In [1]:
```
pwd
```

Out[1]:
```
'C:\\Users\\tuhin'
```

In [2]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

In [3]:
```
df = pd.read_csv(r"C:\Users\tuhin\HR_Analytics - Copy.csv")
```

In [4]:
```
df
```

Out[4]:

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHon
0	RM297	18	18-25	Yes	Travel_Rarely	230	Research & Development	
1	RM302	18	18-25	No	Travel_Rarely	812	Sales	
2	RM458	18	18-25	Yes	Travel_Frequently	1306	Sales	
3	RM728	18	18-25	No	Non-Travel	287	Research & Development	
4	RM829	18	18-25	Yes	Non-Travel	247	Research & Development	
...	...	...	...	...	...	...	...	
1475	RM412	60	55+	No	Travel_Rarely	422	Research & Development	
1476	RM428	60	55+	No	Travel_Frequently	1499	Sales	
1477	RM537	60	55+	No	Travel_Rarely	1179	Sales	
1478	RM880	60	55+	No	Travel_Rarely	696	Sales	
1479	RM1210	60	55+	No	Travel_Rarely	370	Research & Development	

1480 rows × 38 columns

In [5]:
```
df = df[['Age', 'Attrition']]
```

In [6]:
```
df['Attrition'] = df['Attrition'].astype('category')
df['Attrition_Code'] = df['Attrition'].cat.codes
```

```
C:\Users\tuhin\AppData\Local\Temp\ipykernel_12632\417316360.py:1: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df['Attrition'] = df['Attrition'].astype('category')
C:\Users\tuhin\AppData\Local\Temp\ipykernel_12632\417316360.py:2: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df['Attrition_Code'] = df['Attrition'].cat.codes
```

In [7]:
```
mean_age = df['Age'].mean()
std_age = df['Age'].std()
df['Standard_Age'] = (df['Age'] - mean_age) / std_age
```

```
C:\Users\tuhin\AppData\Local\Temp\ipykernel_12632\3810137463.py:3: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  df['Standard_Age'] = (df['Age'] - mean_age) / std_age
```

In [8]:
```
X = df[['Standard_Age']]
y = df['Attrition_Code']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state
```

In [9]:
```
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
nb_score = nb_classifier.score(X_test, y_test)
nb_pred = nb_classifier.predict(X_test)
```

In [10]:
```
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(X_train, y_train)
dt_score = dt_classifier.score(X_test, y_test)
dt_pred = dt_classifier.predict(X_test)
```

In [11]:
```
def evaluate_model(name, classifier, X_test, y_test, pred):
    print(name + " Classifier:")
    print("Score:", classifier.score(X_test, y_test))
    print("Confusion Matrix:\n", confusion_matrix(y_test, pred))
    print("Classification Report:\n", classification_report(y_test, pred))
```

In [12]:
```
evaluate_model("Naive Bayes", nb_classifier, X_test, y_test, nb_pred)
evaluate_model("Decision Tree", dt_classifier, X_test, y_test, dt_pred)
```

```
Naive Bayes Classifier:
Score: 0.8432432432432433
Confusion Matrix:
 [[312   0]
 [ 58   0]]
Classification Report:
               precision    recall  f1-score   support

           0       0.84      1.00      0.91       312
           1       0.00      0.00      0.00        58

    accuracy                           0.84       370
   macro avg       0.42      0.50      0.46       370
weighted avg       0.71      0.84      0.77       370

Decision Tree Classifier:
Score: 0.8405405405405405
Confusion Matrix:
 [[308   4]
 [ 55   3]]
Classification Report:
               precision    recall  f1-score   support

           0       0.85      0.99      0.91       312
           1       0.43      0.05      0.09        58

    accuracy                           0.84       370
   macro avg       0.64      0.52      0.50       370
weighted avg       0.78      0.84      0.78       370
```

```
C:\Users\tuhin\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: U
ndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in
labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\tuhin\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: U
ndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in
labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\tuhin\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1344: U
ndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in
labels with no predicted samples. Use `zero_division` parameter to control this behav
ior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: