

HW1_question5_KNN

March 12, 2021

```
[2]: import pandas as pd

# read the iris dataset which is csv format
col_names = ["sepal_length", "sepal_width", "petal_length", "petal_width", "species"]
iris = pd.read_csv("iris.data", header=None, names=col_names)
iris.head()
```

```
[2]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

```
[3]: # map iris class name to number
iris_class = {'Iris-setosa':0, 'Iris-versicolor':1, 'Iris-virginica':2}
iris['species_tag'] = [iris_class[i] for i in iris.species]
iris.tail()
```

```
[3]:
```

	sepal_length	sepal_width	petal_length	petal_width	species \
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

	species_tag
145	2
146	2
147	2
148	2
149	2

```
[4]: #split data into attributes and target/label
iris_attrs = iris.drop(['species', 'species_tag'], axis=1)
iris_labels = iris.species_tag
```

```
[5]: print(iris_attr)
```

	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
..
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

[150 rows x 4 columns]

```
[14]: from sklearn.model_selection import train_test_split
      from sklearn.neighbors import KNeighborsClassifier

      # avg of score

      best_fit = -1
      best_avg = -1

      # k from 1 to 11
      for j in range(1, 12):
          avg = 0
          # run 10 times
          for i in range(10):
              # split data into training and testing sets
              train_data, test_data, train_label, test_label = \
→train_test_split(iris_attr, iris_labels,

              # set 5 neighbors of knn
              knn = KNeighborsClassifier(n_neighbors = j)
              # fit the model on the training data
              knn.fit(train_data, train_label)
              # see how the model performs
              avg = avg + knn.score(test_data, test_label)
          if (best_avg < avg):
              best_fit = j
              best_avg = avg

      # average accuracy
      print('best_avg',best_avg/10,'best_fit',best_fit)
```

```
best_avg 0.9755555555555556 best_fit 7
```

```
[15]: # predicted label  
print(knn.predict(test_data))
```

```
[0 1 1 2 2 1 2 1 2 1 1 1 2 1 1 0 2 1 0 2 0 2 1 1 2 2 0 1 0 0 0 1 2 1 0 0 0  
 1 0 2 1 2 0 0 1]
```

```
[16]: # actual label  
print(test_label[0:5])
```

```
46      0  
75      1  
65      1  
101     2  
114     2  
Name: species_tag, dtype: int64
```

```
[ ]:
```