```python
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
from math import log2

df = pd.read_csv('EPL.csv')

df = df[['HomeTeam', 'AwayTeam', 'FullTimeResult', 'HalfTimeResult']]
df = df.apply(lambda x: pd.factorize(x)[0])

def entropy(data):
    total = len(data)
    counts = Counter(data)
    return -sum((count/total) * log2(count/total) for count in
counts.values())

target_entropy = entropy(df['HalfTimeResult'])
print(f"Dataset Entropy(half): {target_entropy:.4f}")
target_entropy = entropy(df['FullTimeResult'])
print(f"Dataset Entropy(full): {target_entropy:.4f}")

Dataset Entropy(half): 1.5688
Dataset Entropy(full): 1.5392

X = df.drop(columns=['FullTimeResult'])
y = df['FullTimeResult']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.1, random_state=42)

c45_tree = DecisionTreeClassifier(criterion="entropy", max_depth=3,
min_samples_split=10, min_samples_leaf=5)
c45_tree.fit(X_train, y_train)

DecisionTreeClassifier(criterion='entropy', max_depth=3,
min_samples_leaf=5,
                       min_samples_split=10)

plt.figure(figsize=(12, 6))
plot_tree(c45_tree, feature_names=X.columns, class_names=['H', 'D',
'A'], filled=True)
plt.show()
```
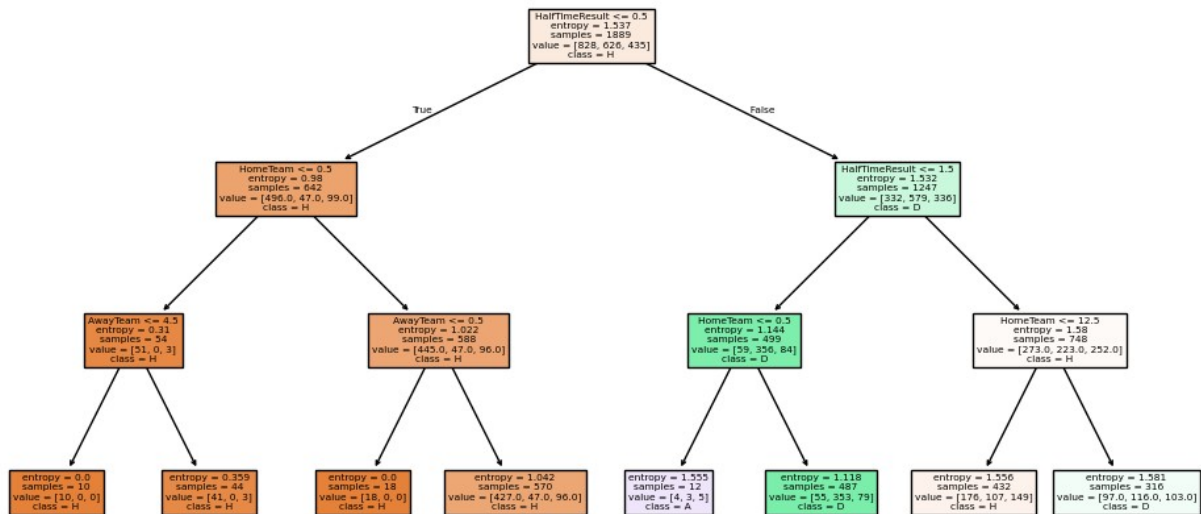
Decision Tree:

- HalfTimeResult <= 0.5, entropy = 1.537, samples = 1889, value = [828, 626, 435], class = H
  - True → HomeTeam <= 0.5, entropy = 0.98, samples = 642, value = [496.0, 47.0, 99.0], class = H
    - AwayTeam <= 4.5, entropy = 0.31, samples = 54, value = [51, 0, 3], class = H
      - entropy = 0.0, samples = 10, value = [10, 0, 0], class = H
      - entropy = 0.359, samples = 44, value = [41, 0, 3], class = H
    - AwayTeam <= 0.5, entropy = 1.022, samples = 588, value = [445.0, 47.0, 96.0], class = H
      - entropy = 0.0, samples = 18, value = [18, 0, 0], class = H
      - entropy = 1.042, samples = 570, value = [427.0, 47.0, 96.0], class = H
  - False → HalfTimeResult <= 1.5, entropy = 1.532, samples = 1247, value = [332, 579, 336], class = D
    - HomeTeam <= 0.5, entropy = 1.144, samples = 499, value = [59, 356, 84], class = D
      - entropy = 1.555, samples = 12, value = [4, 3, 5], class = A
      - entropy = 1.118, samples = 487, value = [55, 353, 79], class = D
    - HomeTeam <= 12.5, entropy = 1.58, samples = 748, value = [273.0, 223.0, 252.0], class = H
      - entropy = 1.556, samples = 432, value = [176, 107, 149], class = H
      - entropy = 1.581, samples = 316, value = [97.0, 116.0, 103.0], class = D

```python
importances = c45_tree.feature_importances_
feature_importance = pd.DataFrame({'Feature': X.columns, 'Importance':
importances})
feature_importance = feature_importance.sort_values(by='Importance',
ascending=False)

print("\nTop Important Features:")
print(feature_importance.head(10))


Top Important Features:
          Feature  Importance
2  HalfTimeResult    0.932996
0        HomeTeam    0.052213
1        AwayTeam    0.014792

y_pred = c45_tree.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")

Accuracy: 0.5905
Precision: 0.5297
```

```
Recall: 0.5905
F1 Score: 0.5145

new_result = {
    'Attribute': 'C4.5',
    'Accuracy': accuracy,
    'Precision': precision,
    'Recall': recall,
    'F1 Score': f1
}

df_existing = pd.read_csv('result.csv')
df_new = pd.DataFrame([new_result])
df_combined = pd.concat([df_existing, df_new], ignore_index=True)
df_combined.to_csv('result.csv', index=False)
```