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 model.

TECHNOLOGIES USED:

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

Tasks 2: Exploratory Data Analysis (EDA)

DESCRIPTION:

Calculated and visualised the overall churn rate using a bar chart.Explored customer demographics like gender, senior citizen status, partner, and dependents.Analysed tenure distribution with summary statistics and grouped ranges.Checked churn by contract type and payment method (only "Mailed check" had valid data due to dataset limitations).

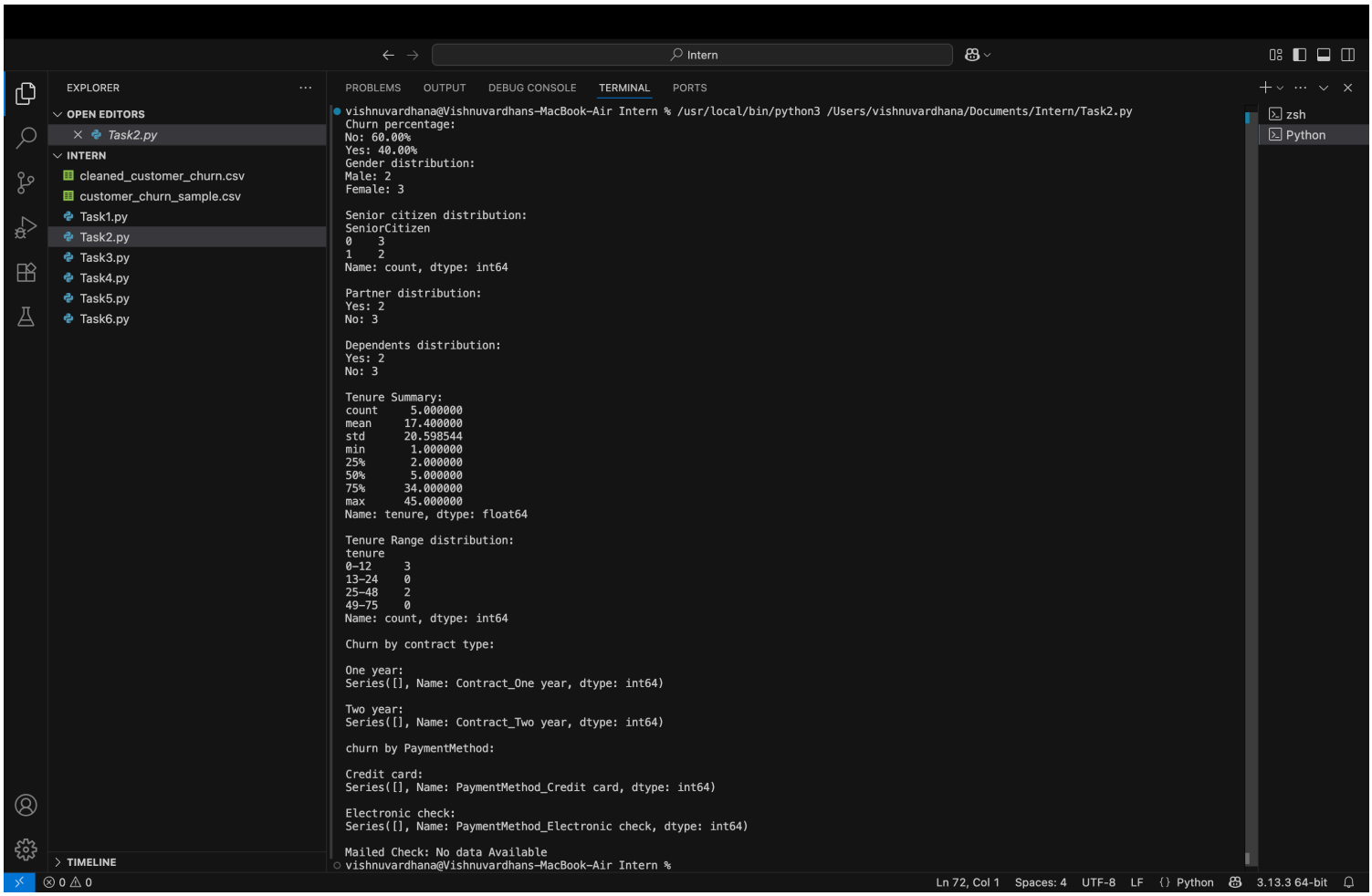
Code Snippet:

