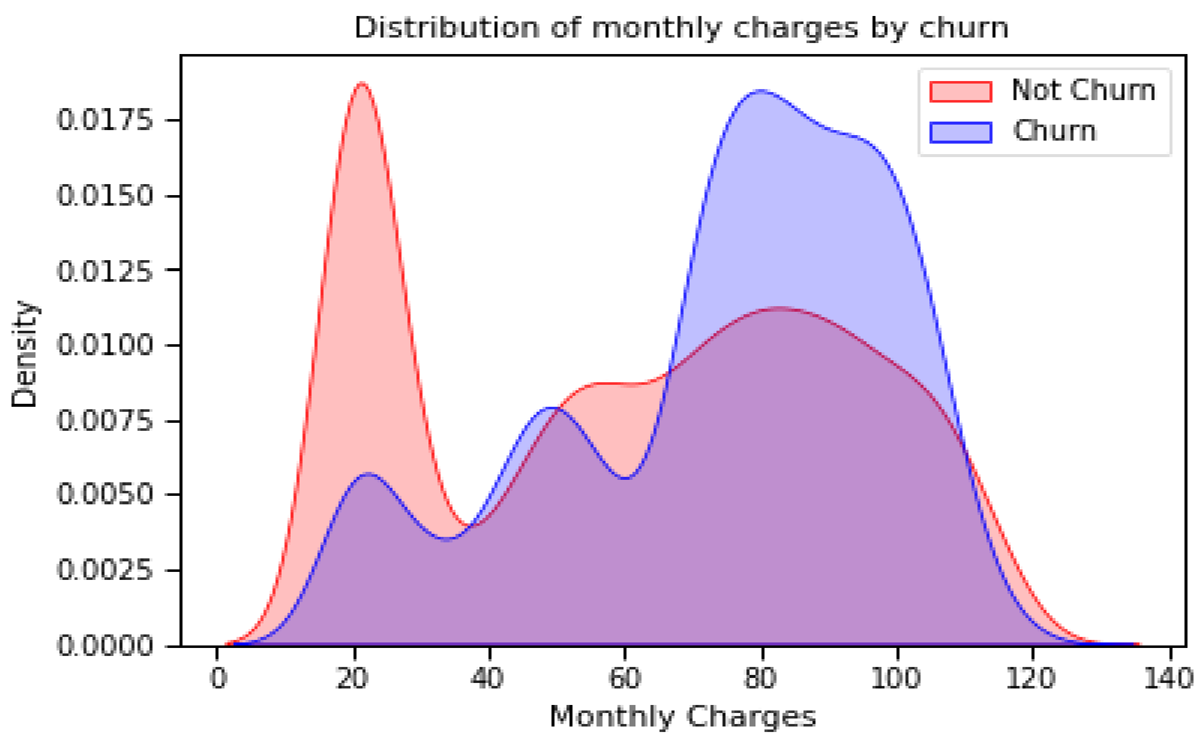
From the above visualisation we can observe that it has no peculiar pattern that stands out. In fact there is no missing data.

6.DATA VISUALIZATION:

In [30]:

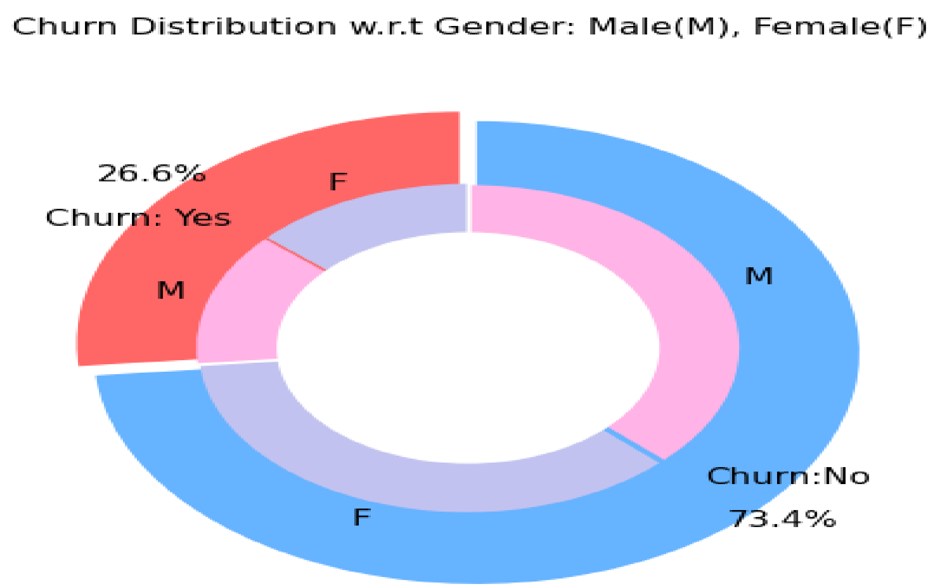
```
fig = go.Figure()

fig.add_trace(go.Bar(

  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],

      ["Female", "Male", "Female", "Male"]],

  y = [965, 992, 219, 240],

name = 'DSL',

))fig.add_trace(go.Bar(

  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],

      ["Female", "Male", "Female", "Male"]],

  y = [889, 910, 664, 633],

  name = 'Fiber optic',
```

```
))
fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [690, 717, 56, 57],
  name = 'No Internet',
))
fig.update_layout(title_text="<b>Churn Distribution w.r.t. Internet Service and Gender</b>")
fig.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)
```

```python
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()
```



Conclusion:

This project amalgamates cutting-edge analytics, innovative strategies, and ethical considerations to address the critical challenge of customer churn. The outcome will empower businesses with actionable insights and personalized retention strategies, ultimately fostering long-term customer loyalty and profitability.