

- **Title of Project:**

- Customer Churn Prediction for Telecom Company**

- **Objective:**

- The objective of this project is to analyze customer data and predict which customers are likely to churn, i.e., terminate their subscription with the telecom company. This prediction will help the company take proactive measures to retain customers.

- **Data Source:**

The dataset used for this project is sourced from the telecom company's database, containing information about customer demographics, usage patterns, services subscribed, and whether they churned or not.

Import Library:

```
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 StandardScaler
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import accuracy_score,  
classification_report, confusion_matrix
```

```
'''
```

- **Import Data:**

```
```python
Assuming 'telecom_data.csv' is the name of the dataset file
data = pd.read_csv('telecom_data.csv')
```
```

- **Describe Data**

```
```python
Displaying first few rows of the dataset
print(data.head())

Getting information about the dataset
print(data.info())

Checking for missing values
print(data.isnull().sum())
```
```

- **Data Visualization:**

```
```python
Visualizing the distribution of churners vs non-churners
sns.countplot(x='Churn', data=data)
plt.title('Churn vs Non-Churn')
plt.xlabel('Churn')
plt.ylabel('Count')
plt.show()

Visualizing correlation between features
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
```
```

```
plt.show()
'''
```

- **Data Preprocessing:**

```
```python
```

```
Handling missing values if any
data.dropna(inplace=True)
```

```
Encoding categorical variables if any
```

```
Feature scaling if necessary
```

```
scaler = StandardScaler()
```

```
scaled_features = scaler.fit_transform(data.drop('Churn',
axis=1))
```

```
data_scaled = pd.DataFrame(scaled_features,
columns=data.columns[:-1])
```

```
'''
```

```
Define Target Variable (y) and Feature Variables (X):
```

```
```python
```

```
X = data_scaled # Features
```

```
y = data['Churn'] # Target variable
```

```
'''
```

```
**Train Test Split:**
```

```
```python
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```
`
```

- **Modeling:**

```
```python
# Using Random Forest Classifier as the model
model = RandomForestClassifier(n_estimators=100,
random_state=42)

# Training the model
model.fit(X_train, y_train)
```
```

**\*\*Model Evaluation:\*\***

```
```python
# Predicting on the test set
y_pred = model.predict(X_test)

# Evaluating the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```
```

- **Prediction:**

```
```python
# Assuming new_data is a DataFrame containing new customer
information
```

```
predicted_churn = model.predict(new_data)
print("Predicted Churn:", predicted_churn)
'''
```

- **Explanation:**

This data analytics project focused on predicting customer churn for a telecom company. We imported necessary libraries, imported the dataset, described and visualized the data, preprocessed it by handling missing values and scaling features. We defined the target variable and feature variables, split the data into training and testing sets, and built a Random Forest Classifier model. After evaluating the model's performance, we made predictions on new data and obtained the predicted churn status for customers. This predictive analysis can help the telecom company in devising strategies to retain customers and reduce churn rates.