knn-wine-classification

January 2, 2024

[2]:
```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
```

[3]:
```
wine = datasets.load_wine()
```

[4]:
```
wine
```

[4]:
```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, …, 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, …, 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, …, 1.030e+00, 3.170e+00,
         1.185e+03],
        …,
        [1.327e+01, 4.280e+00, 2.260e+00, …, 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, …, 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, …, 6.100e-01, 1.600e+00,
         5.600e+02]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2]),
 'frame': None,
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': '.. _wine_dataset:\n\nWine recognition
dataset\n------------------------\n\n**Data Set Characteristics:**\n\n
:Number of Instances: 178\n    :Number of Attributes: 13 numeric, predictive
```

attributes and the class\n   :Attribute Information:\n \t\t- Alcohol\n \t\t- Malic acid\n \t\t- Ash\n\t\t- Alcalinity of ash \n \t\t- Magnesium\n\t\t- Total phenols\n \t\t- Flavanoids\n \t\t- Nonflavanoid phenols\n \t\t- Proanthocyanins\n\t\t- Color intensity\n \t\t- Hue\n \t\t- OD280/OD315 of diluted wines\n \t\t- Proline\n\n   - class:\n   - class_0\n - class_1\n   - class_2\n\t\t\n   :Summary Statistics:\n   \n ============================= ==== ===== ======= =====\n Min   Max   Mean   SD\n   ============================= ==== ===== ======= =====\n   Alcohol: 11.0 14.8   13.0   0.8\n   Malic Acid: 0.74 5.80   2.34 1.12\n   Ash: 1.36 3.23   2.36 0.27\n   Alcalinity of Ash: 10.6 30.0   19.5 3.3\n   Magnesium: 70.0 162.0   99.7 14.3\n   Total Phenols: 0.98 3.88   2.29 0.63\n   Flavanoids: 0.34 5.08   2.03 1.00\n   Nonflavanoid Phenols: 0.13 0.66   0.36 0.12\n   Proanthocyanins: 0.41 3.58   1.59 0.57\n   Colour Intensity: 1.3 13.0   5.1   2.3\n   Hue: 0.48 1.71   0.96 0.23\n   OD280/OD315 of diluted wines: 1.27 4.00   2.61 0.71\n   Proline: 278 1680   746   315\n ============================= ==== ===== ======= =====\n\n   :Missing Attribute Values: None\n   :Class Distribution: class_0 (59), class_1 (71), class_2 (48)\n   :Creator: R.A. Fisher\n   :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n   :Date: July, 1988\n\nThis is a copy of UCI ML Wine recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data\n\nThe data is the results of a chemical analysis of wines grown in the same\nregion in Italy by three different cultivators. There are thirteen different\nmeasurements taken for different constituents found in the three types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVUS - \nAn Extendible Package for Data Exploration, Classification and Correlation. \nInstitute of Pharmaceutical and Food Analysis and Technologies,\nVia Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Repository\n[https://archive.ics.uci.edu/ml]. Irvine, CA: University of California,\nSchool of Information and Computer Science. \n\n|details-start|\n**References**\n|details-split|\n\n(1) S. Aeberhard, D. Coomans and O. de Vel, \nComparison of Classifiers in High Dimensional Settings, \nTech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of \nMathematics and Statistics, James Cook University of North Queensland. \n(Also submitted to Technometrics). \n\nThe data was used with many others for comparing various \nclassifiers. The classes are separable, though only RDA \nhas achieved 100% correct classification. \n(RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)) \n(All results using the leave-one-out technique) \n\n(2) S. Aeberhard, D. Coomans and O. de Vel, \n"THE CLASSIFICATION PERFORMANCE OF RDA" \nTech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \nMathematics and Statistics, James Cook University of North Queensland. \n(Also submitted to Journal of Chemometrics).\n\n|details-end|',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',

```
 'alcalinity_of_ash',
 'magnesium',
 'total_phenols',
 'flavanoids',
 'nonflavanoid_phenols',
 'proanthocyanins',
 'color_intensity',
 'hue',
 'od280/od315_of_diluted_wines',
 'proline']}
```

