PROJECT TITLE:

Customer Churn Analysis and Prediction

DESCRIPTION:

A data-driven machine learning project that identifies key factors behind customer churn in a telecom company.The project aims to analyse patterns and predict whether a customer is likely to leave the company, helping the business make informed decisions to reduce churn.

FEATURES:

1. Data cleaning, handling missing values, and encoding.
2. Exploratory Data Analysis (EDA) to find insights using visualisations.
3. Feature selection and transformation.
4. Machine learning model building and evaluation.
5. Churn prediction using the best-performing mode

TECHNOLOGIES USED:

1. Python
2. Pandas, NumPy
3. Matplotlib, Seaborn

Tasks 6: Visualizations

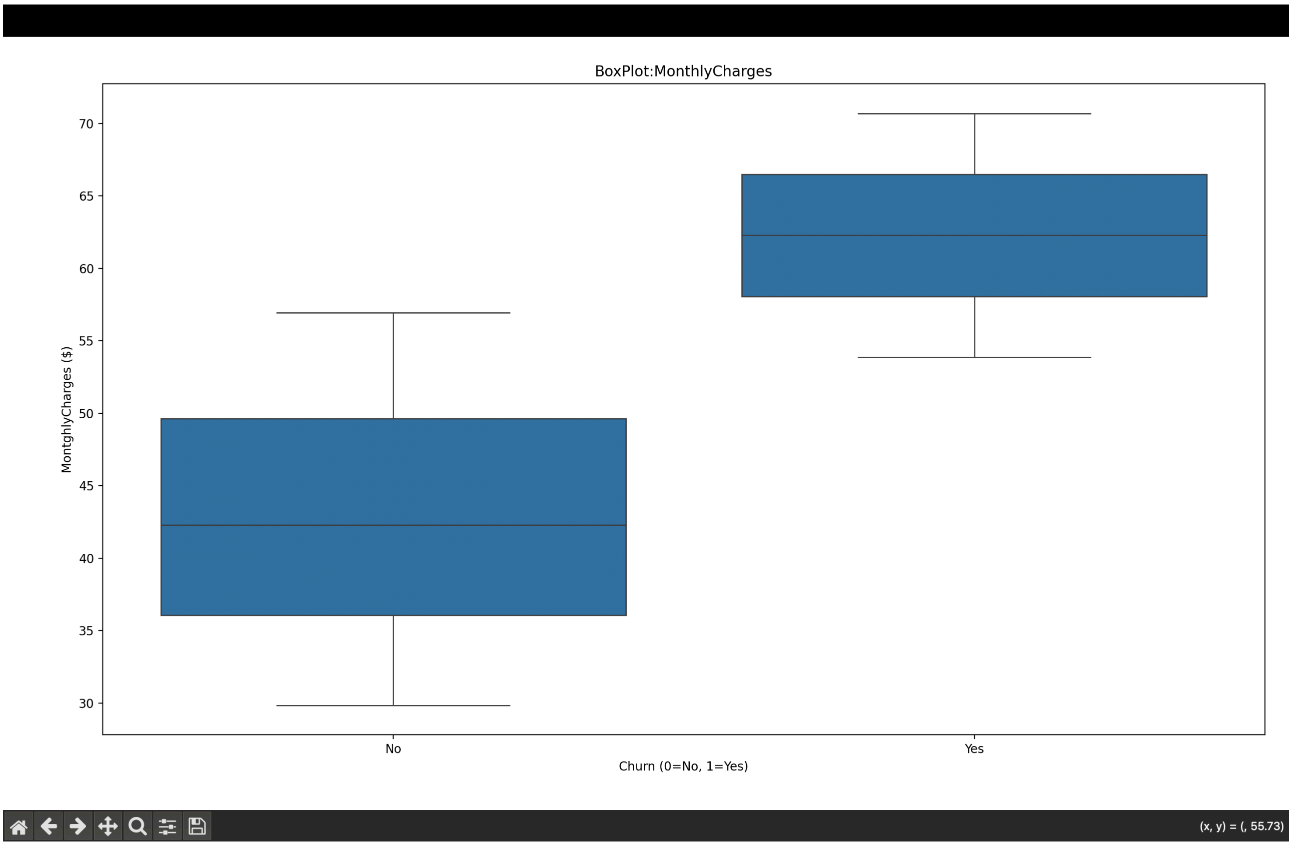
DESCRIPTION:

