

ASSIGNMENT 6

```
[ ] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.naive_bayes import GaussianNB
    from sklearn.metrics import accuracy_score
    from sklearn.preprocessing import LabelEncoder
```

```
▶ data = pd.read_csv("tennis.csv")
```

```
[ ] data.head()
```

	outlook	temp	humidity	windy	play
0	sunny	hot	high	False	no
1	sunny	hot	high	True	no
2	overcast	hot	high	False	yes
3	rainy	mild	high	False	yes
4	rainy	cool	normal	False	yes

```
[ ] data.describe()
```

	outlook	temp	humidity	windy	play
count	14	14	14	14	14
unique	3	3	2	2	2
top	sunny	mild	high	False	yes
freq	5	6	7	8	9

```
[ ] encoders = {col: LabelEncoder().fit(data[col]) for col in data}
    for col, encoder in encoders.items():
        data[col] = encoder.transform(data[col])
```

```
[ ] x = data[["outlook","temp","humidity","windy"]]
    y = data["play"]
```

```
[ ] from sklearn.naive_bayes import GaussianNB
    gnb = GaussianNB()
```

```
[ ] x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=42)
```

```
[ ] gnb.fit(x_train,y_train)
```

▼ GaussianNB ⓘ ⓘ
GaussianNB()

```
[ ] y_predict = gnb.predict(x_test)
    print(f"Accuracy : {accuracy_score(y_test,y_predict)}")
```

Accuracy : 0.6

```
[ ] def Prediction(outlook,temp,humidity,windy):
    inputs = pd.DataFrame([[outlook,temp,humidity,windy]],columns=["outlook","temp","humidity","windy"])
    encoders = {col: LabelEncoder().fit(inputs[col]) for col in inputs}
    for col, encoder in encoders.items():
        inputs[col] = encoder.transform(inputs[col])
    Prediction = gnb.predict(inputs)
    return 'yes' if Prediction[0]==1 else 'No'
```

```
[ ] print("play",Prediction('rainy','cool','high','True'))
```

play yes