[5]:
```
df = pd.DataFrame(wine["data"], columns = wine["feature_names"])
```

[6]:
```
df
```

[6]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	\
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	
..	…	…	…	…	…	…	
173	13.71	5.65	2.45	20.5	95.0	1.68	
174	13.40	3.91	2.48	23.0	102.0	1.80	
175	13.27	4.28	2.26	20.0	120.0	1.59	
176	13.17	2.59	2.37	20.0	120.0	1.65	
177	14.13	4.10	2.74	24.5	96.0	2.05	

	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	\
0	3.06	0.28	2.29	5.64	1.04	
1	2.76	0.26	1.28	4.38	1.05	
2	3.24	0.30	2.81	5.68	1.03	
3	3.49	0.24	2.18	7.80	0.86	
4	2.69	0.39	1.82	4.32	1.04	
..	…	…	…	…	…	
173	0.61	0.52	1.06	7.70	0.64	
174	0.75	0.43	1.41	7.30	0.70	
175	0.69	0.43	1.35	10.20	0.59	
176	0.68	0.53	1.46	9.30	0.60	
177	0.76	0.56	1.35	9.20	0.61	

	od280/od315_of_diluted_wines	proline
0	3.92	1065.0
1	3.40	1050.0
2	3.17	1185.0
3	3.45	1480.0
4	2.93	735.0

```
..                              …       …
173                          1.74   740.0
174                          1.56   750.0
175                          1.56   835.0
176                          1.62   840.0
177                          1.60   560.0

[178 rows x 13 columns]
```

[7]:
```
df["target"] = wine["target"]
```

[8]:
```
df.head()
```

[8]:
```
   alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0    14.23        1.71  2.43               15.6      127.0           2.80
1    13.20        1.78  2.14               11.2      100.0           2.65
2    13.16        2.36  2.67               18.6      101.0           2.80
3    14.37        1.95  2.50               16.8      113.0           3.85
4    13.24        2.59  2.87               21.0      118.0           2.80

   flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0        3.06                  0.28             2.29             5.64  1.04
1        2.76                  0.26             1.28             4.38  1.05
2        3.24                  0.30             2.81             5.68  1.03
3        3.49                  0.24             2.18             7.80  0.86
4        2.69                  0.39             1.82             4.32  1.04

   od280/od315_of_diluted_wines  proline  target
0                          3.92   1065.0       0
1                          3.40   1050.0       0
2                          3.17   1185.0       0
3                          3.45   1480.0       0
4                          2.93    735.0       0
```

[9]:
```
df.shape
```

[9]:
```
(178, 14)
```

[10]:
```
df.isna().sum()
```

[10]:
```
alcohol                         0
malic_acid                      0
ash                             0
alcalinity_of_ash               0
magnesium                       0
total_phenols                   0
flavanoids                      0
```

```
nonflavanoid_phenols            0
proanthocyanins                 0
color_intensity                 0
hue                             0
od280/od315_of_diluted_wines    0
proline                         0
target                          0
dtype: int64
```

[11]:
```
X = df
y = X.pop("target")
```

[12]:
```
X.head()
```

[12]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	\
0	14.23	1.71	2.43	15.6	127.0	2.80	
1	13.20	1.78	2.14	11.2	100.0	2.65	
2	13.16	2.36	2.67	18.6	101.0	2.80	
3	14.37	1.95	2.50	16.8	113.0	3.85	
4	13.24	2.59	2.87	21.0	118.0	2.80	

	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	\
0	3.06	0.28	2.29	5.64	1.04	
1	2.76	0.26	1.28	4.38	1.05	
2	3.24	0.30	2.81	5.68	1.03	
3	3.49	0.24	2.18	7.80	0.86	
4	2.69	0.39	1.82	4.32	1.04	

	od280/od315_of_diluted_wines	proline
0	3.92	1065.0
1	3.40	1050.0
2	3.17	1185.0
3	3.45	1480.0
4	2.93	735.0

