iris-flower

July 20, 2024

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import psutil
```

[2]:
```
from sklearn.cluster import KMeans
```

[3]:
```
df = pd.read_csv("IRIS.csv")
```

[4]:
```
df.head()
```

[4]:

	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

[5]:
```
df['species'],categories= pd.factorize(df['species'])
```

[6]:
```
df.head()
```

[6]:

	sepal_length	sepal_width	petal_length	petal_width	species
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

[7]:
```
df.describe
```

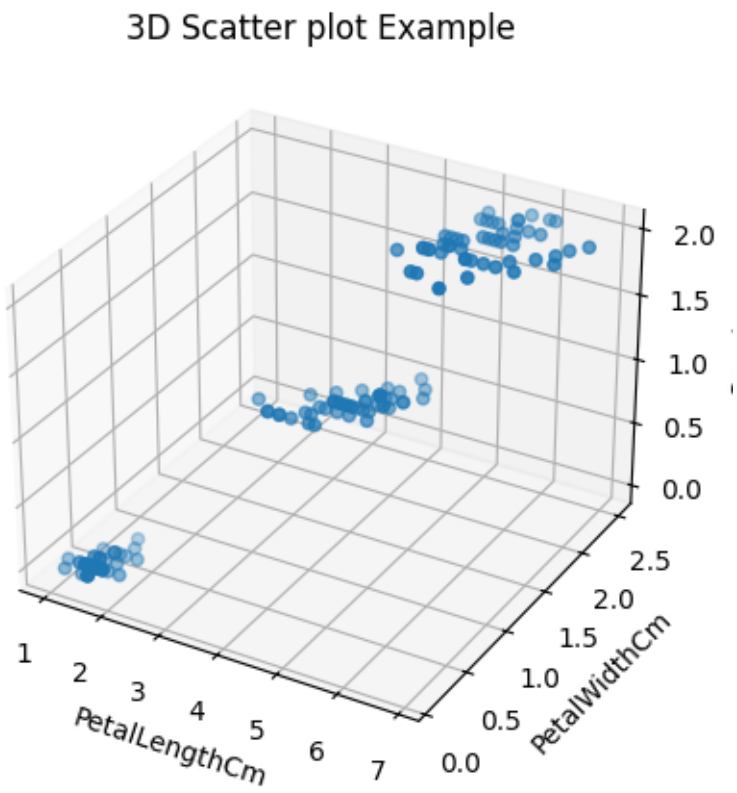
[7]:
```
<bound method NDFrame.describe of      sepal_length  sepal_width  petal_length
petal_width  species
0             5.1          3.5           1.4          0.2        0
1             4.9          3.0           1.4          0.2        0
```

```
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]>
```

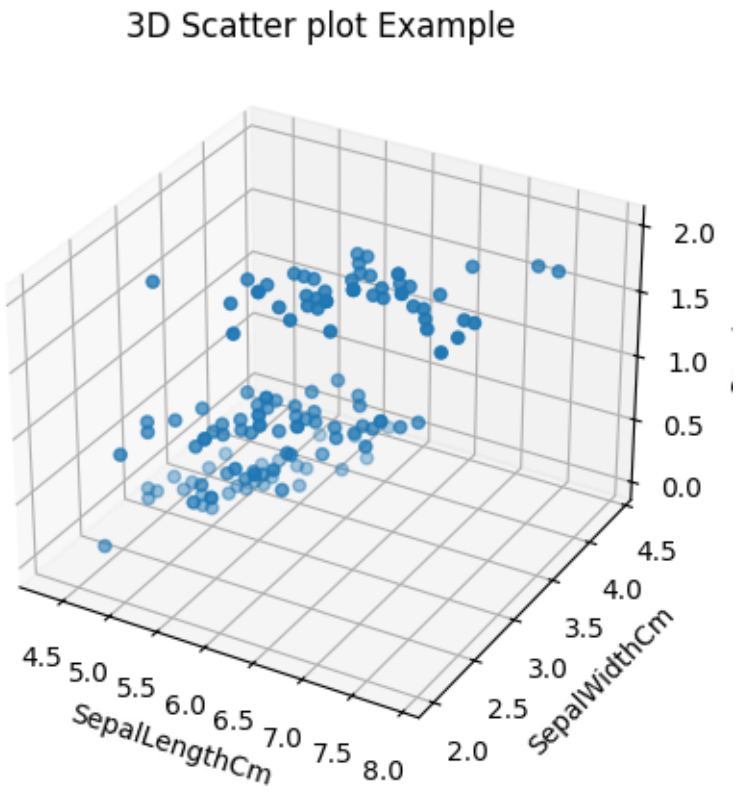
[8]:
```
df.isnull().sum()
```

[8]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

[9]:
```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax=fig.add_subplot(111,projection='3d')
ax.scatter(df.petal_length, df.petal_width, df.species)

