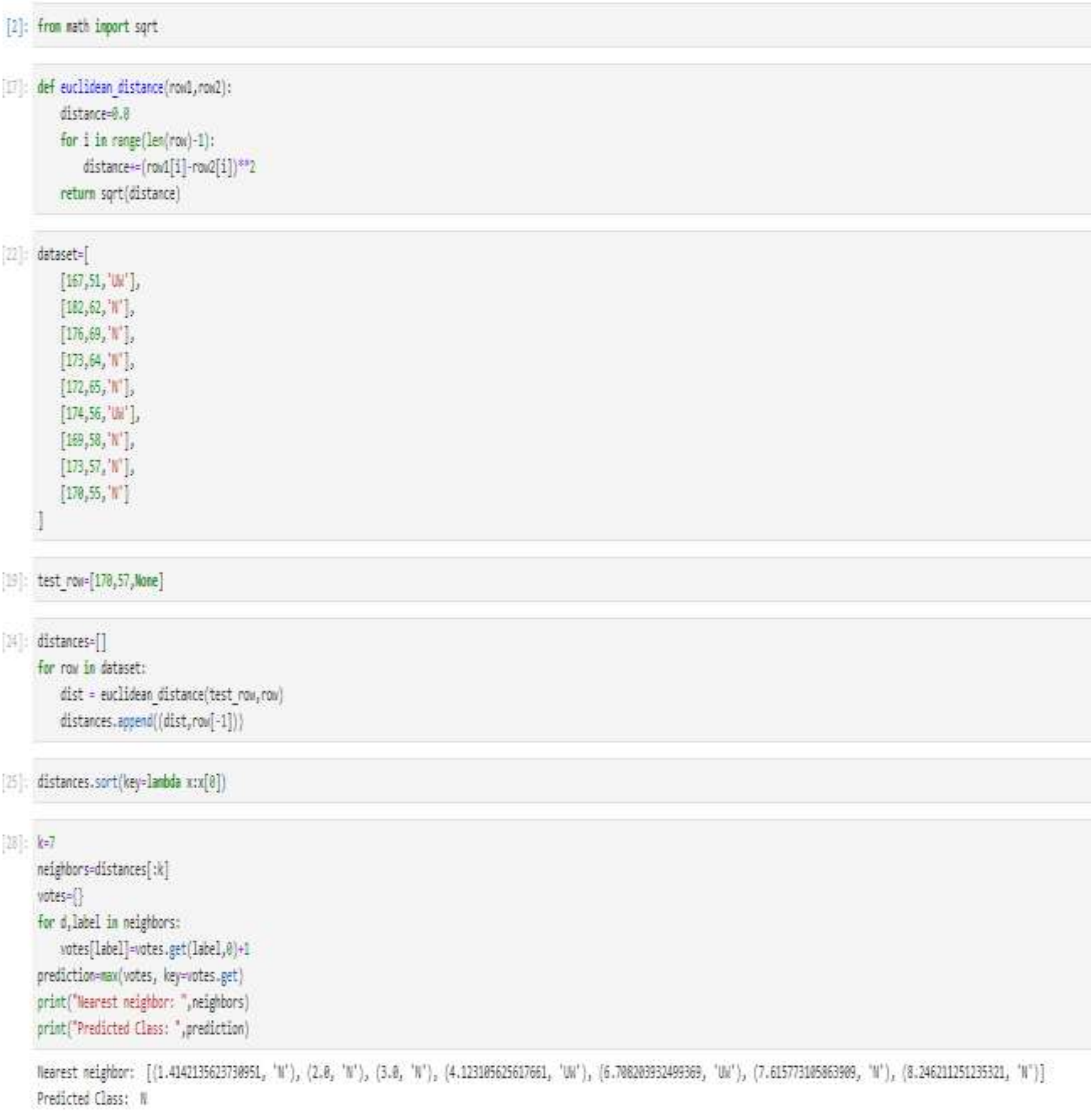
LAB ASSIGNMENT-5

(Debasmita Bagchi ,UG/02/BTCSEAIML/2023/065, Sec-E, Sem-6)

Build KNN Classification model for a given dataset. Vary the number of k values as follows and compare the results: i. 1 ii. 3 iii. 5 iv. 7 v. 11



```
[2]: from math import sqrt
```

```
[17]: def euclidean_distance(row1,row2):
          distance=0.0
          for i in range(len(row)-1):
              distance+=(row1[i]-row2[i])**2
          return sqrt(distance)
```

```
[22]: dataset=[
          [167,51,'UW'],
          [182,62,'N'],
          [176,69,'N'],
          [173,64,'N'],
          [172,65,'N'],
          [174,56,'UW'],
          [169,58,'N'],
          [173,57,'N'],
          [170,55,'N']
      ]
```

```
[19]: test_row=[170,57,None]
```

