Predicting Customer Churn using Machine Learning to uncover Hidden Patterns

Phase 2 submission template:

- Student Name: N.PAVITHRA
- Register Number: 420423106036
- Institution: Adhiparasakthi engineering college.
- Department: Electronics and communication engineering.
- Date of Submission: 08.05.2025

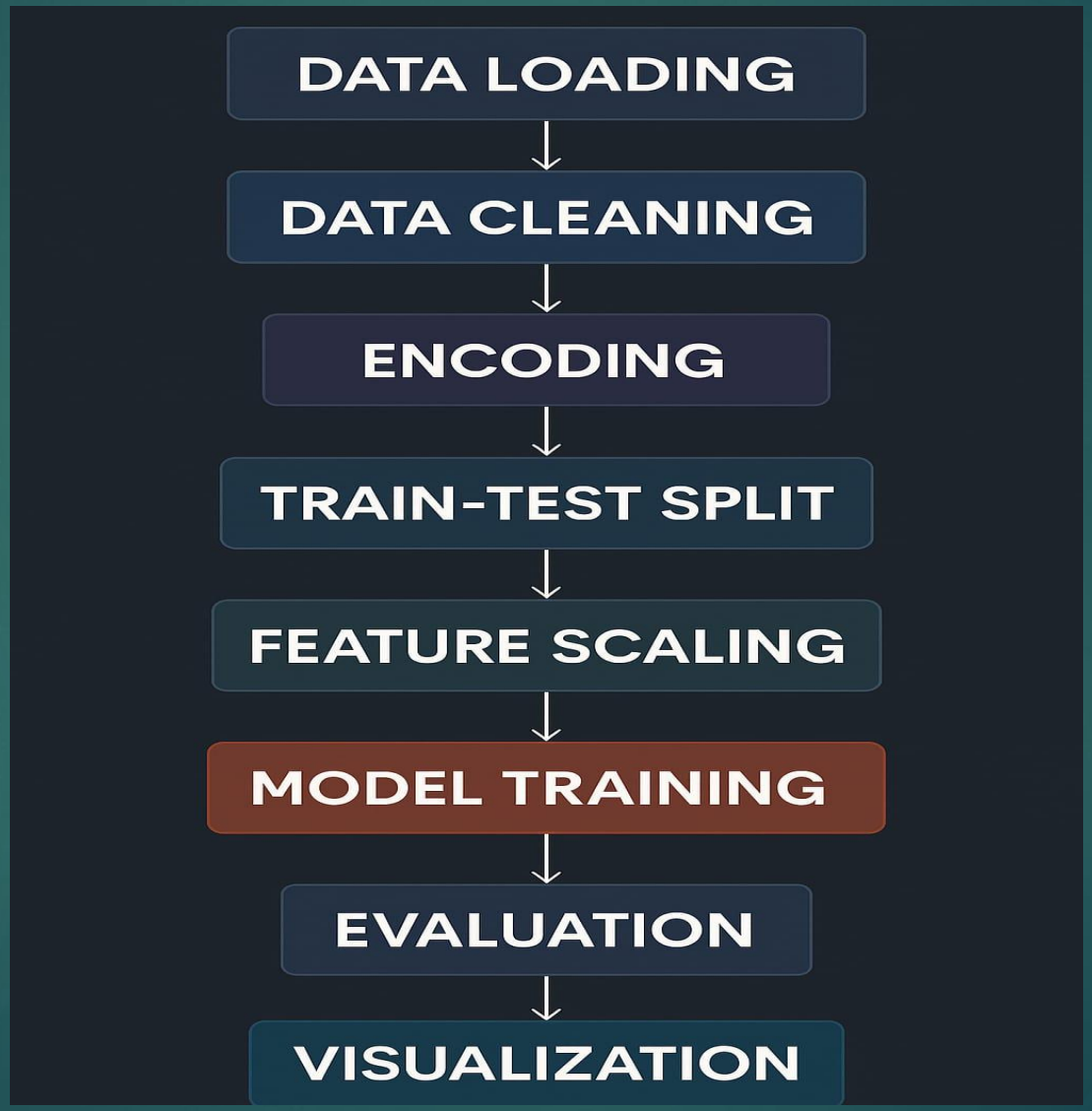
Problem statement

- Customer churn leads to significant revenue loss in the telecom industry.
- Manual methods and static rules fail to capture evolving customer behavior.
- Critical churn indicators often go unnoticed, reducing the window for intervention.
- A machine learning–based solution is essential to proactively detect and retain at-risk customers.

