

Lab 11

Decision Tree Learning with the Restaurant Dataset

1 - Packages

First, let's import the packages needed for decision tree implementation:

```
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
```

2 - Problem Statement

Suppose you are trying to decide whether to wait at a restaurant before getting a table. The decision depends on various attributes such as: - Alternate (Is there an alternative restaurant?) - Bar (Is there a bar to wait in?) - Fri/Sat (Is it Friday or Saturday?) - Hungry (Are you hungry?) - Patrons (How many people are waiting?) - Price (Is it cheap, medium, or expensive?) - Raining (Is it raining?) - Reservation (Do you have a reservation?) - Type (What type of restaurant is it?) - WaitEstimate (Estimated waiting time). Your goal is to build a decision tree that predicts whether you will Wait = Yes/No.

3 - Dataset (Restaurant Dataset from AIMA)

Here's a small sample of the dataset:

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Raining	Reservation	Type	WaitEstimate
Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10
No	Yes	No	No	Full	\$	No	No	Thai	30-60
Yes	No	Yes	Yes	Full	\$\$	Yes	No	Burger	10-30

4 - Building the Decision Tree

Step 1: Load Data

```
data = pd.read_csv("restaurant.csv") # assume dataset in CSV
X = data.drop("Wait", axis=1)
y = data["Wait"]

# Convert categorical variables to numeric
X = pd.get_dummies(X)
```

Step 2: Train Decision Tree

```
tree_clf = DecisionTreeClassifier(criterion="entropy", max_depth=4, random_state=42)
tree_clf.fit(X, y)
```

Step 3: Visualize the Decision Tree

```
plt.figure(figsize=(20,10))
plot_tree(tree_clf, feature_names=X.columns, class_names=["No","Yes"], filled=True)
plt.show()
```

5 - Exercises

Exercise 1: Train the decision tree with criterion="gini" instead of entropy. - Compare the resulting tree. - Does it give the same classification? Exercise 2: Limit the depth of the decision tree to 2 and compare the accuracy with depth = 4.

6 - Evaluation

```
from sklearn.metrics import accuracy_score
y_pred = tree_clf.predict(X)
print("Training Accuracy:", accuracy_score(y, y_pred))
```

7 - Experiment

Try changing the dataset slightly (e.g., removing the Reservation attribute). - Retrain the decision tree. - Observe how the structure of the tree changes.