ML Exp-6

Name: Prerna Sunil Jadhav
Sap ID: 60004220127
Batch: C22
Branch: Computer Engineering
Course: Machine learning

AIM: To implement K-nearest Neighbour.

THEORY: It is a supervised learning technique.
It assumes the similarity between the new case/ data and available cases and put the new case in the category that is most like the available category.
It can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on the underlying data.
Advantage: It is simple to implement, robust to noisy training data. More efficient if the data is large
Disadvantage: The computation cost is high because of calculating the distance between the data points for all training samples.

Q1]

ID	Height	Age	weight	Euclidean dist	Rank
1	5	45	77	7.0178	5
2	5.11	26	47	12.0063	8
3	5.6	30	55	8.0006	6
4	5.9	34	59	4.0199	3
5	4.8	40	72	2.1189	2
6	5.8	36	60	2.0223	1
7	5.3	19	40	19.0010	10
8	5.8	28	60	10.0044	7
9	5.5	23	45	15	9
10	5.6	32	58	6.0008	4
11	5.5	38	?		

For k=1,

weight of ID 11 = 60

for k=3,

weight of ID 11 = $\frac{60+72+59}{3}$ = 63.666 ≈ 64

For k=5,

weight of ID11 = $\frac{(60+72+59+58+77)}{5}$ = 65.2 ≈ 65

Q2

ID	Height	Weight	Class	Euclidean dist.	Rank
1	167	51	Underweight	6.7082	5
2	182	62	Normal	13	8
3	176	69	Normal	13.4164	9
4	173	64	Normal	7.6157	6
5	172	65	Normal	8.2462	7
6	174	56	Underweight	4.1231	4
7	169	58	Normal	1.4142	1
8	173	57	Normal	3	3
9	170	55	Normal	2	2
10	170	57	?		

For K = 1, K = 3, K = 5,
Class of ID 10 is Normal.

CONCLUSION: We see that in dataset 1 the algorithm provides different values for new case where as K value is different.
Dataset 2 despite the value of K, the new case category is Normal.
for the 3rd Dataset attached, the value (accuracy) for K = 1 is the highest, then the accuracy drops for K = 3 and again increases for K = 5.

- Name: Prerna Sunil Jadhav
- Sap ID: 60004220127
- Batch: C22
- Branch: Computer Engineering
- Course: Machine Learning
- Experiment 6: K-Nearest Neighbours

```
import numpy as np
from sklearn.preprocessing import LabelEncoder

def euclidean_distance(p1, p2):
  return np.sqrt(np.sum(np.square(p1 - p2)))

def predict_knn(dataset, new_data, k):
  distances = np.array([euclidean_distance(point[:-1], new_data) for point in dataset])
  sorted_data = np.array([dataset[i] for i in np.argsort(distances)])
  k_nearest_targets = sorted_data[:k, -1]
  prediction = np.mean(k_nearest_targets)
  return prediction

def knn_categorical(dataset, unknown, k):
    num_data = dataset[:, :-1].astype(float)
    cat_data = dataset[:, -1]
    label_encoder = LabelEncoder()
    cat_data_encoded = label_encoder.fit_transform(cat_data)

    unknown_num = unknown[:-1].astype(float)
    distances = np.sqrt(np.sum((num_data - unknown_num)**2, axis=1))
    nearest_indices = np.argsort(distances)[:k]
    nearest_labels = cat_data_encoded[nearest_indices]
    prediction = np.argmax(np.bincount(nearest_labels))
    return label_encoder.inverse_transform([prediction])[0]

dataset2 = np.array([
    [167, 51, 'Underweight'],
    [182, 62, 'Normal'],
    [176, 69, 'Normal'],
    [173, 64, 'Normal'],
    [172, 65, 'Normal'],
    [174, 56, 'Underweight'],
    [169, 58, 'Normal'],
    [173, 57, 'Normal'],
    [170, 55, 'Normal']
])

dataset = np.array([
    [5,  45, 77],
    [5.11, 26, 47],
    [5.6,  30, 55],
    [5.9,  34, 59],
    [4.8,  40, 72],
    [5.8,  36, 60],
    [5.3,  19, 40],
    [5.8,  28, 60],
    [5.5,  23, 45],
    [5.6,  32, 58]
])

# new_data = np.array([170, 57, None])  # None as a placeholder for numerical data
new_data = np.array([5.5, 38])
# print(type(dataset[:, -1]))
if (dataset[:, -1]).dtype != 'float64':
  for i in range(1,6,2):
    prediction_cat = knn_categorical(dataset, new_data, i,typeofknn='categorical')
    print("Predicted value for categorical data k:",i,":", prediction_cat)
else:
  for i in range(1,6,2):
    prediction = predict_knn(dataset, new_data, i)
    print("Predicted target value for numeric data k:",i,":", prediction)
```

```
new_data = np.array([170, 57, None])  # None as a placeholder for numerical data
print("\n")
if (dataset2[:, -1]).dtype != 'float64':
  for i in range(1,6,2):
    prediction_cat = knn_categorical(dataset2, new_data, i)
    print("Predicted value for categorical data k:",i,":", prediction_cat)
else:
  for i in range(1,6,2):
    prediction = predict_knn(dataset2, new_data, i)
    print("Predicted target value for numeric data k:",i,":", prediction)
```

```
Predicted target value for numeric data k: 1 : 60.0
Predicted target value for numeric data k: 3 : 63.666666666666664
Predicted target value for numeric data k: 5 : 65.2


Predicted value for categorical data k: 1 : Normal
Predicted value for categorical data k: 3 : Normal
Predicted value for categorical data k: 5 : Normal
```

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

import sklearn.metrics as metrics

data = pd.read_csv('../content/iphone_purchase_records.csv')
print(data.head())

data = data.drop('Gender',axis=1)

X = data.drop('Purchase Iphone', axis=1)
y = data['Purchase Iphone']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

for i in range(1,6,2):

  knn = KNeighborsClassifier(n_neighbors=i)

  knn.fit(X_train,y_train)

  y_pred_knn = knn.predict(X_test)

  score_knn = metrics.accuracy_score(y_test,knn.predict(X_test))
  print('Accuracy for k =',i,':{0:f}'.format(score_knn))
```

```
   Gender  Age  Salary  Purchase Iphone
0    Male   19   19000                0
1    Male   35   20000                0
2  Female   26   43000                0
3  Female   27   57000                0
4    Male   19   76000                0
Accuracy for k = 1 :0.840000
Accuracy for k = 3 :0.780000
Accuracy for k = 5 :0.830000
```