Project Objective:

- Customer churn causes major revenue loss for telecom companies.
- Traditional methods miss hidden patterns and warning signs.
- Key churn drivers like contract type and monthly charges go unnoticed.
- We use **machine learning** to predict and prevent customer churn early.
- Our model highlights top churn indicators using **Logistic Regression**.

Flow chart:


DATA LOADING
DATA CLEANING
ENCODING
TRAIN-TEST SPLIT
FEATURE SCALING
MODEL TRAINING
EVALUATION
VISUALIZATION

Data description:

- **Dataset Name**: Telco-Customer-Churn.csv
- **Source**: Public Telecom Dataset (Kaggle)
- **Data Type**: Structured, Tabular
- **Features Include**:
- Demographics: gender, SeniorCitizen, Partner, Dependents
- Services: InternetService, Contract, PaymentMethod, etc.
- Usage & Billing: tenure, MonthlyCharges, TotalCharges

Data Preprocessing:

- **Handled Missing Values**

→ Converted TotalCharges to numeric & removed null row

- **Dropped Unnecessary Columns**

→ Removed customerID

- **Encoded Categorical Features**

→ Used LabelEncoder to convert text to numbers

- **Scaled Numerical Data**

→ Applied StandardScaler for better model accuracy

Exploratory Data Analysis (EDA)

- **Purpose**: Uncover patterns and identify key features (e.g., MonthlyCharges, Tenure) linked to churn.
- **Visualizations & Insights**:
- **Churn Distribution**: Imbalance in churn vs non-churn suggests oversampling/SMOTE.
- **Histograms**: Shorter tenure and higher charges correlate with churn.
- **Correlation Heatmap**: Strong correlation between features like Tenure and churn.
- **Boxplots**: Churned customers have shorter tenure and higher charges.
- **Key Insight**: Higher charges and shorter tenure strongly indicate churn.

Feature Engineering

Data Preprocessing:

• Handled missing values by converting TotalCharges to numeric and dropping rows with missing data.

•Removed customerID as it does not contribute to churn prediction.

Categorical Encoding:

•Transformed categorical variables into numeric values using label encoding.

Data Splitting:

•Divided the dataset into training (80%) and testing (20%) sets evaluate model performance.

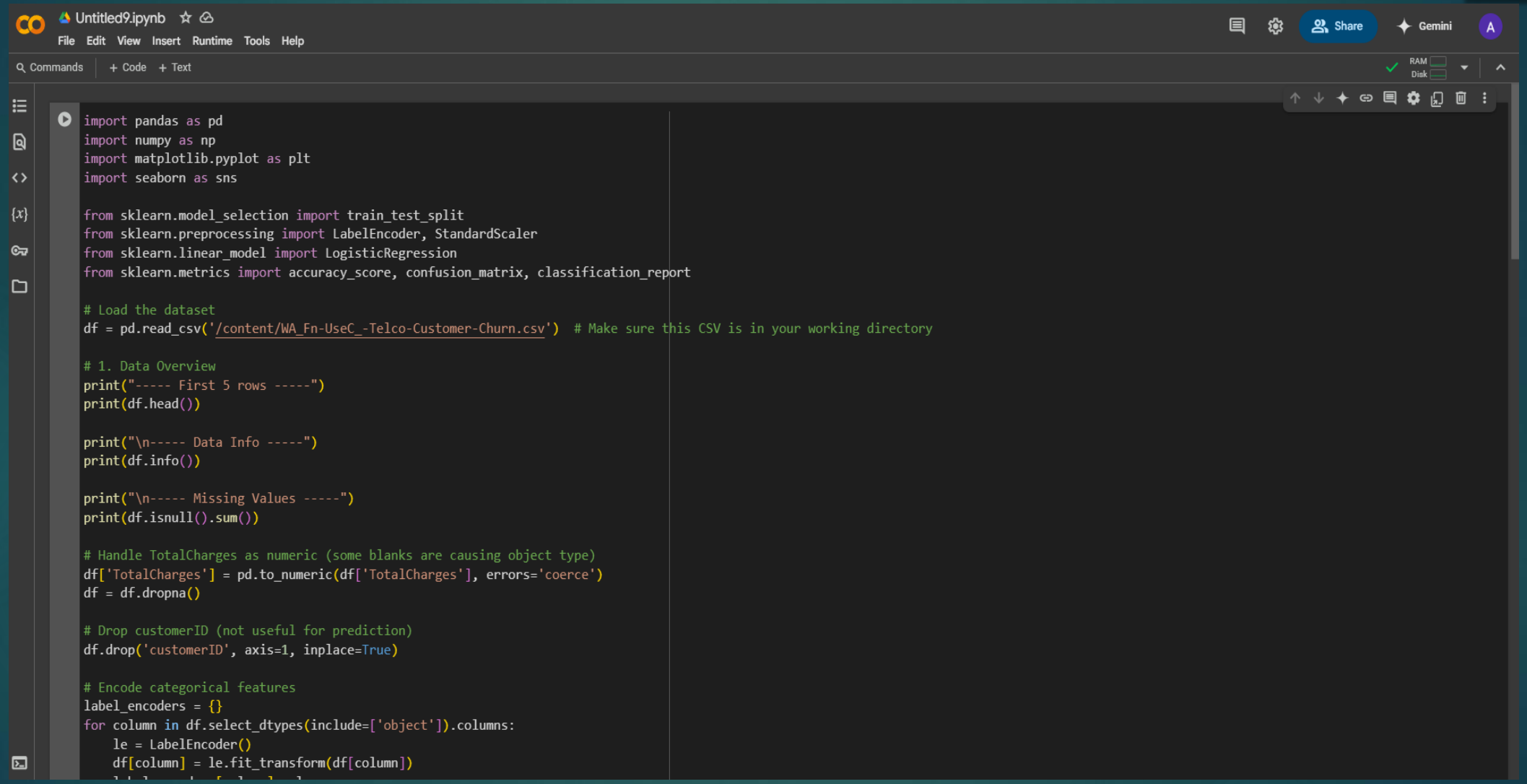
Feature Scaling:

•Standardized the numerical features to ensure equal weight in model training.

Feature Importance:

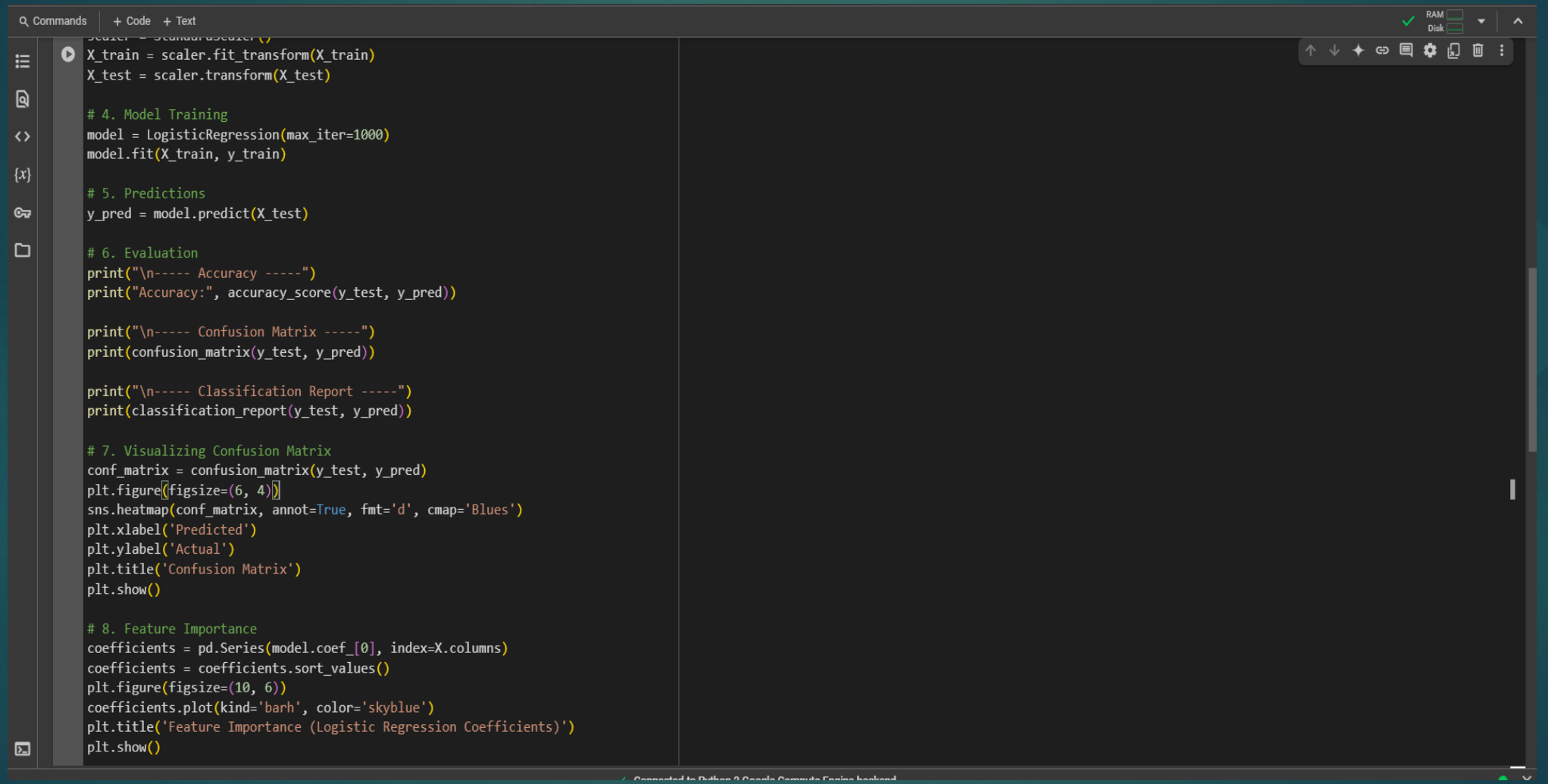
•Identified key features impacting churn prediction by analyzing model coefficients.