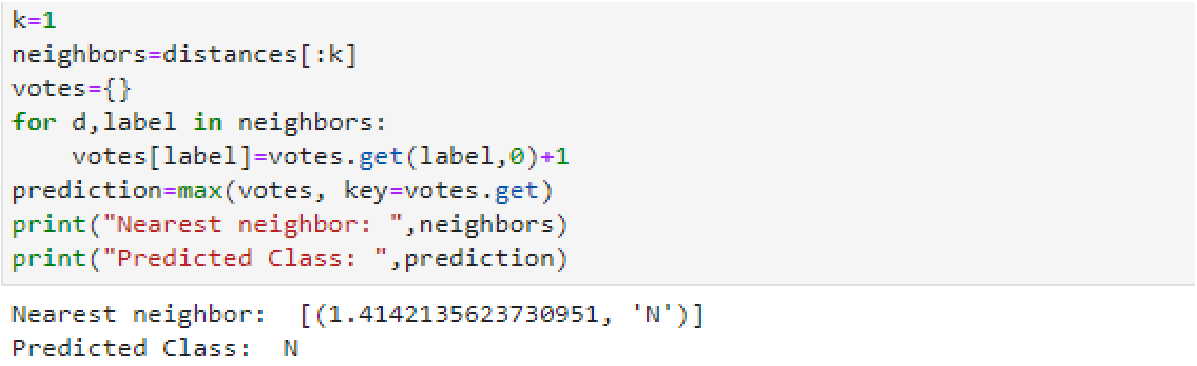
```
[24]: distances=[]
      for row in dataset:
          dist = euclidean_distance(test_row,row)
          distances.append((dist,row[-1]))
```

```
[25]: distances.sort(key=lambda x:x[0])
```

```
[28]: k=7
      neighbors=distances[:k]
      votes={}
      for d,label in neighbors:
          votes[label]=votes.get(label,0)+1
      prediction=max(votes, key=votes.get)
      print("Nearest neighbor: ",neighbors)
      print("Predicted Class: ",prediction)
```

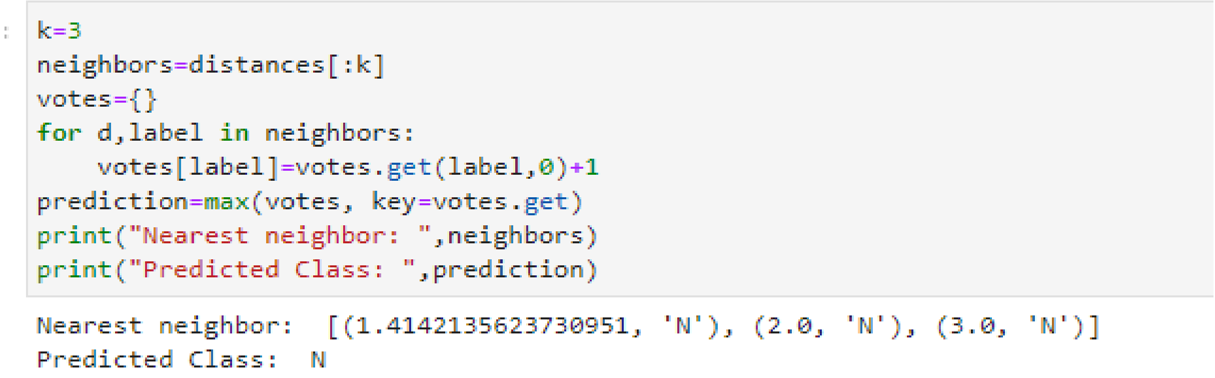
```
Nearest neighbor:  [(1.4142135623730951, 'N'), (2.0, 'N'), (3.0, 'N'), (4.123105625617661, 'UW'), (6.708203932499369, 'UW'), (7.615773105863909, 'N'), (8.246211251235321, 'N')]
Predicted Class:  N
```

(Debasmita Bagchi ,UG/02/BTCSEAIML/2023/065, Sec-E, Sem-6)



