

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
```

```
data = pd.read_csv('Bank_Stock_Price_10Y.csv')
data
```

	Open	High	Low	Close	Adj Close	Volume
0	1980	2000	1965	1965	1691.382568	55407000
1	1970	1980	1940	1970	1695.686035	83683500
2	1980	1990	1965	1990	1712.901367	42715000
3	1975	2030	1970	2030	1747.331299	63581000
4	2050	2060	2035	2050	1764.546753	104825500
...	...	...	...	...	...	...
2478	9600	9600	9475	9525	9525.000000	84906000
2479	9450	9575	9450	9500	9500.000000	63212700
2480	9400	9450	9300	9350	9350.000000	78977300
2481	9400	9550	9375	9550	9550.000000	73452800
2482	9500	9650	9475	9650	9650.000000	73136400

2483 rows × 6 columns

```
data.head()
```

	Open	High	Low	Close	Adj Close	Volume
0	1980	2000	1965	1965	1691.382568	55407000
1	1970	1980	1940	1970	1695.686035	83683500
2	1980	1990	1965	1990	1712.901367	42715000
3	1975	2030	1970	2030	1747.331299	63581000
4	2050	2060	2035	2050	1764.546753	104825500

```
X = data.values[:, 1:5]
y = data.values[:, 0]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 100)
```

```
classification_entropy = DecisionTreeClassifier(criterion='entropy', random_state = 100, max_depth = 3, min_samples_leaf = 5)
classification_entropy.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_leaf=5,
                      random_state=100)
```

```
y_pred_entropy = classification_entropy.predict(X_test)
y_pred_entropy
```

```
9000., 2620., 6200., 4450., 9000., 2200., 2200., 9000., 1400.,
2620., 2620., 4450., 9000., 9000., 6000., 3100., 6000., 2200.,
2620., 2620., 2620., 2620., 6000., 7400., 6000., 2200., 6000.,
2620., 2620., 3100., 9000., 2620., 6200., 4450., 6200., 9000.,
6200., 9000., 3100., 9000., 9000., 7400., 6000., 2200., 3100.,
4450., 9000., 4450., 6000., 2200., 6000., 4450., 4450., 2620.,
7400., 4450., 7400., 7400., 6000., 6000., 2620., 6000., 2620.,
4450., 2620., 2200., 9000., 2200., 6200., 9000., 2200., 4450.,
6200., 9000., 4450., 4450., 9000., 6000., 6200., 4450., 3100.,
2620., 2200., 6000., 2620., 4450., 7400., 3100., 3100., 6000.,
9000., 2200., 2620., 4450., 3100., 3100., 2620., 2200.,
6000., 2200., 3100., 9000., 9000., 2620., 2620., 2620., 6000.,
2200., 4450., 4450., 2620., 2620., 3100., 2200., 9000., 6200.,
2200., 9000., 6200., 4450., 4450., 6200., 2620., 6200., 9000.,
9000., 2200., 2620., 9000., 2200., 2620., 9000., 4450., 2620.,
7400., 7400., 3100., 3100., 3100., 3100., 9000., 2200., 6000.,
6200., 7400., 3100., 3100., 2620., 7400., 4450., 7400., 9000.,
6000., 6000., 6000., 9000., 2620., 2620., 3100., 9000., 3100.,
6000., 2620., 4450., 9000., 3100., 6200., 2620., 2200., 2200.,
6200., 2620., 6200., 6000., 7400., 7400., 2200., 2620.,
9000., 3100., 2620., 7400., 3100., 7400., 6000., 2620.,
4450., 7400., 2200., 6000., 6200., 4450., 6000., 2620., 2620.,
7400., 6000., 2620., 6000., 9000., 4450., 7400., 7400.,
4450., 6200., 7400., 6200., 6200., 2200., 2200., 9000., 6000.,
2620., 7400., 2620., 2200., 3100., 4450., 7400., 9000., 3100.,
3100., 4450., 2620., 2620., 3100., 6200., 2200., 6000., 4450.,
7400., 7400., 7400., 2620., 6200., 2200., 6000., 6200., 3100.,
6000., 3100., 6200., 7400., 2620., 6000., 2620., 2200., 2620.,
6000., 4450., 2620., 3100., 7400., 6000., 6000., 4450., 3100.,
6000., 2200., 9000., 9000., 6000., 2200., 2620., 6000., 2200.,
9000., 2620., 2620., 9000., 2620., 2200., 2620., 7400., 9000.,
7400., 4450., 7400., 4450., 6000., 4450., 2620., 6000., 7400.,
2620., 6000., 2620., 3100., 6200., 6200., 2620., 4450., 4450.,
2620., 6200., 7400., 2620., 6000., 6000., 4450., 7400., 7400.,
6200., 7400., 6200., 6000., 6000., 6200., 6000., 9000., 6200.,
4450., 4450., 6000., 6200., 6000., 2620., 2200., 6000., 7400.,
7400., 2620., 4450., 2620., 6000., 2620., 6000., 2620., 2620.,
6200., 2620., 2200., 3100., 3100., 3100., 2620., 2620.,
7400., 3100., 9000., 2200., 9000., 6200., 6000., 9000., 2200.,
3100., 6200., 6200., 6200., 6200., 9000., 4450., 6000., 2620.,
2620., 6200., 2620., 2620., 6200., 9000., 3100., 4450., 3100.,
4450., 3100., 2200., 6200., 2620., 6200., 2620., 4450., 4450.,
2200., 6200., 3100., 6000., 9000., 9000., 3100., 7400., 2620.,
4450., 4450., 7400., 9000., 6200., 3100., 6200., 4450., 3100.,
6000., 6000., 2200., 6200., 2620., 6000., 7400., 2620., 6200.,
9000., 7400., 6000., 2620., 6200., 2620., 2620., 7400., 2620.,
2620., 7400., 4450., 9000., 2200., 7400., 6000., 2200., 4450.,
6200., 4450., 6000., 6200., 6200., 2200., 6000.])
```

```
print(("accuracy: "), accuracy_score(y_test, y_pred_entropy)*100)
```

```
accuracy: 3.6241610738255035
```