```
Import pandas as pd

Import numpy as np

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

Import seaborn as sns

Import matplotlib.pyplot as plt


# 1. Create a synthetic dataset

Data = {

    “gender”: [“Female”, “Male”, “Female”, “Male”, “Female”, “Male”, “Male”, “Female”, “Female”, “Male”],

    “SeniorCitizen”: [0, 1, 0, 1, 0, 0, 1, 1, 0, 0],

    “Partner”: [“Yes”, “No”, “No”, “Yes”, “No”, “Yes”, “Yes”, “No”, “No”, “Yes”],

    “Dependents”: [“No”, “No”, “Yes”, “No”, “Yes”, “No”, “Yes”, “No”, “Yes”, “No”],

    “tenure”: [1, 34, 12, 45, 5, 22, 10, 3, 18, 6],

    “PhoneService”: [“No”, “Yes”, “Yes”, “Yes”, “No”, “Yes”, “No”, “Yes”, “Yes”, “Yes”],

    “MultipleLines”: [“No phone service”, “No”, “Yes”, “Yes”, “No phone service”, “No”, “No phone service”, “Yes”, “Yes”, “No”],

    “InternetService”: [“DSL”, “DSL”, “Fiber optic”, “Fiber optic”, “DSL”, “DSL”, “DSL”, “Fiber optic”, “DSL”, “Fiber optic”],

    “OnlineSecurity”: [“No”, “Yes”, “No”, “Yes”, “No”, “No”, “Yes”, “No”, “Yes”, “No”],

    “OnlineBackup”: [“Yes”, “No”, “Yes”, “No”, “Yes”, “No”, “Yes”, “Yes”, “No”, “Yes”],

    “DeviceProtection”: [“No”, “Yes”, “Yes”, “No”, “Yes”, “Yes”, “No”, “No”, “Yes”, “Yes”],
```

```
    “TechSupport”: [“No”, “No”, “Yes”, “Yes”, “No”, “Yes”, “Yes”, “No”, “No”, “Yes”],
    “StreamingTV”: [“No”, “Yes”, “Yes”, “No”, “No”, “Yes”, “No”, “Yes”, “Yes”, “No”],
    “StreamingMovies”: [“No”, “Yes”, “Yes”, “No”, “No”, “Yes”, “No”, “Yes”, “Yes”, “Yes”],
    “Contract”: [“Month-to-month”, “One year”, “Month-to-month”, “Two year”, “Month-to-month”, “One year”, “Two year”, “Month-to-month”, “One year”, “Month-to-month”],
    “PaperlessBilling”: [“Yes”, “No”, “Yes”, “No”, “Yes”, “Yes”, “No”, “Yes”, “No”, “Yes”],
    “PaymentMethod”: [“Electronic check”, “Mailed check”, “Bank transfer (automatic)”, “Credit card (automatic)”, “Electronic check”,
                “Bank transfer (automatic)”, “Mailed check”, “Credit card (automatic)”, “Mailed check”, “Electronic check”],
    “MonthlyCharges”: [29.85, 56.95, 53.85, 42.30, 70.70, 49.95, 30.00, 80.00, 59.90, 65.25],
    “TotalCharges”: [29.85, 1889.50, 646.00, 1840.75, 151.65, 1098.45, 300.00, 240.00, 1078.20, 391.50],
    “Churn”: [“No”, “No”, “Yes”, “No”, “Yes”, “No”, “Yes”, “Yes”, “No”, “Yes”]
}

Df = pd.DataFrame(data)

# 2. Encode categorical variables
Le = LabelEncoder()
For col in df.select_dtypes(include=’object’).columns:
    Df[col] = le.fit_transform(df[col])

# 3. Scale numeric columns
```

```
Scaler = StandardScaler()
Df[[‘tenure’, ‘MonthlyCharges’, ‘TotalCharges’]] = scaler.fit_transform(df[[‘tenure’, ‘MonthlyCharges’, ‘TotalCharges’]])

# 4. Split data
X = df.drop(‘Churn’, axis=1)
Y = df[‘Churn’]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 5. Train model
Model = RandomForestClassifier(n_estimators=100, random_state=42)
Model.fit(X_train, y_train)

# 6. Predict and evaluate
Y_pred = model.predict(X_test)
Print(“Classification Report:\n”, classification_report(y_test, y_pred))
Print(“Confusion Matrix:\n”, confusion_matrix(y_test, y_pred))

# 7. Feature Importance
Importances = model.feature_importances_
Features = X.columns
Sns.barplot(x=importances, y=features)
Plt.title(“Feature Importances”)
Plt.tight_layout()
Plt.show()
```

Output

Usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))

Classification Report:

	Precision	recall	f1-score	support
0	1.00	0.50	0.67	2
1	0.00	0.00	0.00	0
Accuracy			0.50	2
Macro avg	0.50	0.25	0.33	2
Weighted avg	1.00	0.50	0.67	2

Confusion Matrix:

[[1 1]

[0 0]]