▼ CF066-LAB3-ML

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.n

```
import pandas as pd
dataset = pd.read_csv('/content/drive/My Drive/Dataset2.csv');
dataset.head(7)
```

₽		Outlook	Temp	Wind	Humidity	Class
	0	R	Н	F	Normal	N
	1	R	Н	Т	High	N
	2	0	Н	F	Normal	Υ
	3	S	M	F	Normal	Ν
	4	S	М	F	Normal	Υ
	5	S	M	Т	Low	Υ
	6	0	С	Т	Normal	Υ

Content of Dataset

```
from sklearn import preprocessing
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
le = preprocessing.LabelEncoder()
outlook_encoded = le.fit_transform(dataset['Outlook'])
Outlook_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("Outlook:" , outlook_encoded)

temp_encoded = le.fit_transform(dataset['Temp'])
Temperature_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("Temp:" , temp_encoded)

wind_encoded = le.fit_transform(dataset['Wind'])
Wind_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("Wind:" , wind_encoded)

humidity_encoded = le.fit_transform(dataset['Humidity'])
Humidity_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("Humidity:" , humidity_encoded)
```

```
class_encoded = le.fit_transform(dataset['Class'])
Play_name_mapping = dict(zip(le.classes_, le.transform(le.classes_)))
print("Class:" , class_encoded)

print("\n\n")
print("Weather mapping :-- " ,Outlook_name_mapping)
print("\n")
print("Temerature mapping :--" ,Temperature_name_mapping)
print("\n")
print("Humidity mapping :--" ,Humidity_name_mapping)
print("\n")
print("Wind mapping :--" ,Wind_name_mapping)
print("\n")
print("Play mapping :--" ,Play_name_mapping)
```

```
C Outlook: [1 1 0 2 2 2 0 1 1 2 1 0 0 2]
Temp: [1 1 1 2 2 2 0 0 0 2 2 2 1 2]
Wind: [0 1 0 0 0 1 1 0 0 0 1 1 0 1]
Humidity: [2 0 2 2 2 1 2 2 1 0 0 1 2 2]
Class: [0 0 1 0 1 1 1 1 1 1 1 1 1 0]
Weather mapping :-- {'O': 0, 'R': 1, 'S': 2}

Temerature mapping :-- {'C': 0, 'H': 1, 'M': 2}

Humidity mapping :-- {'High': 0, 'Low': 1, 'Normal': 2}

Wind mapping :-- {'F': 0, 'T': 1}

Play mapping :-- {'N': 0, 'Y': 1}
```

Features

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, test_pred)
```

```
□→ array([[2]])
```

```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
precision = precision_score(y_test,test_pred)
recall = recall_score(y_test,test_pred)
print('precision: {}'.format(precision))
print('recall: {}'.format(recall))
```

```
precision: 1.0
recall: 1.0
```

```
play_predict = model.predict([[1,2,0,2],[2,0,1,0]])
print(play_predict)
```

[1 1]