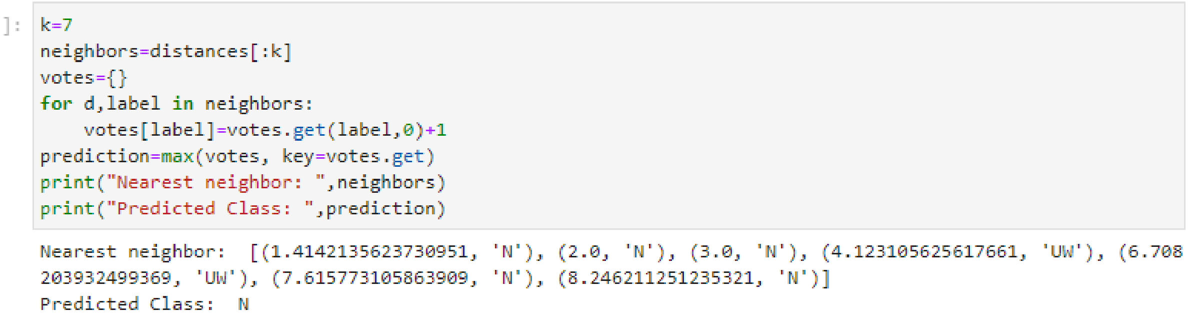
```
k=1
neighbors=distances[:k]
votes={}
for d,label in neighbors:
    votes[label]=votes.get(label,0)+1
prediction=max(votes, key=votes.get)
print("Nearest neighbor: ",neighbors)
print("Predicted Class: ",prediction)
```

```
Nearest neighbor:  [(1.4142135623730951, 'N')]
Predicted Class:  N
```



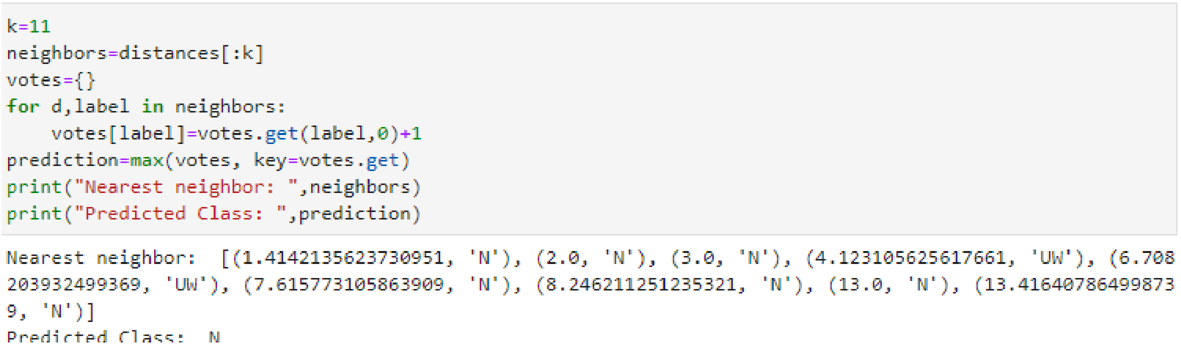
```
k=3
neighbors=distances[:k]
votes={}
for d,label in neighbors:
    votes[label]=votes.get(label,0)+1
prediction=max(votes, key=votes.get)
print("Nearest neighbor: ",neighbors)
print("Predicted Class: ",prediction)
```

```
Nearest neighbor:  [(1.4142135623730951, 'N'), (2.0, 'N'), (3.0, 'N')]
Predicted Class:  N
```



```
k=7
neighbors=distances[:k]
votes={}
for d,label in neighbors:
    votes[label]=votes.get(label,0)+1
prediction=max(votes, key=votes.get)
print("Nearest neighbor: ",neighbors)
print("Predicted Class: ",prediction)
```

```
Nearest neighbor:  [(1.4142135623730951, 'N'), (2.0, 'N'), (3.0, 'N'), (4.123105625617661, 'UW'), (6.708
203932499369, 'UW'), (7.615773105863909, 'N'), (8.246211251235321, 'N')]
Predicted Class:  N
```



```
k=11
neighbors=distances[:k]
votes={}
for d,label in neighbors:
    votes[label]=votes.get(label,0)+1
prediction=max(votes, key=votes.get)
print("Nearest neighbor: ",neighbors)
print("Predicted Class: ",prediction)
```

```
Nearest neighbor:  [(1.4142135623730951, 'N'), (2.0, 'N'), (3.0, 'N'), (4.123105625617661, 'UW'), (6.708
203932499369, 'UW'), (7.615773105863909, 'N'), (8.246211251235321, 'N'), (13.0, 'N'), (13.41640786499873
9, 'N')]
Predicted Class:  N
```