Application 7

Supervised Machine Learning

Play Predictor using K Nearest Neighbour Algorithm

There is one data set of wether conditions.
That dataset contains information as wether and we have to decides whether to play or not.

Data set contains the target variable as Play which indicates whether to play or not.

Consider below Marvellous Infosystems Play Predictor Dataset as

Marvellous Infosystems Play Predictor			
	Wether	**Temperature**	**Play**
1	Sunny	Hot	No
2	Sunny	Hot	No
3	Overcast	Hot	Yes
4	Rainy	Mild	Yes
5	Rainy	Cool	Yes
6	Rainy	Cool	No
7	Overcast	Cool	Yes
8	Sunny	Mild	No
9	Sunny	Cool	Yes
10	Rainy	Mild	Yes

According to above dataset there are two features as
1. Wether
2. Temperature

We have two labels as
1. Yes
2. No

There are three types of different entries under Wether as
1. Sunny
2. Overcast
3. Rainy

There are three types of different entries under Temperature as
1. Hot
2. Cold
3. Mild

Consider below characteristics of Machine Learning Application :

Classifier :   **K Nearest Neighbour**
DataSet :   **Play Predictor Dataset**
Features :   **Whether , Temperature**
Labels :   **Yes, No**
Training Dataset :   **30 Entries**
Testing Dataset :   **1 Entry**

Consider below Machine Learning Application

```python
1   import numpy as np
2   import pandas as pd
3   from sklearn import preprocessing
4   from sklearn.neighbors import KNeighborsClassifier
5
6   def MarvellousPlayPredictor(data_path):
7
8       # Step 1 : Load data
9       data = pd.read_csv(data_path, index_col=0)
10
11      print("Size of Actual dataset", len(data))
12
13      # Step 2 : Clean, Prepare and manipulate data
14      feature_names = ['Whether', 'Temperature']
15
16      print("Names of Features", feature_names)
17
18      whether = data.Whether
19      Temperature = data.Temperature
20      play = data.Play
21
22      #creating labelEncoder
23      le = preprocessing.LabelEncoder()
24
25      # Converting string labels into numbers.
26      weather_encoded=le.fit_transform(whether)
27      print(weather_encoded)
28
29      # converting string labels into numbers
30      temp_encoded=le.fit_transform(Temperature)
31      label=le.fit_transform(play)
32
33      print(temp_encoded)
34
35      #combinig weather and temp into single listof tuples
36      features=list(zip(weather_encoded,temp_encoded))
37
38      # Step 3 : Train Data
39      model = KNeighborsClassifier(n_neighbors=3)
40
41      # Train the model using the training sets
42      model.fit(features,label)
43
44      # Step 4 : Test data
45      predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
46      print(predicted)
47
48   def main():
49       print("---- Marvellous Infosystems by Piyush Khairnar-----")
50
51       print("Machine Learning Application")
52
53       print("Play predictor application using K Nearest Knighbor algorithm")
54
55       MarvellousPlayPredictor("PlayPredictor.csv")
56
57   if __name__ == "__main__":
58       main()
59
```

Output of above Application :

```
(base) MacBook-Pro-de-MARVELLOUS:PlayPredictor_Line
ar_Regression marvellous$ python3 MarvellousPlayPre
dictorUsingKNN.py
---- Marvellous Infosystems by Piyush Khairnar-----
Machine Learning Application
Play predictor application using K Nearest Knighbor
 algorithm
Size of Actual dataset 30
Names of Features ['Whether', 'Temperature']
[2 2 0 1 1 1 0 2 2 1 2 0 0 1 1 1 1 0 2 2 1 2 2 2 0
1 1 0 2 2]
[1 1 1 2 0 0 0 2 0 2 2 2 1 2 2 0 0 0 2 0 2 2 1 1 1
2 0 0 2 0]
[1]
(base) MacBook-Pro-de-MARVELLOUS:PlayPredictor_Line
ar_Regression marvellous$ ▉
```