

```
import pandas as pd
dt=pd.read_csv("/content/diabetes_prediction_dataset.csv")
dt.head()
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	
0	Female	80.0	0	1	never	25.19	6.6	140	0	
1	Female	54.0	0	0	No Info	27.32	6.6	80	0	
2	Male	28.0	0	0	never	27.32	5.7	158	0	
3	Female	36.0	0	0	current	23.45	5.0	155	0	
4	Male	76.0	1	1	current	20.14	4.8	155	0	

Next steps:

[Generate code with dt](#)
[View recommended plots](#)
[New interactive sheet](#)

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in dt.columns:
    if dt[i].dtype=='object':
        dt[i]=le.fit_transform(dt[i])
dt.head()
```

	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes	
0	0	80.0	0	1	4	25.19	6.6	140	0	
1	0	54.0	0	0	0	27.32	6.6	80	0	
2	1	28.0	0	0	4	27.32	5.7	158	0	
3	0	36.0	0	0	1	23.45	5.0	155	0	
4	1	76.0	1	1	1	20.14	4.8	155	0	

Next steps:

[Generate code with dt](#)[View recommended plots](#)[New interactive sheet](#)

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x=dt.drop("diabetes",axis=1)
y=dt['diabetes']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
x_train_scaled=sc.fit_transform(x_train)
x_test_scaled=sc.transform(x_test)
from sklearn.tree import DecisionTreeClassifier
dc=DecisionTreeClassifier()
DecisionTreeClassifier()
```



▼ DecisionTreeClassifier ⓘ ?  
DecisionTreeClassifier()

```
dc.fit(x_train,y_train)
```



▼ DecisionTreeClassifier ⓘ ?  
DecisionTreeClassifier()

```
y_pred=dc.predict(x_test)
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```




	precision	recall	f1-score	support
0	0.98	0.97	0.97	18292
1	0.72	0.74	0.73	1708
accuracy			0.95	20000

macro avg	0.85	0.86	0.85	20000
weighted avg	0.95	0.95	0.95	20000

```
from sklearn.metrics import confusion_matrix
cn=confusion_matrix(y_pred,y_test)
cn
```

```
➡ array([[17789,  445],
        [  503, 1263]])
```

```
from sklearn.metrics import ConfusionMatrixDisplay
ConfusionMatrixDisplay(cn).plot()
```

 <sklearn.metrics.\_plot.confusion\_matrix.ConfusionMatrixDisplay at 0x7a0fda91c410>

