```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import
LabelEncoder, StandardScaler
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import
classification_report, confusion_matrix,
roc_auc_score
import shap

# Sample dataset
data = {
   'customerID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
   'gender': ['Male', 'Female'] * 5,
   'SeniorCitizen': ['No', 'Yes'] * 5,
   'Partner': ['Yes', 'No'] * 5,
   'Dependents': ['No', 'Yes'] * 5,
   'tenure': [12, 24, 6, 36, 18, 12, 24, 6, 36,
18],
   'PhoneService': ['Yes', 'No'] * 5,
   'MultipleLines': ['Yes', 'No'] * 5,
   'InternetService': ['DSL', 'Fiber optic'] * 5,
   'OnlineSecurity': ['Yes', 'No'] * 5,
   'OnlineBackup': ['Yes', 'No'] * 5,
   'DeviceProtection': ['Yes', 'No'] * 5,
   'TechSupport': ['Yes', 'No'] * 5,
   'StreamingTV': ['Yes', 'No'] * 5,
   'StreamingMovies': ['Yes', 'No'] * 5,
   'Contract': ['Month-to-month', 'One year']
* 5,
   'PaperlessBilling': ['Yes', 'No'] * 5,
   'PaymentMethod': ['Credit card', 'Bank
transfer'] * 5,
   'MonthlyCharges': [50, 60, 40, 70, 55, 65,
45, 75, 50, 60],
   'TotalCharges': [600, 720, 480, 840, 660,
780, 540, 900, 600, 720],
   'Churn': ['No', 'Yes'] * 5
}

df = pd.DataFrame(data)

# Drop customerID
df.drop("customerID", axis=1, inplace=True)

# Encode categorical variables
for column in
df.select_dtypes(include=['object']).columns:
   if column != 'Churn':
      if df[column].nunique() == 2:
         df[column] =
LabelEncoder().fit_transform(df[column])
      else:
         df = pd.get_dummies(df,
columns=[column])

# Encode target
df['Churn'] =
LabelEncoder().fit_transform(df['Churn'])

# Features and target
X = df.drop("Churn", axis=1)
y = df["Churn"]

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

# Scale
scaler = StandardScaler()
X_train_scaled =
scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train model
model =
RandomForestClassifier(n_estimators=100,
random_state=42)
model.fit(X_train_scaled, y_train)

# Evaluate
y_pred = model.predict(X_test_scaled)
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
print("ROC AUC Score:",
roc_auc_score(y_test,
model.predict_proba(X_test_scaled)[:, 1]))
```