Task 5: Decision Trees and Random Forests

Objective:

Learn tree-based models for classification & regression.

Tools:

- Python (Scikit-learn, Graphviz/Plotly for visualization)
- Pandas, NumPy, Matplotlib

Step-by-Step Guide

1. Load and Prepare the Dataset

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
import pandas as pd

data = load_breast_cancer()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = data.target

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

2. Train a Decision Tree Classifier and Visualize

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
```

```
dt_model = DecisionTreeClassifier(max_depth=4, random_state=42)
dt_model.fit(X_train, y_train)

plt.figure(figsize=(20, 10))
plot_tree(dt_model, feature_names=data.feature_names, class_names=data.target_names,
filled=True)
plt.title("Decision Tree")
plt.show()
```

3. Analyze Overfitting and Control Tree Depth

```
from sklearn.metrics import accuracy_score

dt_overfit = DecisionTreeClassifier(max_depth=None, random_state=42)
dt_overfit.fit(X_train, y_train)
train_acc = accuracy_score(y_train, dt_overfit.predict(X_train))
test_acc = accuracy_score(y_test, dt_overfit.predict(X_test))

print(f"Overfit Tree - Train Accuracy: {train_acc:.2f}, Test Accuracy: {test_acc:.2f}")
```

4. Train a Random Forest and Compare Accuracy

```
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
rf_train_acc = accuracy_score(y_train, rf_model.predict(X_train))
rf_test_acc = accuracy_score(y_test, rf_model.predict(X_test))

print(f"Random Forest - Train Accuracy: {rf_train_acc:.2f}, Test Accuracy: {rf_test_acc:.2f}")
```

5. Interpret Feature Importances

```
import numpy as np

importances = rf_model.feature_importances_
indices = np.argsort(importances)[::-1]
features = X.columns

plt.figure(figsize=(12, 6))
plt.title("Feature Importances (Random Forest)")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), features[indices], rotation=90)
plt.tight_layout()
plt.show()
```

6. Evaluate Using Cross-Validation

```
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(rf_model, X, y, cv=5)
print(f"Cross-validation Accuracy Scores: {cv_scores}")
```

```
print(f"Mean CV Accuracy: {cv_scores.mean():.2f}")
```

Outcomes:

- You learned how to use Decision Trees and Random Forests.
- Understood how to tune depth and prevent overfitting.
- Explored feature importance and cross-validation.