Code:



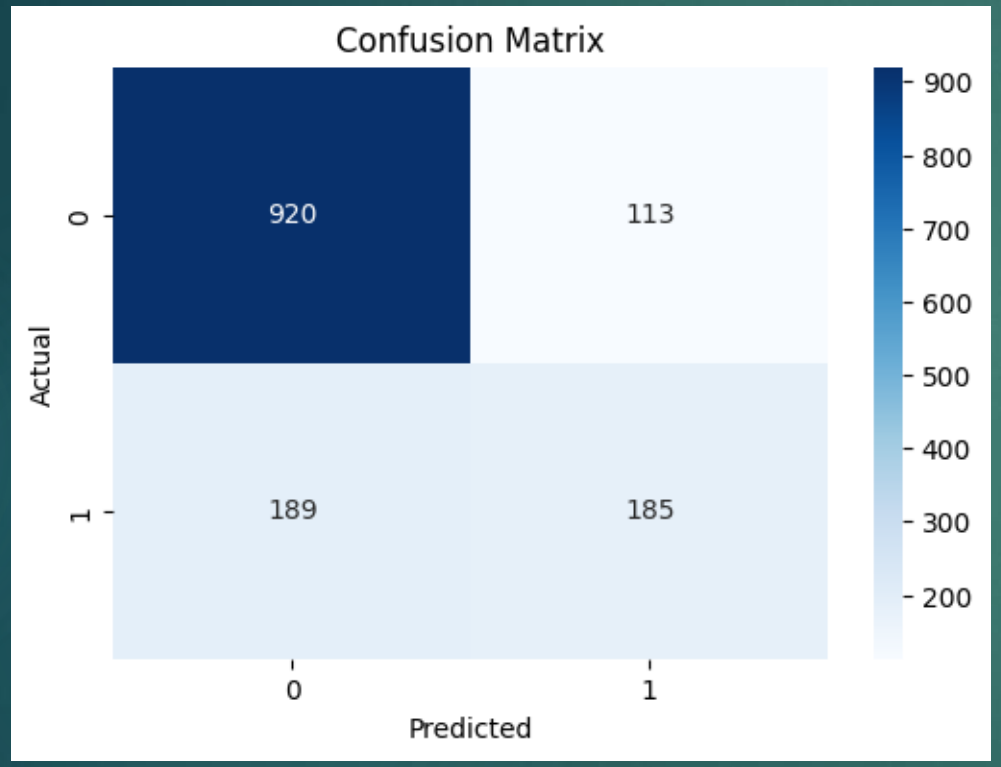

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Load the dataset
df = pd.read_csv('/content/WA_Fn-UseC_-Telco-Customer-Churn.csv')  # Make sure this CSV is in your working directory

# 1. Data Overview
print("----- First 5 rows -----")
print(df.head())

print("\n----- Data Info -----")
print(df.info())

print("\n----- Missing Values -----")
print(df.isnull().sum())

# Handle TotalCharges as numeric (some blanks are causing object type)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df = df.dropna()

