PHASE-5
71772117301– DEEPIKAA V
COLLEGE CODE : 7177
CUSTOMER CHURN PREDICTION USING DATA ANALYTICS WITH COGNOS
PROJECT CODE

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.

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.

CODE:

```
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).

CODE:

```
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'
```

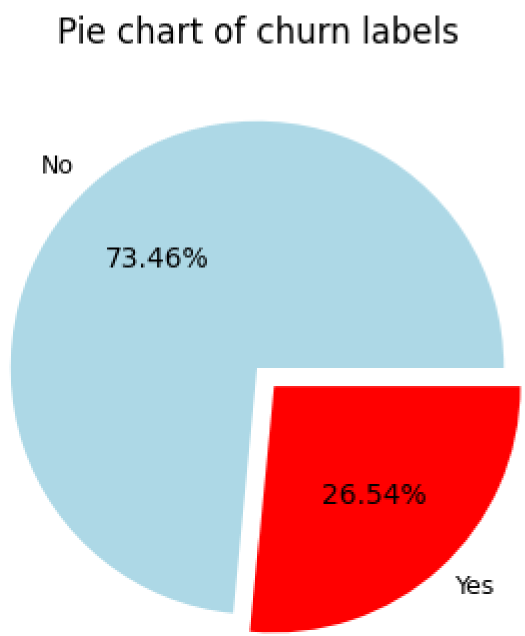
3. **Churn Distribution Analysis**:

• 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.

CODE:

```
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)
```

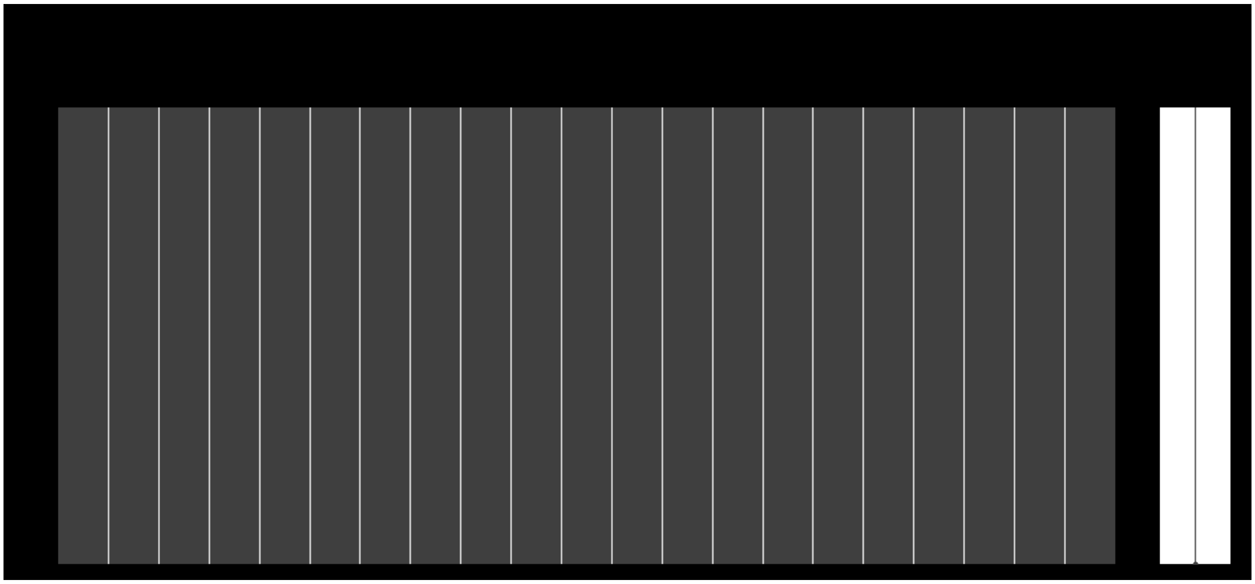


4. Visualize missing values

Visualize missing values as a matrix

CODE:

```
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.

5.DATA VISUALIZATION:

CODE:

```
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)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()
```

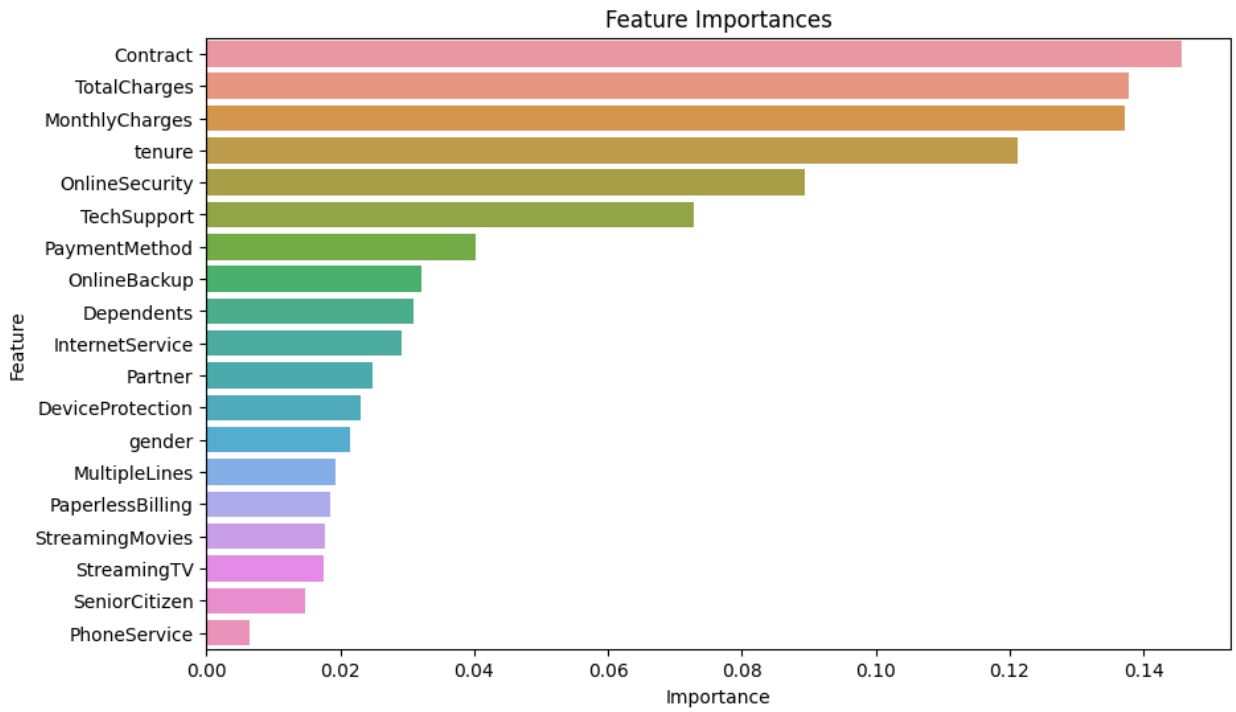
6.Feature Importance Analysis:

Analyzing feature importances and visualizing them using a bar plot.

CODE:

```
In [16]: linkcode
# Analyze feature importances
feature_importances = rf.feature_importances_
features = X.columns
importance_df = pd.DataFrame({'Feature': features, 'Importance':
feature_importances})
importance_df = importance_df.sort_values(by='Importance', ascending=False)
```

```
# Visualize feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importance_df)
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importances')
plt.show()
```



CONCLUSION:

This document contains the details about the project overview and the code for processing the customer churn prediction using data analytics with cognos.