

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn import metrics
from sklearn.metrics import accuracy_score

```

```

df= pd.read_csv('/content/Cancer_DS.csv')
df.head()

```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean \
0	842302	M	17.99	10.38	122.80	1001.0
1	842517	M	20.57	17.77	132.90	1326.0
2	84300903	M	19.69	21.25	130.00	1203.0
3	84348301	M	11.42	20.38	77.58	386.1
4	84358402	M	20.29	14.34	135.10	1297.0

	smoothness_mean	compactness_mean	concavity_mean	concave	points_mean \
0	0.11840	0.27760	0.3001		0.14710
1	0.08474	0.07864	0.0869		0.07017
2	0.10960	0.15990	0.1974		0.12790
3	0.14250	0.28390	0.2414		0.10520
4	0.10030	0.13280	0.1980		0.10430

	...	texture_worst	perimeter_worst	area_worst	smoothness_worst \
0	...	17.33	184.60	2019.0	0.1622
1	...	23.41	158.80	1956.0	0.1238
2	...	25.53	152.50	1709.0	0.1444
3	...	26.50	98.87	567.7	0.2098
4	...	16.67	152.20	1575.0	0.1374

	compactness_worst	concavity_worst	concave points_worst
0	0.6656	0.7119	0.2654
1	0.1866	0.2416	0.1860
2	0.4245	0.4504	0.2430
3	0.8663	0.6869	0.2575
4	0.2050	0.4000	0.1625

	fractal_dimension_worst	Unnamed: 32
0	0.11890	NaN
1	0.08902	NaN
2	0.08758	NaN
3	0.17300	NaN
4	0.07678	NaN

[5 rows x 33 columns]

df.isnull().sum()

id	0
diagnosis	0
radius_mean	0
texture_mean	0
perimeter_mean	0
area_mean	0
smoothness_mean	0
compactness_mean	0
concavity_mean	0
concave points_mean	0
symmetry_mean	0
fractal_dimension_mean	0
radius_se	0
texture_se	0
perimeter_se	0
area_se	0
smoothness_se	0
compactness_se	0
concavity_se	0
concave points_se	0
symmetry_se	0
fractal_dimension_se	0
radius_worst	0
texture_worst	0
perimeter_worst	0

```

area_worst      0
smoothness_worst 0
compactness_worst 0
concavity_worst 0
concave points_worst 0
symmetry_worst 0
fractal_dimension_worst 0
Unnamed: 32      569
dtype: int64

```

```

X = df.drop(['id', 'diagnosis', 'Unnamed: 32'], axis=1)
y = df['diagnosis']

```

```

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

```

```

dt_clf = DecisionTreeClassifier(random_state=42)

```

```

dt_clf.fit(X_train, y_train)

```

```

DecisionTreeClassifier(random_state=42)

```

```

y_pred = dt_clf.predict(X_test)

```

```

accuracy = accuracy_score(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
print(class_report)

```

	precision	recall	f1-score	support
B	0.96	0.96	0.96	71
M	0.93	0.93	0.93	43
accuracy			0.95	114
macro avg	0.94	0.94	0.94	114
weighted avg	0.95	0.95	0.95	114

```

print(f"Accuracy: {accuracy:.4f}")
print("\nClassification Report:\n", class_report)

```

```

Accuracy: 0.9474

```

```

Classification Report:

```

	precision	recall	f1-score	support
B	0.96	0.96	0.96	71
M	0.93	0.93	0.93	43
accuracy			0.95	114
macro avg	0.94	0.94	0.94	114

weighted avg	0.95	0.95	0.95	114
--------------	------	------	------	-----