



Ayesha Zamurd

49733

BSCS-06

Artificial Intelligence

Lab: 13

Train a decision tree to predict whether a student will **Pass (1)** or **Fail (0)** based on their study habits. The dataset is given below

Hours_Studied	Sleep_Hours	Tuition_Attended	Pass
2	5	0	0
4	6	1	1
1	4	0	0
5	7	1	1
3	6	0	0
6	8	1	1
4	5	1	1
2	6	0	0

- Use this dataset in your code as features X and labels Y.
- Train a Decision Tree Classifier.
- Predict the result for a new student who:
 - Studied 3 hours
 - Slept 7 hours
 - Did attend tuition (`Tuition_Attended = 1`)
- Show the application and decision tree diagram.
- Visualize the decision tree.

```

import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

# Create the dataset
data = {
    'Hours_Studied': [2, 4, 1, 5, 3, 6, 4, 2],
    'Sleep_Hours': [5, 6, 4, 7, 6, 8, 5, 6],
    'Tuition_Attended': [0, 1, 0, 1, 0, 1, 1, 0],
    'Pass': [0, 1, 0, 1, 0, 1, 1, 0]
}

df = pd.DataFrame(data)

# Separate features (X) and Labels (y)
X = df[['Hours_Studied', 'Sleep_Hours', 'Tuition_Attended']]
y = df['Pass']

# Display the dataset
print("Dataset:")
print(df)
print("\n" + "="*50 + "\n")

# Train the Decision Tree Classifier
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X, y)

# Make prediction for the new student
new_student = [[3, 7, 1]] # Studied 3 hours, Slept 7 hours, Attended tuition
prediction = dt_classifier.predict(new_student)

print("Prediction for new student:")
print(f"Hours Studied: 3, Sleep Hours: 7, Tuition Attended: 1")
print(f"Predicted Result: {'Pass (1)' if prediction[0] == 1 else 'Fail (0)'}")
print("\n" + "="*50 + "\n")

# Model evaluation (using the same data for demonstration)
y_pred = dt_classifier.predict(X)
accuracy = accuracy_score(y, y_pred)
print(f"Model Accuracy on training data: {accuracy:.2f}")

# Feature importance
feature_importance = dt_classifier.feature_importances_
features = X.columns
print("\nFeature Importance:")
for feature, importance in zip(features, feature_importance):

```

```

print(f"\n{feature}: {importance:.3f}\n")

# Visualize the decision tree
plt.figure(figsize=(12, 8))
plot_tree(dt_classifier,
          feature_names=['Hours_Studied', 'Sleep_Hours', 'Tuition_Attended'],
          class_names=['Fail', 'Pass'],
          filled=True,
          rounded=True,
          fontsize=10)
plt.title('Decision Tree for Student Pass/Fail Prediction')
plt.show()

# Alternative visualization with more details
plt.figure(figsize=(15, 10))
plot_tree(dt_classifier,
          feature_names=['Hours_Studied', 'Sleep_Hours', 'Tuition_Attended'],
          class_names=['Fail', 'Pass'],
          filled=True,
          rounded=True,
          proportion=True,
          fontsize=9)
plt.title('Decision Tree Visualization - Student Performance Prediction', fontsize=14)
plt.tight_layout()
plt.show()

# Print tree structure information
print("\n" + "="*50)
print("DECISION TREE STRUCTURE INFORMATION")
print("="*50)
print(f"Number of nodes: {dt_classifier.tree_.node_count}")
print(f"Tree depth: {dt_classifier.get_depth()}")
print(f"Number of leaves: {dt_classifier.get_n_leaves()}")

# Display the decision path for the new student
print(f"\nDecision path for new student (3, 7, 1):")
decision_path = dt_classifier.decision_path(new_student)
node_indicator = decision_path.toarray()

print("Nodes visited in decision path:", np.where(node_indicator[0] == 1)[0])

```

Dataset:

	Hours_Studied	Sleep_Hours	Tuition_Attended	Pass
0	2	5	0	0
1	4	6	1	1
2	1	4	0	0
3	5	7	1	1
4	3	6	0	0
5	6	8	1	1
6	4	5	1	1
7	2	6	0	0