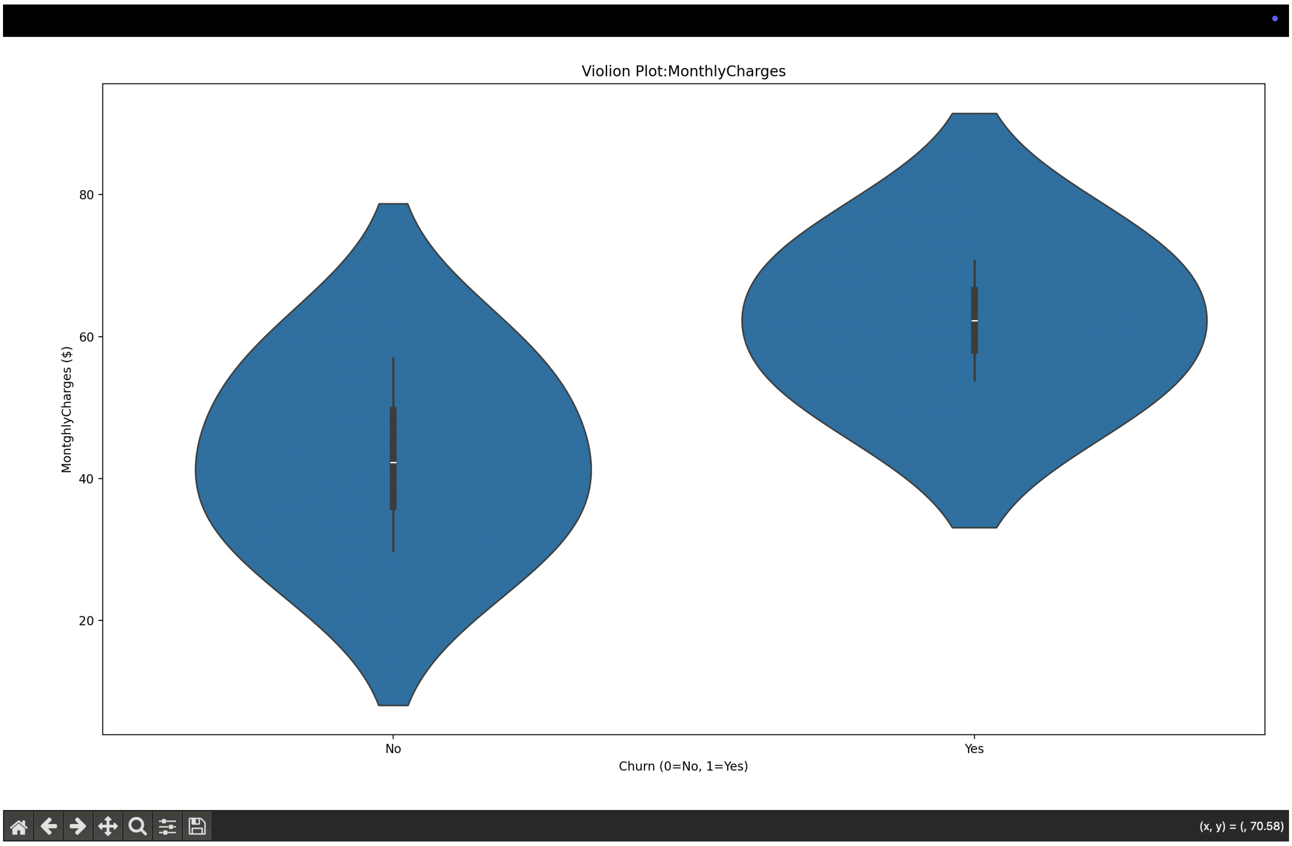
The program analyses customer churn data using visualisations like box plot, violin plot, heatmap, and bar chart to compare churn vs. non-churn customers and explore correlations between numeric features.