[13]:
```
X.shape
```

[13]:
```
(178, 13)
```

[14]:
```
y.head()
```

[14]:
```
0    0
1    0
2    0
3    0
4    0
Name: target, dtype: int32
```

[15]:
```
y.unique()
```

[15]:
```
array([0, 1, 2])
```

[16]:
```
# Split the data into training and validation sets
X_train, X_val, y_train, val_y = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=56)
```

[17]:
```
X_train.shape
```

[17]:
```
(142, 13)
```

[18]:
```
from sklearn.neighbors import KNeighborsClassifier
```

[19]:
```
knn = KNeighborsClassifier(n_neighbors = 3)
```

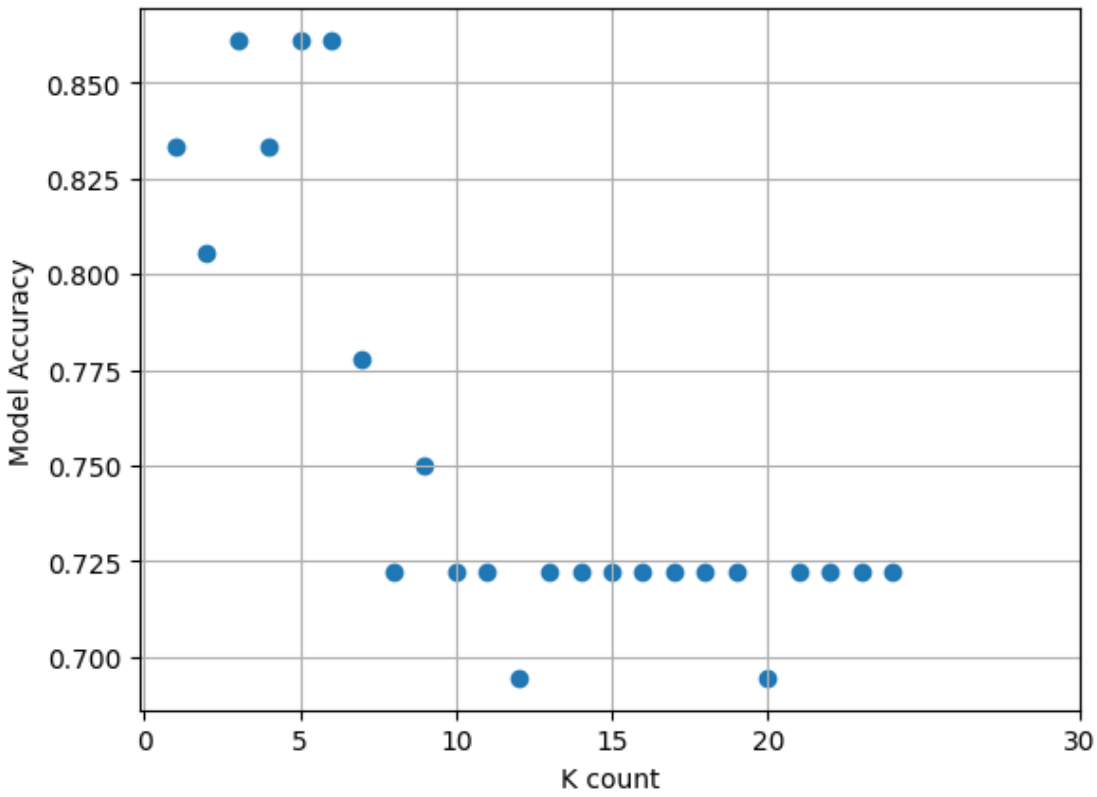
[20]:
```
knn.fit(X_train , y_train)
```

