

```

se import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

```

```
data = pd.read_csv("/content/Admission_Predict.csv")
```

```
data.head()
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

## Preprocessing

```
print(data.isnull().sum())
```

```

Serial No.      0
GRE Score      0
TOEFL Score    0
University Rating 0
SOP            0
LOR            0
CGPA           0
Research       0
Chance of Admit 0
dtype: int64

```

```
data.columns
```

```

Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
      'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
      dtype='object')

```

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

```
numerical_features = ['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR ', 'CGPA']
```

```
# Initialize the scaler
```

```
scaler = StandardScaler()
```

```
# Fit and transform the numerical features
```

```
data[numerical_features] = scaler.fit_transform(data[numerical_features])
```

```
print(data.head())
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR \
0	1	1.762107	1.746971	0.798829	1.093864	1.167321
1	2	0.627656	-0.067635	0.798829	0.596653	1.167321
2	3	-0.070467	-0.562528	-0.076600	-0.397769	0.052933
3	4	0.453126	0.427257	-0.076600	0.099442	-1.061454
4	5	-0.244998	-0.727492	-0.952029	-1.392191	-0.504260

  

	CGPA	Research	Chance of Admit
0	1.764818	1	0.92
1	0.455151	1	0.76
2	-1.005631	1	0.72
3	0.119339	1	0.80
4	-0.653029	0	0.65

```
y = data.iloc[:, -1].round()
```

```
y
```

```

0      1.0
1      1.0
2      1.0
3      1.0
4      1.0

```

```

...
395    1.0
396    1.0
397    1.0
398    1.0
399    1.0
Name: Chance of Admit , Length: 400, dtype: float64

```

### Split dataset

```

X = data[['GRE Score', 'CGPA']]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```

clf = DecisionTreeClassifier()

```

```

clf.fit(X_train, y_train)

```

```

DecisionTreeClassifier
DecisionTreeClassifier()

```

```

y_pred = clf.predict(X_test)

```

```

accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

```

```

print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)

```

```

Accuracy: 0.875
Confusion Matrix:
[[ 4  6]
 [ 4 66]]
Classification Report:

```

	precision	recall	f1-score	support
0.0	0.50	0.40	0.44	10
1.0	0.92	0.94	0.93	70
accuracy			0.88	80
macro avg	0.71	0.67	0.69	80
weighted avg	0.86	0.88	0.87	80