

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

```
df=pd.read_csv('/content/drive/MyDrive/AI ML Bootcamp/Week 1/diabetes_dataset.csv')
```

```
df.head(1)
```



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1

```
df.columns
```



```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
      'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
df.isna().sum()
```



	0
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

```
df.isnull().sum()
```



	0
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Outcome	0

```
df.dtypes
```



	0
<b>Pregnancies</b>	int64
<b>Glucose</b>	int64
<b>BloodPressure</b>	int64
<b>SkinThickness</b>	int64
<b>Insulin</b>	int64
<b>BMI</b>	float64
<b>DiabetesPedigreeFunction</b>	float64
<b>Age</b>	int64
<b>Outcome</b>	int64



```
df.nunique()
```



	0
<b>Pregnancies</b>	17
<b>Glucose</b>	136
<b>BloodPressure</b>	47
<b>SkinThickness</b>	51
<b>Insulin</b>	186
<b>BMI</b>	248
<b>DiabetesPedigreeFunction</b>	517
<b>Age</b>	52
<b>Outcome</b>	2



```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
```

```
x = df.drop('Outcome', axis=1)
y = df['Outcome']
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
model=DecisionTreeClassifier(random_state=42)
model.fit(x_train,y_train)
```



```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

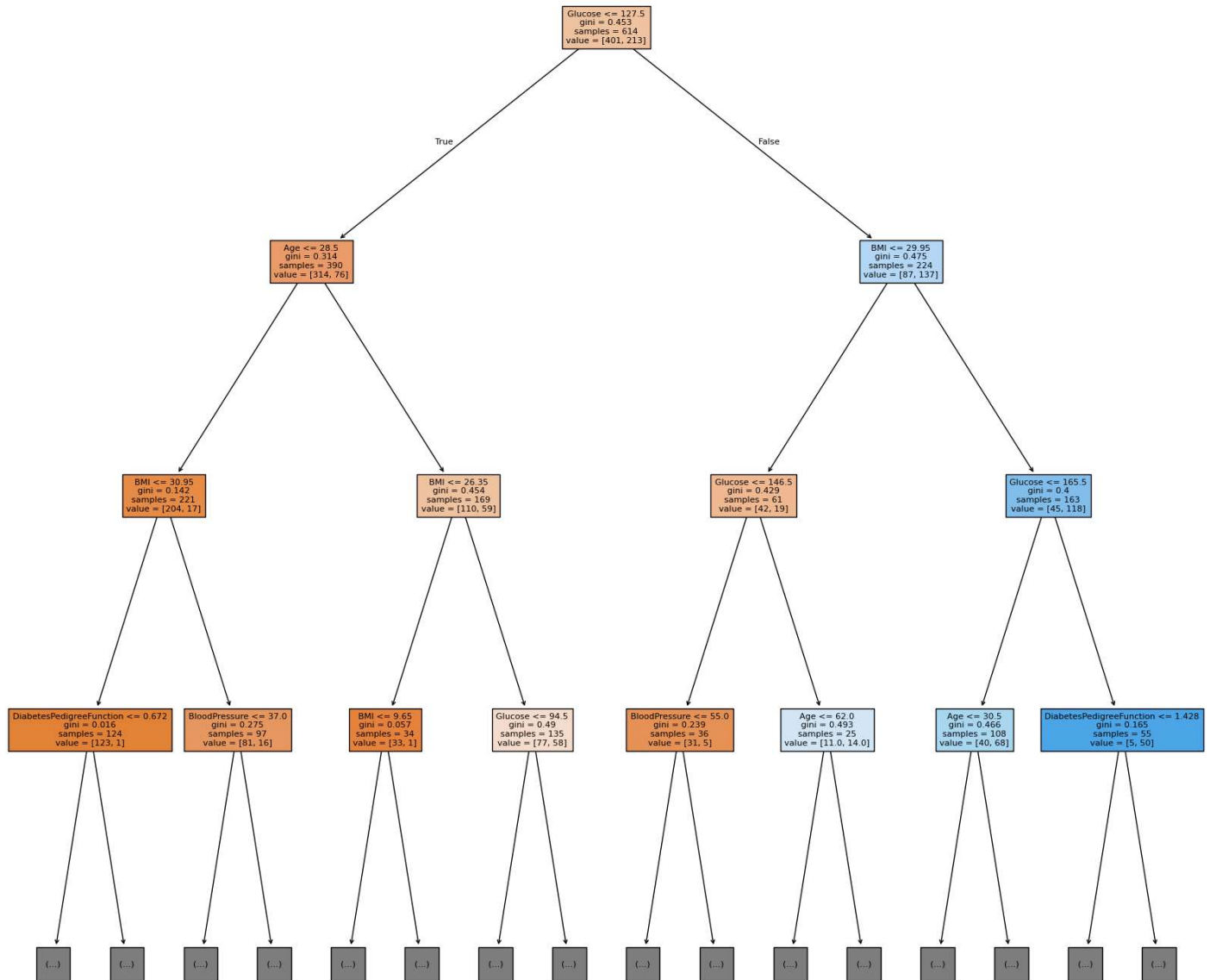
```
y_pred=model.predict(x_test)
print(accuracy_score(y_test,y_pred))
```



```
0.7467532467532467
```

```
from matplotlib import pyplot as plt
from sklearn.tree import plot_tree
```

```
plt.figure(figsize=(20,20))
plot_tree(model, filled=True, feature_names=x.columns, max_depth=3, fontsize=8)
plt.show()
```



```

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
classification_rep= classification_report(y_test, y_pred)
print("Classification Report:\n", classification_rep)

```

```

↗ Accuracy: 0.7467532467532467
Classification Report:

```

	precision	recall	f1-score	support
0	0.83	0.76	0.79	99
1	0.62	0.73	0.67	55
accuracy			0.75	154
macro avg	0.73	0.74	0.73	154
weighted avg	0.76	0.75	0.75	154

```

from sklearn.ensemble import GradientBoostingClassifier
model_gb=GradientBoostingClassifier(random_state=42)
model_gb.fit(x_train,y_train)

```

```

↗ GradientBoostingClassifier ⓘ ?
GradientBoostingClassifier(random state=42)

```

```

y_pred_gb=model_gb.predict(x_test)

```

```

accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy:{accuracy:.5f}' )
accuracy_gb = accuracy_score(y_test, y_pred_gb)
print(f'Accuracy Gradient Boosting:{accuracy_gb:.5f}')

```

```

↗ Accuracy:0.74675
Accuracy Gradient Boosting:0.74675

```