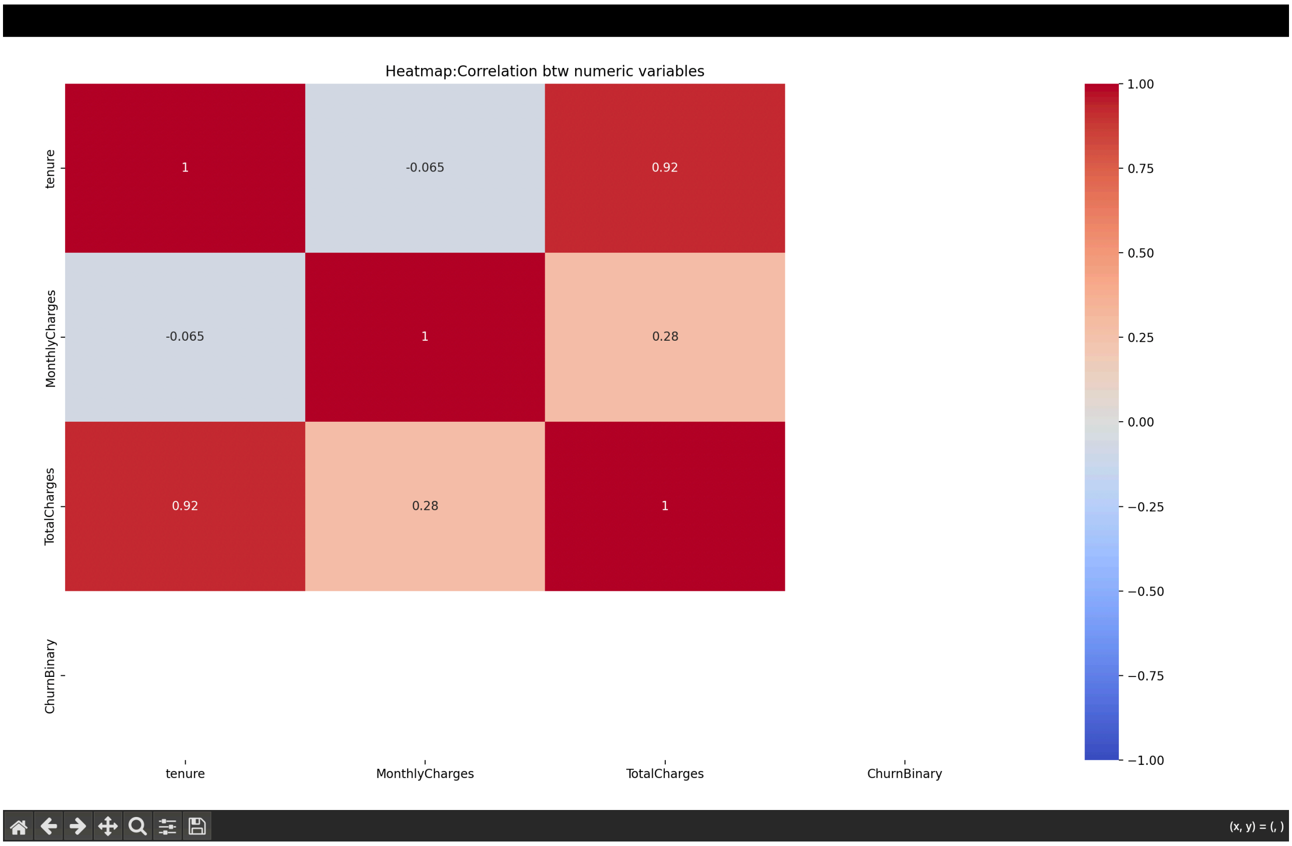
Code Snippet:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df = pd.read_csv("cleaned_customer_churn.csv")
df['ChurnBinary']=df["Churn_Yes"].map({"Yes":1,"No":0})
df['MonthlyCharges']=pd.to_numeric(df['MonthlyCharges'],
errors='coerce')
df['tenure']=pd.to_numeric(df['tenure'], errors='coerce')
df['TotalCharges']=pd.to_numeric(df['TotalCharges'],
errors='coerce')


plt.figure(figsize=(8,6))
sns.boxplot(x="Churn_Yes",y="MonthlyCharges",data=df)
plt.title("BoxPlot:MonthlyCharges")
plt.xlabel("Churn (0=No, 1=Yes)")
plt.ylabel("MontghlyCharges ($)")
plt.xticks([0,1],['No','Yes'])
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,6))
sns.violinplot(x="Churn_Yes",y="MonthlyCharges",data=df)
plt.title("Violion Plot:MonthlyCharges")
plt.xlabel("Churn (0=No, 1=Yes)")
plt.ylabel("MontghlyCharges ($)")
plt.xticks([0,1],['No','Yes'])
plt.tight_layout()
plt.show()

numeric_cols=["tenure","MonthlyCharges","TotalCharges","ChurnB
inary"]
missing_cols=[col for col in numeric_cols if col not in df.columns]
if missing_cols:
    print(f'Error:Missing columns :{missing_cols}')
    exit()
non_numeric_cols=[col for col in numeric_cols if not
pd.api.types.is_numeric_dtype(df[col])]
if non_numeric_cols:
    print(f'Error:Non-Numeric columns :{non_numeric_cols}')
    exit()
plt.figure(figsize=(8,6))
correlation_matrix=df[numeric_cols].corr()
sns.heatmap(correlation_matrix,annot=True,cmap='coolwarm',vmin
=-1,vmax=1)
plt.title("Heatmap:Correlation btw numeric variables")
plt.tight_layout()
plt.show()

