

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
df = pd.read_csv("HR-Employee-Attrition.csv")
df.shape
```

Out[2]:

(1470, 35)

In [4]:

```
df.head()
```

Out[4]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educ
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Lif
1	49	No	Travel_Frequently	279	Research & Development	8	1	Lif
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Lif
4	27	No	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

In [5]:

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [6]:

```
le = LabelEncoder()
```

In [7]:

```
le_count = 0
for col in df.columns[1:]:
    if df[col].dtype == 'object':
        if len(list(df[col].unique())) <= 2:
            le.fit(df[col])
            df[col] = le.transform(df[col])
            le_count += 1
print('{} columns were label encoded.'.format(le_count))
```

4 columns were label encoded.

In [8]:

```
df.head()
```

Out[8]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Educ
0	41	1	Travel_Rarely	1102	Sales	1	2	Lif
1	49	0	Travel_Frequently	279	Research & Development	8	1	Lif
2	37	1	Travel_Rarely	1373	Research & Development	2	2	
3	33	0	Travel_Frequently	1392	Research & Development	3	4	Lif
4	27	0	Travel_Rarely	591	Research & Development	2	1	

5 rows × 35 columns

In [10]:

```
df = pd.get_dummies(df, drop_first=True)
```

In [11]:

```
df.dtypes
```

Out[11]:

Age	int64
Attrition	int64
DailyRate	int64
DistanceFromHome	int64
Education	int64
EmployeeCount	int64
EmployeeNumber	int64
EnvironmentSatisfaction	int64
Gender	int64
HourlyRate	int64
JobInvolvement	int64
JobLevel	int64
JobSatisfaction	int64
MonthlyIncome	int64
MonthlyRate	int64
NumCompaniesWorked	int64
Over18	int64
OverTime	int64
PercentSalaryHike	int64
PerformanceRating	int64
RelationshipSatisfaction	int64
StandardHours	int64
StockOptionLevel	int64
TotalWorkingYears	int64
TrainingTimesLastYear	int64
WorkLifeBalance	int64
YearsAtCompany	int64
YearsInCurrentRole	int64
YearsSinceLastPromotion	int64
YearsWithCurrManager	int64
BusinessTravel_Travel_Frequently	uint8
BusinessTravel_Travel_Rarely	uint8
Department_Research & Development	uint8
Department_Sales	uint8
EducationField_Life Sciences	uint8
EducationField_Marketing	uint8
EducationField_Medical	uint8
EducationField_Other	uint8
EducationField_Technical Degree	uint8
JobRole_Human Resources	uint8
JobRole_Laboratory Technician	uint8
JobRole_Manager	uint8
JobRole_Manufacturing Director	uint8
JobRole_Research Director	uint8
JobRole_Research Scientist	uint8
JobRole_Sales Executive	uint8
JobRole_Sales Representative	uint8
MaritalStatus_Married	uint8
MaritalStatus_Single	uint8
dtype:	object

In [12]:

```
df.head()
```

Out[12]:

	Age	Attrition	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber
0	41	1	1102	1	2	1	1
1	49	0	279	8	1	1	2
2	37	1	1373	2	2	1	4
3	33	0	1392	3	4	1	5
4	27	0	591	2	1	1	7

5 rows × 49 columns

In [13]:

```
target = df['Attrition']
```

In [15]:

```
df = df.drop(columns=['Attrition'])
```

In [16]:

```
target.shape
```

Out[16]:

(1470,)

In [17]:

```
df.shape
```

Out[17]:

(1470, 48)

In [19]:

```
df = df.drop(columns=['EmployeeCount', 'EmployeeNumber', 'StandardHours', 'Over18'])
```

In [20]:

```
df.shape
```

Out[20]:

(1470, 44)

In [25]:

```
df.head()
```

Out[25]:

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	Gender	HourlyRate
0	41	1102	1	2	2	0	94
1	49	279	8	1	3	1	61
2	37	1373	2	2	4	1	92
3	33	1392	3	4	4	0	56
4	27	591	2	1	1	1	40

5 rows × 44 columns

In [28]:

```
target.value_counts(normalize = True)
```

Out[28]:

```
0    0.838776
1    0.161224
Name: Attrition, dtype: float64
```

In [22]:

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

In [29]:

```
X_train, X_test, y_train, y_test = train_test_split(df, target, test_size=0.25, rand
```

In [30]:

```
X_train.shape, y_train.shape
```

Out[30]:

```
((1102, 44), (1102,))
```

In [31]:

```
X_test.shape, y_test.shape
```

Out[31]:

```
((368, 44), (368,))
```

In [33]:

```
y_train.value_counts(normalize=True)
```

Out[33]:

```
0    0.838475
1    0.161525
Name: Attrition, dtype: float64
```

In [34]:

```
y_test.value_counts(normalize=True)
```

Out[34]:

```
0    0.839674
1    0.160326
Name: Attrition, dtype: float64
```

In [35]:

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

In [61]:

```
tree = DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features=20)
```

In [62]:

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

Out[62]:

```
DecisionTreeClassifier(criterion='entropy', max_depth=3, max_features=
20)
```

In [63]:

```
tree.score(X_train, y_train)
```

Out[63]:

```
0.8448275862068966
```

In [64]:

```
tree.score(X_test, y_test)
```

Out[64]:

```
0.8396739130434783
```

In [ ]:

In [ ]:

