Decision Trees and Random Forests

Objective: Learn tree-based models for classification & regression.

Tools: Scikit-learn, Graphviz

Hints/Mini Guide:

1. Train a Decision Tree Classifier and visualize the tree.
2. Analyze overfitting and control tree depth.
3. Train a Random Forest and compare accuracy.
4. Interpret feature importances.
5. Evaluate using cross-validation.

Dataset: Heart Disease Dataset from UCI repository.

```
import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
url =
'https://raw.githubusercontent.com/ansh941/Machine-Learning-Projects/master/Heart%20Disease%20UCI/heart.csv'
data = pd.read_csv(url)

X = data.drop('target', axis=1)
y = data['target']

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

# Decision Tree
dtree = DecisionTreeClassifier(max_depth=3, random_state=42)
dtree.fit(X_train, y_train)
y_pred_dt = dtree.predict(X_test)
print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))
```

```
# Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))

# Feature Importance
feat_importances = pd.Series(rf.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()

# Cross-validation
cv_scores_dt = cross_val_score(dtree, X, y, cv=5)
cv_scores_rf = cross_val_score(rf, X, y, cv=5)
print("Decision Tree CV Scores:", cv_scores_dt)
print("Random Forest CV Scores:", cv_scores_rf)
```