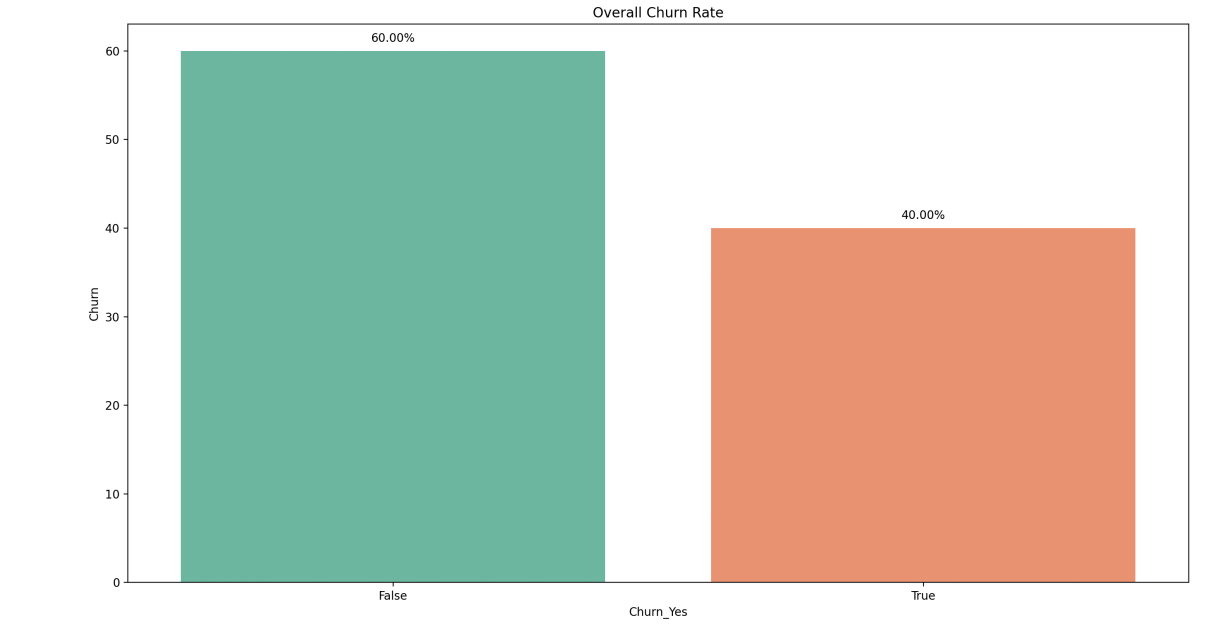
```
import warnings
warnings.simplefilter(action='ignore',category=FutureWarning)

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("cleaned_customer_churn.csv")
churn_counts = df['Churn_Yes'].value_counts()
churn_percent = churn_counts / churn_counts.sum()*100
labels = ['No' if i==0 else 'Yes' for i in churn_percent.index]
print('Churn percentage:')
for i in range (len(labels)):
    print(f"{labels[i]}: {churn_percent.values[i]:.2f}%")


#Bar plot for overall churn rate:
plt.figure(figsize=(6,4))
sns.barplot(x=churn_percent.index,y=churn_percent.values,palette
='Set2')
plt.title('Overall Churn Rate')
plt.ylabel('Percentage')
plt.ylabel('Churn')
```

```
for i, v in enumerate(churn_percent.values):
    plt.text(i, v + 1, f"{v:.2f}%", ha='center')
plt.tight_layout()
plt.show()

#customer distribution by demographics:
gender_counts = df['gender_Male'].value_counts()
print("Gender distribution:")
print("Male:",gender_counts.get(1,0))
print("Female:",gender_counts.get(0,0))
print("\nSenior citizen distribution:")
print(df['SeniorCitizen'].value_counts())
partner_counts = df['Partner_Yes'].value_counts()
print('\nPartner distribution:')
print("Yes:",partner_counts.get(1,0))
print("No:",partner_counts.get(0,0))
dependents_counts=df['Dependents_Yes'].value_counts()
print("\nDependents distribution:")
print("Yes:",dependents_counts.get(1,0))
print("No:",dependents_counts.get(0,0))

#tenure distribution:
print("\nTenure Summary:")
print(df['tenure'].describe())
tenure_bins = pd.cut(df['tenure'], bins=[0,12,24,48,72],
labels=['0-12','13-24','25-48','49-75'])
print("\nTenure Range distribution:")
print(tenure_bins.value_counts().sort_index())

df['Churn_Label']=df['Churn_Yes'].map({1 : 'Yes', 0 : 'No'})
contract_columns=['Contract_One year','Contract_Two year']
print("\nChurn by contract type:")
for col in contract_columns:
    if col in df.columns and df[col].sum()>0:
        print(f'\n{col.replace('Contract_','')}:')
        print(df.groupby('Churn_Label')[col].sum())
    else:
        print(f'\n{col.replace('Contract_','')}: No data Available')
payment_columns=['PaymentMethod_Credit card',
                 'PaymentMethod_Electronic check',
```

```
                'PaymentMethod_Mailed Check']
print("\nchurn by PaymentMethod:")
for col in payment_columns:
    if col in df.columns and df[col].sum()>0:
        print(f'\n{col.replace('PaymentMethod_','')}:')
        print(df.groupby('Churn_Label')[col].sum())
    else:
        print(f'\n{col.replace('PaymentMethod_','')}: No data
Available')
```

OUTPUT:




Overall Churn Rate
60.00%
40.00%
Churn
0
10
20
30
40
50
60
False
True
Churn_Yes