[20]:
```
KNeighborsClassifier(n_neighbors=3)
```

[21]:
```
knn.score(X_val ,val_y)
```

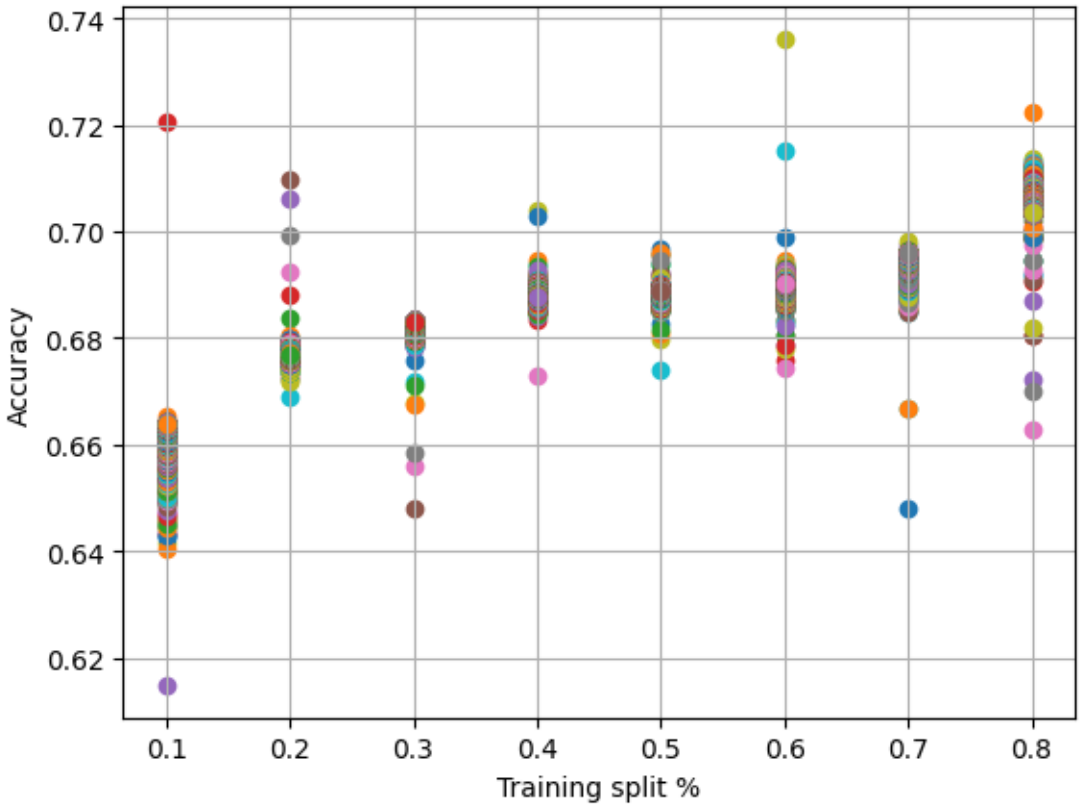
[21]:
```
0.8611111111111112
```

[22]:
```
import matplotlib.pyplot as plt
k_range = range (1, 25)
scores = []
for k  in k_range:
    knn = knn = KNeighborsClassifier(k)
    knn.fit(X_train , y_train)
    scores.append(knn.score(X_val ,val_y))
plt.figure()
plt.xlabel("K count")
plt.ylabel("Model Accuracy")
plt.scatter(k_range, scores)
plt.grid()
plt.xticks([0,5,10,15,20,30])
plt.show()
```



[23]:
```
test_sizes = [0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1]
knn = KNeighborsClassifier(n_neighbors = 3)
plt.figure()
plt.xlabel("Training split %")
plt.ylabel("Accuracy")
plt.grid()
for test_size in test_sizes:
    scores = []

    for i in range (1, 1000):
        X_train, X_val, y_train, val_y = train_test_split(X, y,␣
 ↪test_size=1-test_size)
        knn.fit(X_train , y_train)
        scores.append(knn.score(X_val ,val_y))
        plt.scatter(test_size, np.mean(scores))
plt.show()
```



