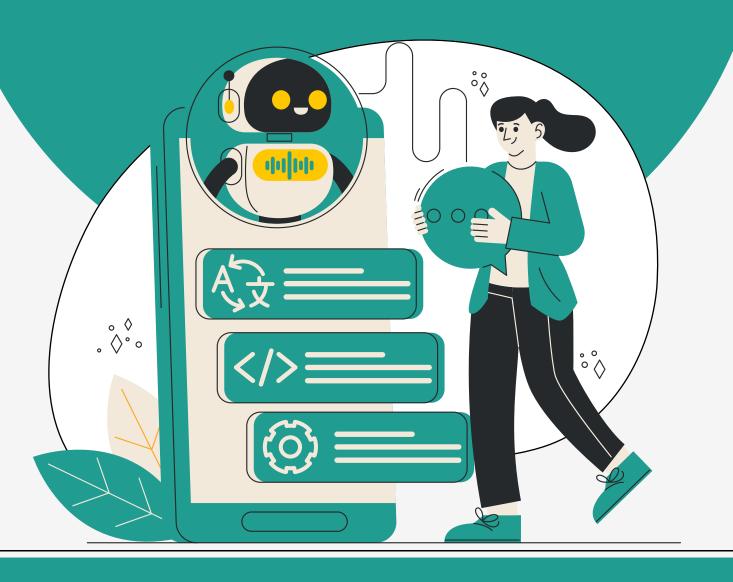# Decision Tree Practical

Interview Questions

(Practice Project)

Easy Questions

1. How would you load the iris dataset and train a Decision Tree classifier using scikit-learn in Python?

Ans:

```python
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split

iris = load_iris()
X_train, X_test, y_train, y_test =
train_test_split(iris.data, iris.target, test_size=0.3,
random_state=42)
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

2. How do you visualize a trained Decision Tree using graphviz in scikit-learn?

Ans:

```python
from sklearn.tree import export_graphviz
import graphviz

dot_data = export_graphviz(clf, out_file=None,
feature_names=iris.feature_names,
class_names=iris.target_names, filled=True)
graph = graphviz.Source(dot_data)
graph.render("iris_tree")
```

3. How do you predict the class of a new sample using a trained Decision Tree classifier?

Answer: We can use the predict() method to make predictions:

```python
new_sample = [[5.1, 3.5, 1.4, 0.2]]
prediction = clf.predict(new_sample)
print(prediction)
```

4. How do you find the accuracy of a Decision Tree model on the test data?

Answer: We can calculate the accuracy using the score() method:

```python
accuracy = clf.score(X_test, y_test)
print(f"Accuracy: {accuracy}")
```

5. How do you set a maximum depth for a Decision Tree in scikit-learn?

Answer: We can set the max_depth parameter when initializing the DecisionTreeClassifier:

```
clf = DecisionTreeClassifier(max_depth=3)
```

6. How do you set the criterion to entropy in a Decision Tree classifier?

Ans: clf = DecisionTreeClassifier(criterion="entropy")

7.How do you split your data into training and testing sets with a 75-25 split?

Answer: Use train_test_split() and set test_size to 0.25:
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.25, random_state=42)

8.How do you ensure that a Decision Tree classifier always splits on a minimum of 5 samples?
Answer: Set the min_samples_split parameter to 5:
clf = DecisionTreeClassifier(min_samples_split=5)

9.How do you extract the feature importances from a trained Decision Tree model?

Answer: Use the feature_importances_ attribute:
importances = clf.feature_importances_
print(importances)

10.How do you ensure that a leaf node in a Decision Tree contains at least 2 samples?

Answer: Set the min_samples_leaf parameter to 2:
clf = DecisionTreeClassifier(min_samples_leaf=2)

Medium Questions:

11. How would you perform 5-fold cross-validation on a Decision Tree model using scikit-learn?

Answer: We can perform 5-fold cross-validation using cross_val_score:

```
from sklearn.model_selection import cross_val_score

scores = cross_val_score(clf, iris.data, iris.target, cv=5)
print(scores)
```

12. How do you tune the hyperparameters of a Decision Tree using GridSearchCV?

Ans:

```
from sklearn.model_selection import GridSearchCV

param_grid = {'max_depth': [3, 5, 7], 'min_samples_split':
[2, 5, 10]}
grid_search = GridSearchCV(DecisionTreeClassifier(),
param_grid, cv=5)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
```

13. How would you plot a confusion matrix for a Decision Tree classifier's predictions?

Answer: Use confusion_matrix and heatmap from seaborn:

```python
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

y_pred = clf.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d')
plt.show()
```

14. How do you handle unbalanced data when training a Decision Tree model?

Answer: We can use class weights to handle unbalanced data:

```python
clf = DecisionTreeClassifier(class_weight='balanced')
```

15. How do you handle overfitting in Decision Trees?

Answer: You can prune the tree or limit its depth to prevent overfitting:

```python
clf = DecisionTreeClassifier(max_depth=5,
min_samples_split=10)
```