8. K-Nearest Neighbour (KNN) Classifier

November 15, 2022

1 K-Nearest Neighbour (KNN) Classifier

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
df=pd.read_csv('car_data.csv')
df.head()
```

[2]:
```
   userID  gender  age  salary  purchased
0     565    male   19   19000          0
1     566    male   26   35000          1
2     567  female   20   17000          0
3     568    male   25   32000          1
4     569  female   22   10000          0
```

[3]:
```
x= df.iloc[:, [2,3]].values  #independent variables-- age and salary
y= df.iloc[:, 4].values      #dependent variables-- purchased {0-not purchased;␣
 ↪1-purchased}
```

[9]:
```
# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.30,␣
 ↪random_state=0)
```

[10]:
```
#Fitting K-NN classifier to the training set
from sklearn.neighbors import KNeighborsClassifier
classifier= KNeighborsClassifier(n_neighbors=5, metric='euclidean', p=2 )
classifier.fit(x_train, y_train)
```

[10]:
```
KNeighborsClassifier(metric='euclidean')
```

[11]:
```
#Predicting the test set result
y_pred= classifier.predict(x_test)
```

[12]:
```
list(x_test)
```

[12]:

```
[array([   40, 62000]),
 array([   29, 49000]),
 array([   21, 23000]),
 array([   22, 10000]),
 array([   42, 33000]),
 array([   22, 17500]),
 array([   31, 39500]),
 array([   52, 45000]),
 array([   43, 37000]),
 array([   24, 23000]),
 array([   18, 18500]),
 array([   28, 47500])]
```

[13]:

```
y_pred.reshape(-1,1)
```

[13]:

```
array([[1],
       [1],
       [0],
       [0],
       [0],
       [0],
       [1],
       [1],
       [1],
       [0],
       [0],
       [1]])
```

[14]:

```
from sklearn.metrics import accuracy_score
print("Accuracy:",accuracy_score(y_test, y_pred)*100)
```

```
Accuracy: 83.33333333333334
```

[16]:

```
age=int(input("enter your age: "))
salary=int(input("enter your salary: "))

to_predict_list=[age,salary]
input_data= np.asarray(to_predict_list)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data.reshape(1,-1)

prediction = classifier.predict(input_data_reshaped)

if(prediction[0]==1):
    print("The person owns a car")
else:
```

```python
    print("The person doesn't have a car")
```

```
enter your age: 35
enter your salary: 52000
The person owns a car
```