```
!pip install pandas scikit-learn matplotlib
!pip install pandas matplotlib seaborn
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dis
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dis
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/di
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/d
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-pack
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, cross_val_score, StratifiedK
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score, cross_val_predict
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_sco
import matplotlib.pyplot as plt
import seaborn as sns
```

```
diabetes_data = pd.read_csv('/content/diabetes_prediction_dataset.csv')
print(diabetes_data.head())
```

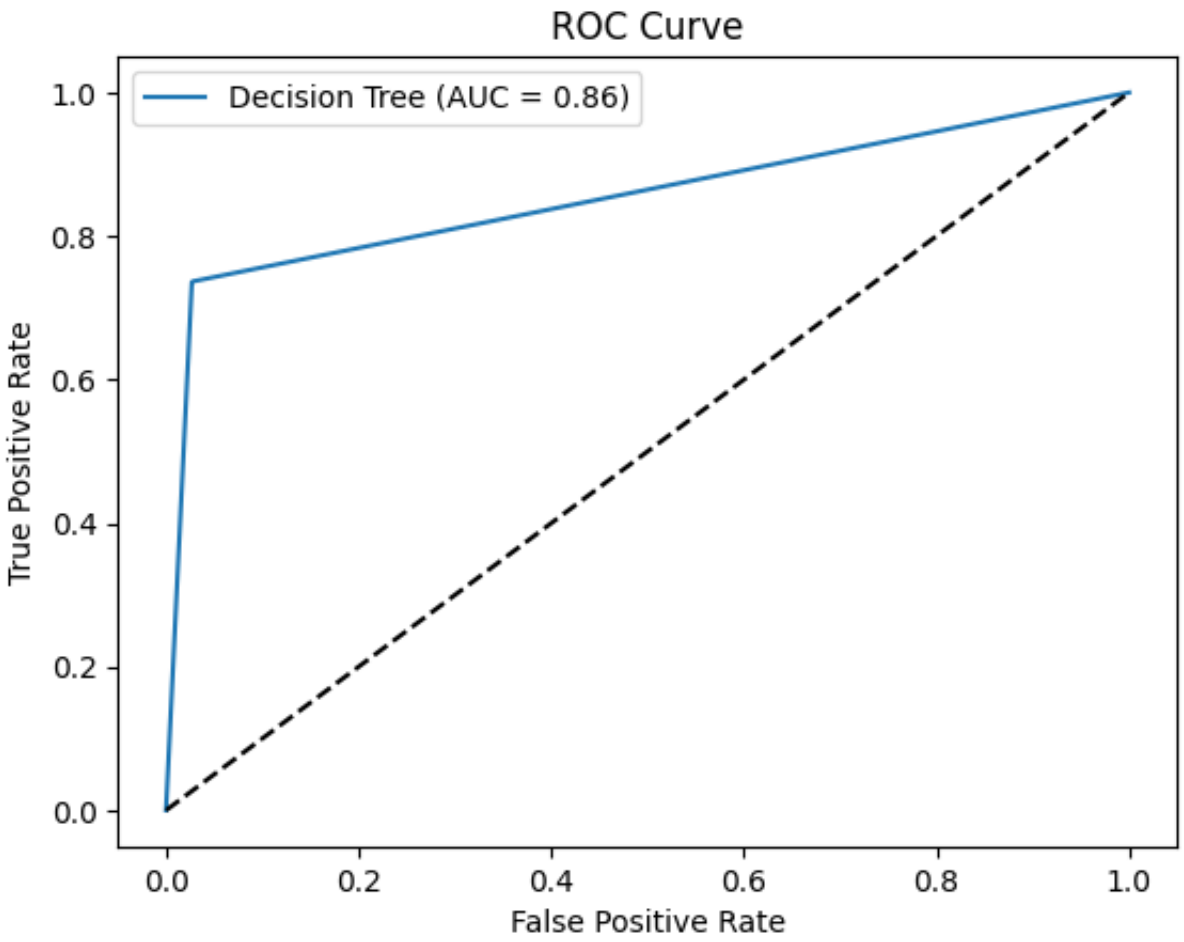
```
   gender   age  hypertension  heart_disease smoking_history    bmi  \
0  Female  80.0             0              1           never  25.19
1  Female  54.0             0              0         No Info  27.32
2    Male  28.0             0              0           never  27.32
3  Female  36.0             0              0         current  23.45
4    Male  76.0             1              1         current  20.14

   HbA1c_level  blood_glucose_level  diabetes
0          6.6                  140         0
1          6.6                   80         0
2          5.7                  158         0
3          5.0                  155         0
4          4.8                  155         0
```

```
# Separate features and target
X = diabetes_data.drop('diabetes', axis=1)
y = diabetes_data['diabetes']

# Identify categorical and numerical columns
categorical_cols = ['gender', 'smoking_history']
numerical_cols = X.columns.difference(categorical_cols)

# Preprocess categorical variables
encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
encoded_categorical_cols = pd.DataFrame(encoder.fit_transform(X[categorical_cols]
encoded_categorical_cols.columns = encoder.get_feature_names_out(categorical_cols

# Preprocess numerical variables
imputer = SimpleImputer(strategy='median')
imputed_numerical_cols = pd.DataFrame(imputer.fit_transform(X[numerical_cols]))
imputed_numerical_cols.columns = numerical_cols

# Combine preprocessed categorical and numerical columns
X_processed = pd.concat([encoded_categorical_cols, imputed_numerical_cols], axis=

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_processed, y, test_size=0.3
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:868
  warnings.warn(
```

```
# Initialize Decision Tree classifier
decision_tree = DecisionTreeClassifier(random_state=42)

# Cross-validation predictions
y_pred_cv = cross_val_predict(decision_tree, X_train, y_train, cv=5)

# Calculate metrics
accuracy = accuracy_score(y_train, y_pred_cv)
precision = precision_score(y_train, y_pred_cv)
recall = recall_score(y_train, y_pred_cv)
f1 = f1_score(y_train, y_pred_cv)
roc_auc = roc_auc_score(y_train, y_pred_cv)

# Fit the model to the full training data and get predictions on the test set
decision_tree.fit(X_train, y_train)
y_pred_test = decision_tree.predict(X_test)
y_pred_proba_test = decision_tree.predict_proba(X_test)[:, 1]
```