plt.figure(figsize=(8,6))
sns.countplot(x="Churn_Yes",data=df)
plt.title("Bar Plot:Count of Customers")
plt.xlabel("Churn (0:No, 1:Yes)")
plt.ylabel("Number of costumers")
plt.xticks([0,1],['No','Yes'])
for i,v in enumerate(df["Churn_Yes"].value_counts()):
    plt.text(i,v,str(v),ha="center",va="bottom")
plt.tight_layout()
plt.show()
```

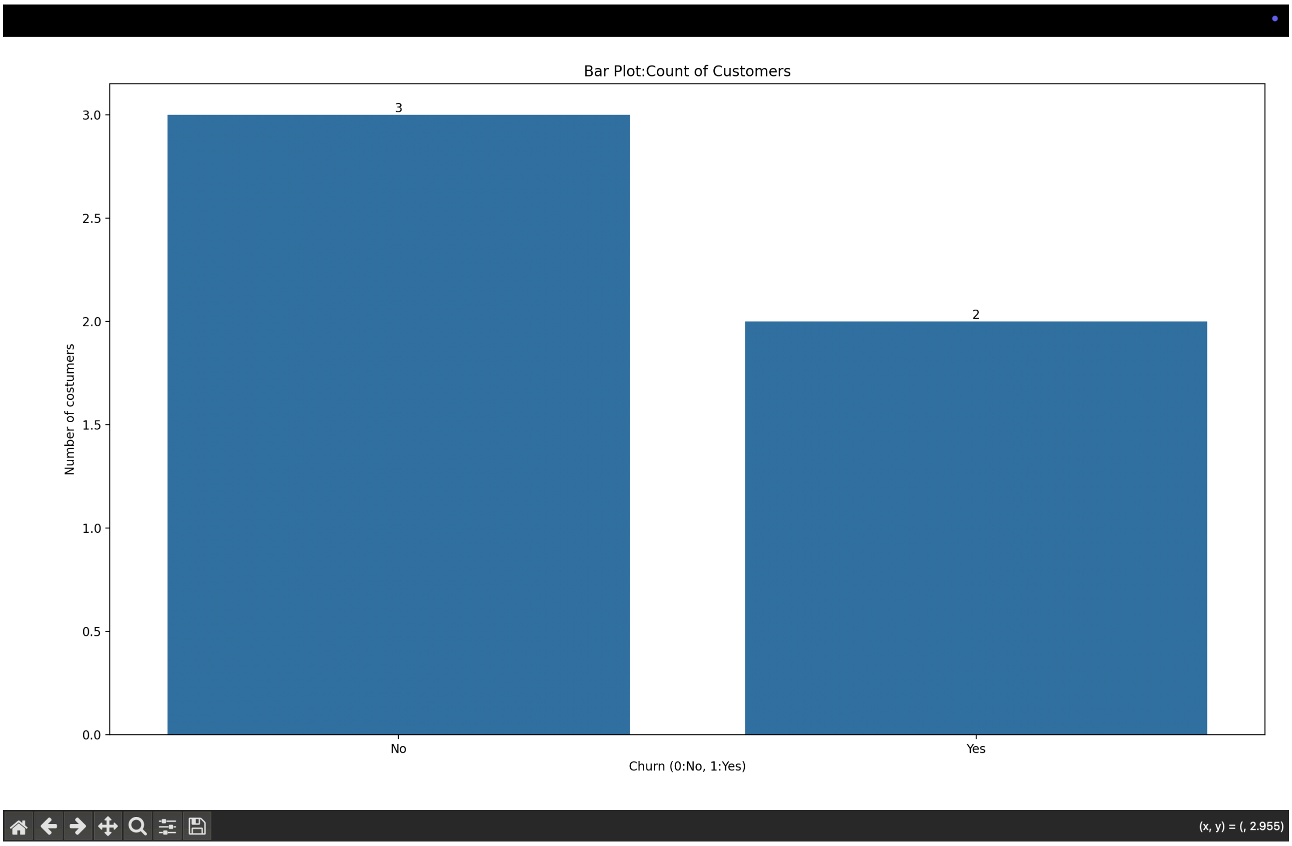
OUTPUT:


BoxPlot:MonthlyCharges
MontghlyCharges ($)
70
65
60
55
50
45
40
35
30
No
Yes
Churn (0=No, 1=Yes)
(x, y) = (, 55.73)


Violion Plot:MonthlyCharges
MontghlyCharges ($)
80
60
40
20
No
Yes
Churn (0=No, 1=Yes)
(x, y) = (, 70.58)



Heatmap:Correlation btw numeric variables
tenure
MonthlyCharges
TotalCharges
ChurnBinary
1
-0.065
0.92
-0.065
1
0.28
0.92
0.28
1
tenure
MonthlyCharges
TotalCharges
ChurnBinary
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
(x, y) = (,)



Bar Plot:Count of Customers
Number of costumers
3.0
2.5
2.0
1.5
1.0
0.5
0.0
3
2
No
Yes
Churn (0:No, 1:Yes)
(x, y) = (, 2.955)