# Drop customerID (not useful for prediction)
df.drop('customerID', axis=1, inplace=True)

# Encode categorical features
label_encoders = {}
for column in df.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    df[column] = le.fit_transform(df[column])
```

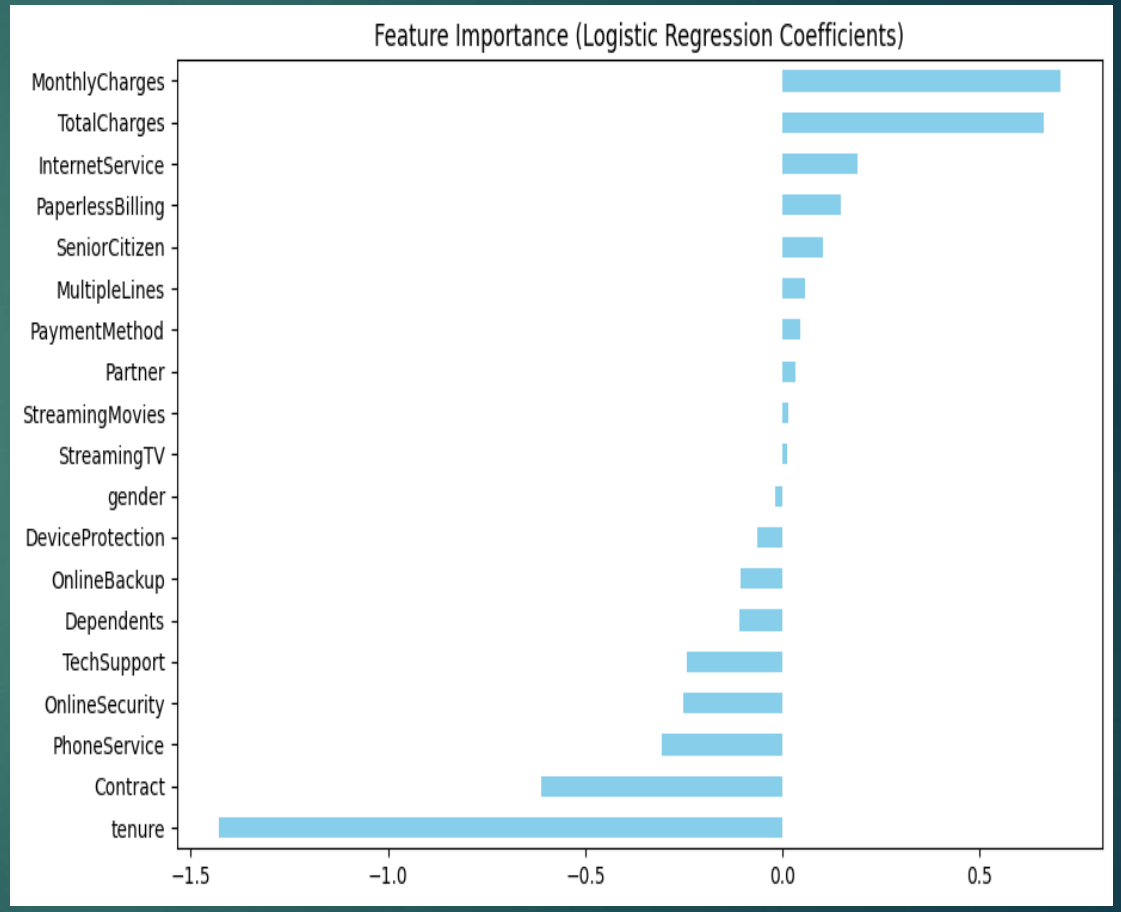
CODE:



```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# 4. Model Training
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# 5. Predictions
y_pred = model.predict(X_test)

# 6. Evaluation
print("\n----- Accuracy -----")
print("Accuracy:", accuracy_score(y_test, y_pred))

print("\n----- Confusion Matrix -----")
print(confusion_matrix(y_test, y_pred))

print("\n----- Classification Report -----")
print(classification_report(y_test, y_pred))

# 7. Visualizing Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

# 8. Feature Importance
coefficients = pd.Series(model.coef_[0], index=X.columns)
coefficients = coefficients.sort_values()
plt.figure(figsize=(10, 6))
coefficients.plot(kind='barh', color='skyblue')
plt.title('Feature Importance (Logistic Regression Coefficients)')
plt.show()
```

Visualization of Results & Model Insights





Tools and Technologies Used

- **Programming Language**: Python
- **IDE/Notebook**: Jupyter Notebook / Google Colab
- **Libraries**:

pandas – data manipulation
numpy – numerical operations
seaborn, matplotlib – data visualization
scikit-learn – preprocessing, model building, evaluation

- **Visualization Tools**:

Used built-in libraries (matplotlib, seaborn) for EDA and model insights

Team Members and Contributions:

- **S. Roshini**– Data Cleaning
- **K. Gayathri** – Exploratory Data Analysis (EDA)
- **A. Savitha** – Feature Engineering
- N. Padmasri--Model Building
- **N. Pavithra** – Documentation & Presentation