KNN_iris_crossval

July 5, 2022

[ ]:
```
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn.datasets import load_iris

iris = load_iris()
```

[ ]:
```
df = pd.DataFrame(data = iris.data, columns = iris.feature_names)
df['target'] = iris.target
df
```

[ ]:
```
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
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

     target
0         0
1         0
2         0
3         0
4         0
..      …
145       2
146       2
147       2
148       2
149       2

[150 rows x 5 columns]
```

[ ]:
```
df.columns = ['sl', 'sw', 'pl', 'pw', 'label']
df
```

[ ]:
```
      sl   sw   pl   pw  label
0    5.1  3.5  1.4  0.2      0
1    4.9  3.0  1.4  0.2      0
2    4.7  3.2  1.3  0.2      0
3    4.6  3.1  1.5  0.2      0
4    5.0  3.6  1.4  0.2      0
..   …   …   …   …    …
145  6.7  3.0  5.2  2.3      2
146  6.3  2.5  5.0  1.9      2
147  6.5  3.0  5.2  2.0      2
148  6.2  3.4  5.4  2.3      2
149  5.9  3.0  5.1  1.8      2

[150 rows x 5 columns]
```

[ ]:
```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df, test_size = 0.2)
```

[ ]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
```

[ ]:
```
max_k_range = train.shape[0]//2
k_list = []
for i in range(3, max_k_range, 2):
    k_list.append(i)
```

[ ]:
```
cross_validation_scores = []
x_train = train[['pl', 'pw']]
y_train = train[['label']]
```

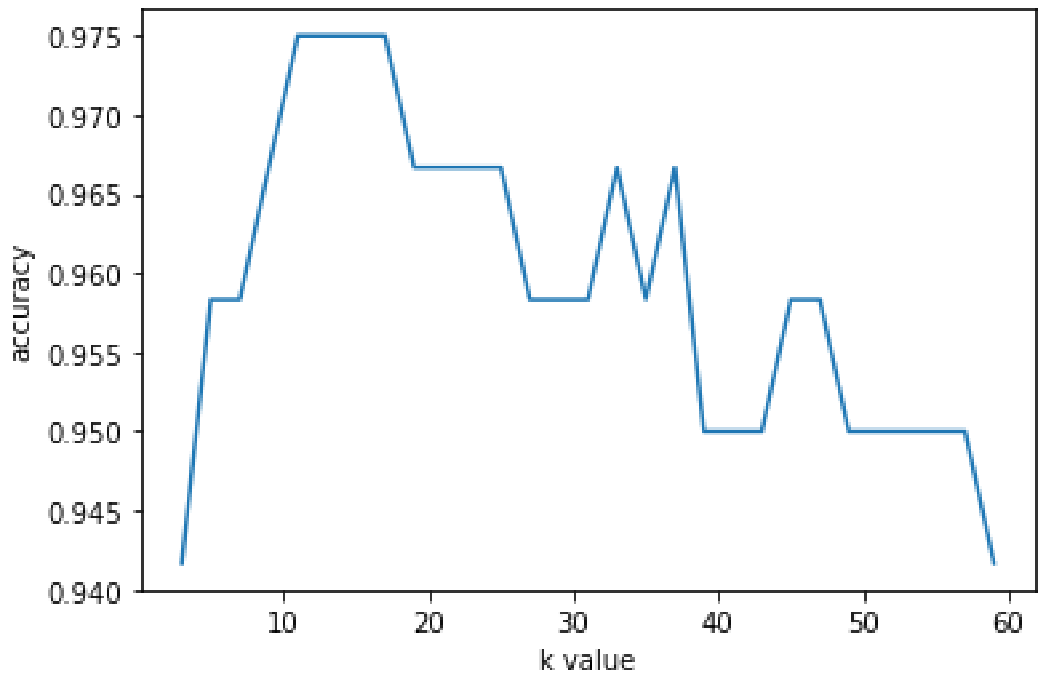
[ ]:
```
for k in k_list:
    knn = KNeighborsClassifier(n_neighbors = k)
    scores = cross_val_score(knn, x_train, y_train.values.ravel(), cv = 10,␣
 ↪scoring = 'accuracy')
    cross_validation_scores.append(scores.mean())
```

[ ]:
```
cross_validation_scores
```

[ ]:
```
[0.9416666666666667,
 0.9583333333333333,
 0.9583333333333333,
 0.9666666666666666,
 0.975,
 0.975,
 0.975,
 0.975,
 0.9666666666666666,
 0.9666666666666666,
 0.9666666666666666,
 0.9666666666666666,
 0.9583333333333334,
 0.9583333333333334,
 0.9583333333333333,
 0.9666666666666666,
 0.9583333333333333,
 0.9666666666666666,
 0.95,
 0.95,
 0.95,
 0.9583333333333333,
 0.9583333333333333,
 0.95,
 0.95,
 0.95,
 0.95,
 0.95,
 0.9416666666666667]
```

[ ]:
```
import matplotlib.pyplot as plt
```

[ ]:
```
plt.plot(k_list, cross_validation_scores)
plt.xlabel('k value')
plt.ylabel('accuracy')
plt.show()
```



```
[ ]: best_k = k_list[cross_validation_scores.index(max(cross_validation_scores))]
     best_k
```

```
[ ]: 11
```

```
[ ]: knn = KNeighborsClassifier(n_neighbors = best_k)
```

```
[ ]: knn.fit(x_train, y_train.values.ravel())
```

```
[ ]: KNeighborsClassifier(n_neighbors=11)
```

```
[ ]: x_test = test[['pl', 'pw']]
     y_test = test[['label']]
```

```
[ ]: predictions = knn.predict(x_test)
     predictions
```

```
[ ]: array([2, 1, 0, 2, 0, 0, 2, 2, 0, 2, 0, 1, 0, 2, 2, 0, 0, 1, 1, 2, 2, 0,
            2, 1, 1, 1, 0, 1, 1, 0])
```

```
[ ]: from sklearn.metrics import accuracy_score
```

```
[ ]: print("accuracy is " + str(accuracy_score(y_test.values.ravel(),predictions)))
```

```
accuracy is 0.9666666666666667
```

```
[ ]: comparison = pd.DataFrame(
         {'pred' : predictions, 'truth' : y_test.values.ravel()}
     )
     comparison
```

```
[ ]:     pred  truth
     0      2      2
     1      1      1
     2      0      0
     3      2      1
     4      0      0
     5      0      0
     6      2      2
     7      2      2
     8      0      0
     9      2      2
     10     0      0
     11     1      1
     12     0      0
     13     2      2
     14     2      2
     15     0      0
     16     0      0
     17     1      1
     18     1      1
     19     2      2
     20     2      2
     21     0      0
     22     2      2
     23     1      1
     24     1      1
     25     1      1
     26     0      0
     27     1      1
     28     1      1
     29     0      0
```