=====

Prediction for new student:

Hours Studied: 3, Sleep Hours: 7, Tuition Attended: 1
Predicted Result: Pass (1)

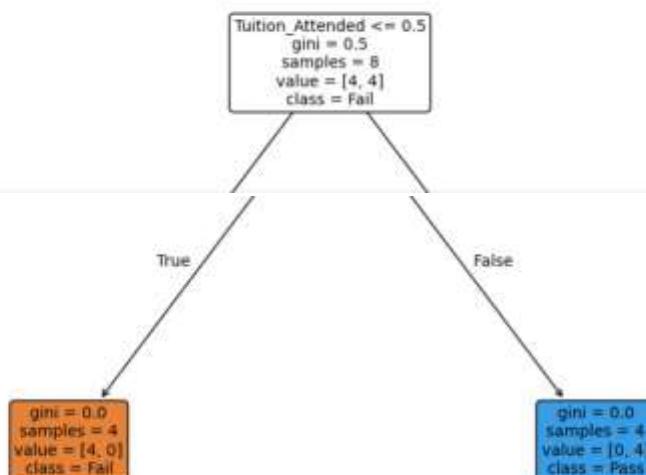
=====

Model Accuracy on training data: 1.00

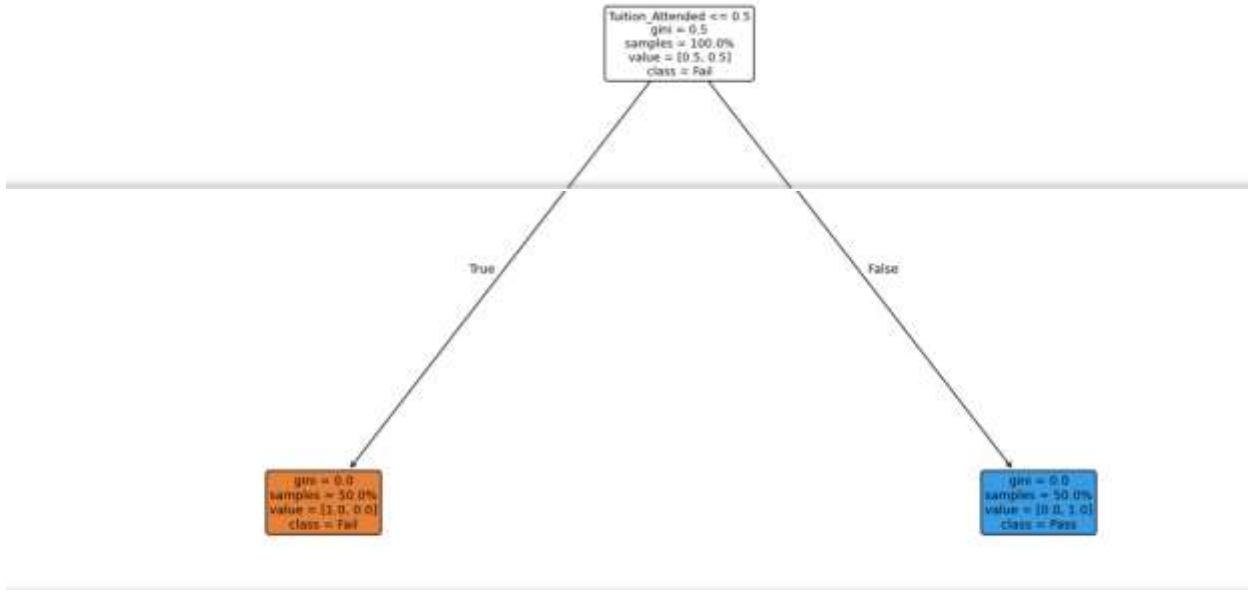
Feature Importance:

Hours_Studied: 0.000
Sleep_Hours: 0.000
Tuition_Attended: 1.000

Decision Tree for Student Pass/Fail Prediction



Decision Tree Visualization - Student Performance Prediction



=====

DECISION TREE STRUCTURE INFORMATION

=====

Number of nodes: 3

Tree depth: 1

Number of leaves: 2

Decision path for new student (3, 7, 1):

Nodes visited in decision path: [0 2]