[24]:
```
prediction = knn.predict(X_val)
```

[25]:
```
from sklearn.metrics import accuracy_score
```

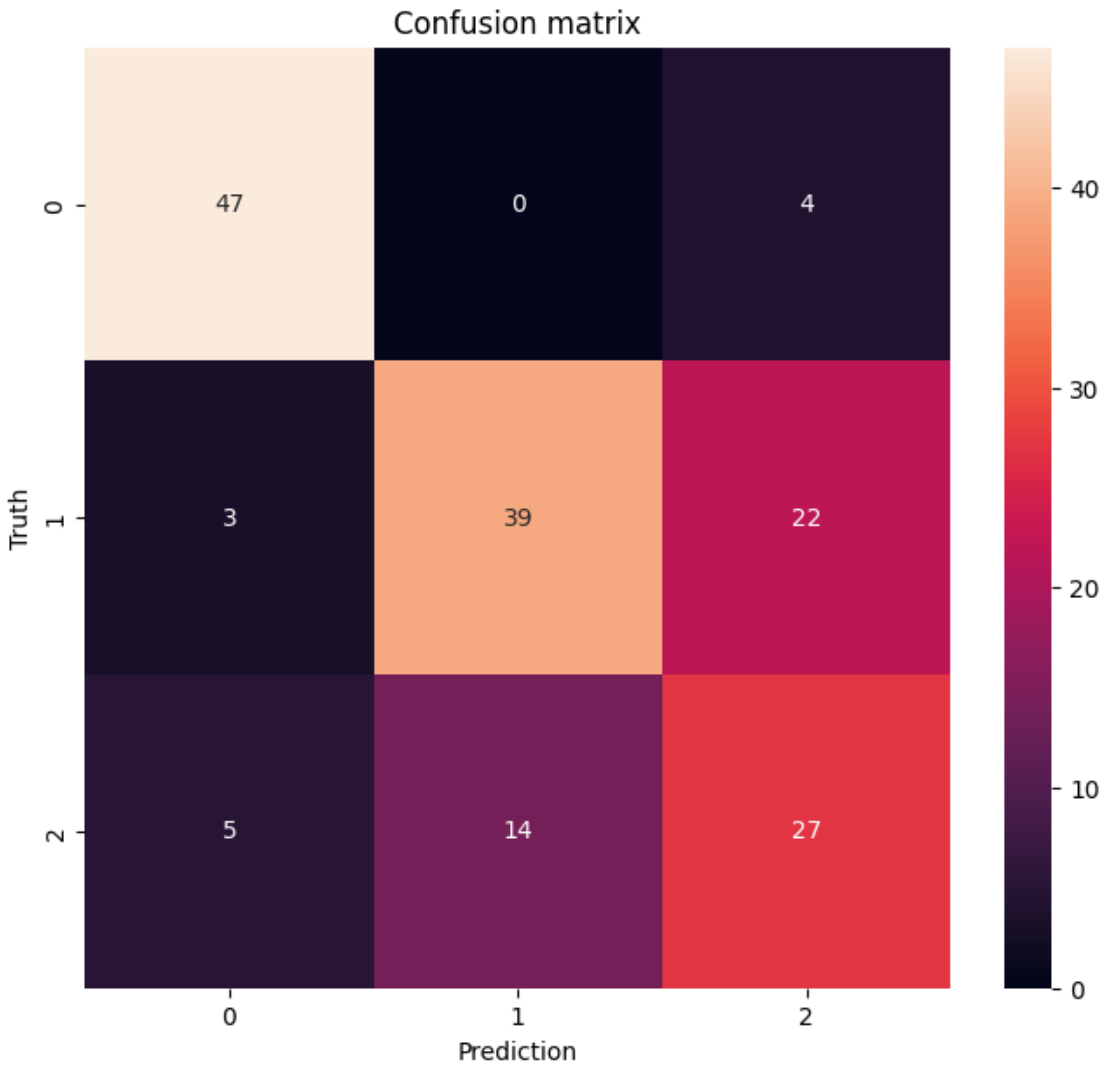
[26]:
```
from sklearn.metrics import confusion_matrix
```

[27]:
```
cm = confusion_matrix(val_y,prediction)
```

[28]:
```
cm
```

[28]:
```
array([[47,  0,  4],
       [ 3, 39, 22],
       [ 5, 14, 27]], dtype=int64)
```

[29]:
```
import seaborn as sns
plt.figure(figsize = (8,7))
sns.heatmap(cm ,annot = True)
plt.xlabel("Prediction")
plt.ylabel("Truth")
plt.title("Confusion matrix")
plt.show()
```


Confusion matrix
47
0
4
3
39
22
5
14
27
Truth
Prediction
0
1
2
40
30
20
10
0

[ ]: