

```

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# Step 1: Load the dataset
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ["Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin",
           "BMI", "DiabetesPedigreeFunction", "Age", "Outcome"]
df = pd.read_csv(url, names=columns)

# Step 2: Features and Target
X = df.drop("Outcome", axis=1)
y = df["Outcome"]

# Step 3: Train/Test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 4: Train Decision Tree
dt_clf = DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=42)
dt_clf.fit(X_train, y_train)

# Step 5: Train Random Forest
rf_clf = RandomForestClassifier(n_estimators=100, random_state=42)
rf_clf.fit(X_train, y_train)

# Step 6: Compare Accuracy
print("Decision Tree - Training Accuracy:", dt_clf.score(X_train, y_train))
print("Decision Tree - Testing Accuracy:", dt_clf.score(X_test, y_test))

print("Random Forest - Training Accuracy:", rf_clf.score(X_train, y_train))
print("Random Forest - Testing Accuracy:", rf_clf.score(X_test, y_test))

# Step 7: Feature Importance from Random Forest
importances = rf_clf.feature_importances_
feat_importance = pd.Series(importances, index=X.columns).sort_values(ascending=False)

# Step 8: Plot Feature Importance
plt.figure(figsize=(8, 5))
feat_importance.plot(kind='bar', color="skyblue")
plt.title("Feature Importance - Random Forest")
plt.ylabel("Importance Score")
plt.tight_layout()
plt.show()

```

Decision Tree - Training Accuracy: 0.7752442996742671
 Decision Tree - Testing Accuracy: 0.7662337662337663
 Random Forest - Training Accuracy: 1.0
 Random Forest - Testing Accuracy: 0.7207792207792207

