

```
In [1]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
In [2]: import pandas as pd
dataset = pd.read_csv('/content/drive/My Drive/Dataset2.csv');
dataset.head(5)
```

```
Out[2]:
```

	Outlook	Temp	Wind	Humidity	Class
0	R	H	F	Normal	N
1	R	H	T	High	N
2	O	H	F	Normal	Y
3	S	M	F	Normal	N
4	S	M	F	Normal	Y

```
In [3]: from sklearn import preprocessing
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

```
In [4]: 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("Temperature mapping : " ,Temperature_name_mapping)
print("Humidity mapping : " ,Humidity_name_mapping)
print("Wind mapping : " ,Wind_name_mapping)
print("Play mapping : " ,Play_name_mapping)
```

```
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}
Temperature 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}
```

```
In [5]: features = tuple(zip(outlook_encoded, temp_encoded, wind_encoded, humidity_encoded))
print("Features : " ,features)
```

```
Features : ((1, 1, 0, 2), (1, 1, 1, 0), (0, 1, 0, 2), (2, 2, 0, 2), (2, 2, 0, 2), (2, 2, 1, 1), (0, 0, 1, 2), (1, 0, 0, 2), (1, 0, 0, 1), (2, 2, 0, 0), (1, 2, 1, 0), (0, 2, 1, 1), (0, 1, 0, 2), (2, 2, 1, 2))
```

```
In [6]: x_train,x_test,y_train,y_test = train_test_split(features,class_encoded,test_size=0.2)
```

```
In [7]: model = MultinomialNB()
model.fit(x_train, y_train)
test_pred = model.predict(x_test)
print("Accuracy : ",metrics.accuracy_score(y_test,test_pred))
```

Accuracy : 1.0

```
In [8]: from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, test_pred)
```

Out[8]: array([[2]])

```
In [9]: 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

```
In [10]: play_predict = model.predict([[1,2,0,2],[2,0,1,0]])
print(play_predict)
```

[1 1]