

```
# Decision Tree Classification
```

```
# Importing the libraries
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import pandas as pd
```

```
# Importing the dataset
```

```
dataset = pd.read_csv(r"/Users/venkat/workspace/gitRepos/python-genAi-agenticAI/day-46-2026Feb2-ml-naive-bayes/logistic-classification.csv")
```

```
X = dataset.iloc[:, [2, 3]].values
```

```
y = dataset.iloc[:, -1].values
```

```
# Splitting the dataset into the Training set and Test set
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

```
'''
```

```
# Feature Scaling
```

```
from sklearn.preprocessing import StandardScaler
```

```
sc = StandardScaler()
```

```
X_train = sc.fit_transform(X_train)
```

```
X_test = sc.transform(X_test)
```

```
'''
```

```
# Training the Decision Tree Classification model on the Training set
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
classifier = DecisionTreeClassifier(criterion='entropy', max_depth=5)
```

```
classifier.fit(X_train, y_train)
```

```
'''
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
classifier = RandomForestClassifier(max_depth=4,n_estimators=30, criterion="entropy",  
random_state=0)
```

```
classifier.fit(X_train, y_train)
```

```
'''
```

```
# Predicting the Test set results
```

```
y_pred = classifier.predict(X_test)
```

```
# Making the Confusion Matrix
```

```
from sklearn.metrics import confusion_matrix
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
print(cm)
```

```
from sklearn.metrics import accuracy_score
```

```
ac = accuracy_score(y_test, y_pred)
```

```
print(ac)
```

```
bias = classifier.score(X_train, y_train)
```

```
bias
```

```
variance = classifier.score(X_test, y_test)
```

```
variance
```

```
# auc & roc curve
```

```
from sklearn.metrics import roc_auc_score, roc_curve
```

```
y_pred_prob = classifier.predict_proba(X_test)[:, 1]
```

```
auc_score = roc_auc_score(y_test, y_pred_prob)
```

```
auc_score
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
```

```
plt.figure(figsize=(8,6))
```

```
plt.plot(fpr, tpr, label=f'Logistic Regression (AUC = {auc_score:.2f})')
```

```
plt.plot([0,1], [0,1], 'k--') # Random classifier line
```

```
plt.xlabel('False Positive Rate')
```

```
plt.ylabel('True Positive Rate')
```

```
plt.title('ROC Curve')
```

```
plt.legend(loc='lower right')
```

```
plt.grid()
```

```
plt.show()
```