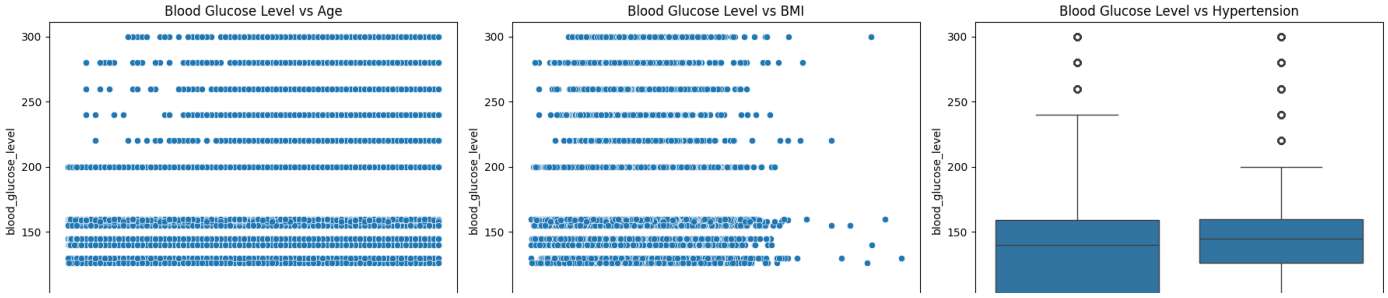
```
# ROC curve
fpr, tpr, _ = roc_curve(y_test, y_pred_proba_test)

# Plot ROC curve
plt.figure()
plt.plot(fpr, tpr, label=f'Decision Tree (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()

# Display metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
print("ROC AUC:", roc_auc)
```

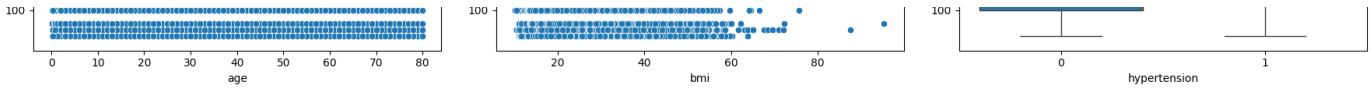


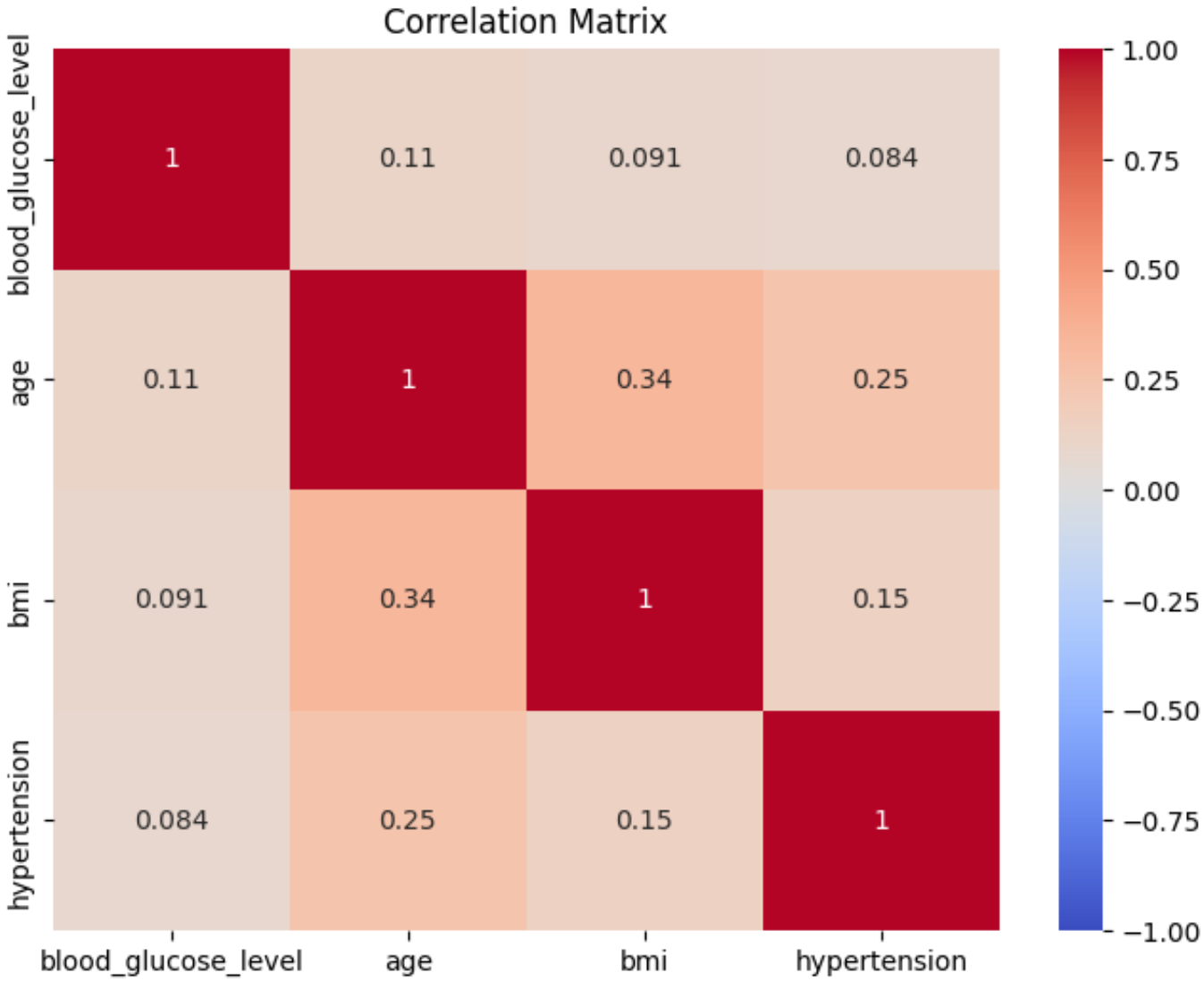

```
Accuracy: 0.9515
Precision: 0.7045745361484325
Recall: 0.7399630438434402
F1 Score: 0.721835313396149
ROC AUC: 0.8555624234471624
```


Start coding or generate with AI.

	blood_glucose_level	age	bmi	hypertension
blood_glucose_level	1.000000	0.110672	0.091261	0.084429
age	0.110672	1.000000	0.337396	0.251171
bmi	0.091261	0.337396	1.000000	0.147666
hypertension	0.084429	0.251171	0.147666	1.000000




100
0 10 20 30 40 50 60 70 80
age
100
20 40 60 80
bmi
100
0 1
hypertension


Correlation Matrix
blood_glucose_level
age
bmi
hypertension
1
0.11
0.091
0.084
0.11
1
0.34
0.25
0.091
0.34
1
0.15
0.084
0.25
0.15
1
blood_glucose_level
age
bmi
hypertension
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00