ax.set_xlabel('PetalLengthCm')
ax.set_ylabel('PetalWidthCm')
ax.set_zlabel('Species')
plt.title('3D Scatter plot Example')
plt.show()
```

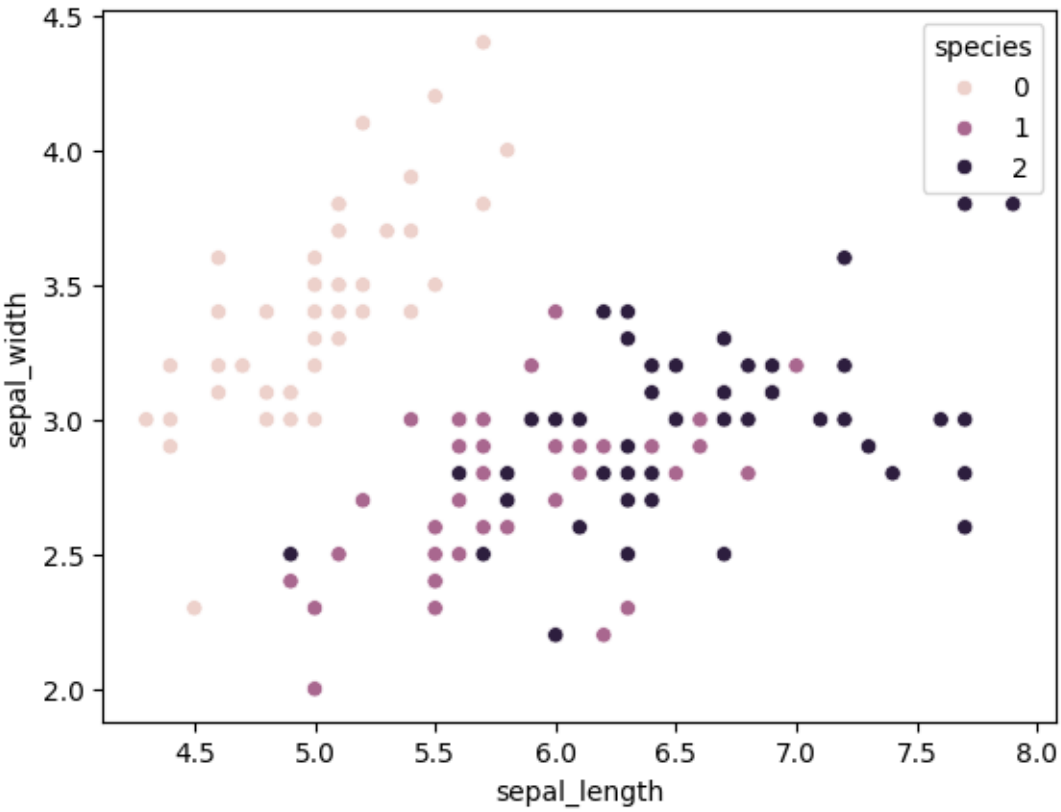


[10]:
```
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax=fig.add_subplot(111,projection='3d')
ax.scatter(df.sepal_length, df.sepal_width, df.species)

ax.set_xlabel('SepalLengthCm')
ax.set_ylabel('SepalWidthCm')
ax.set_zlabel('Species')
plt.title('3D Scatter plot Example')
plt.show()
```



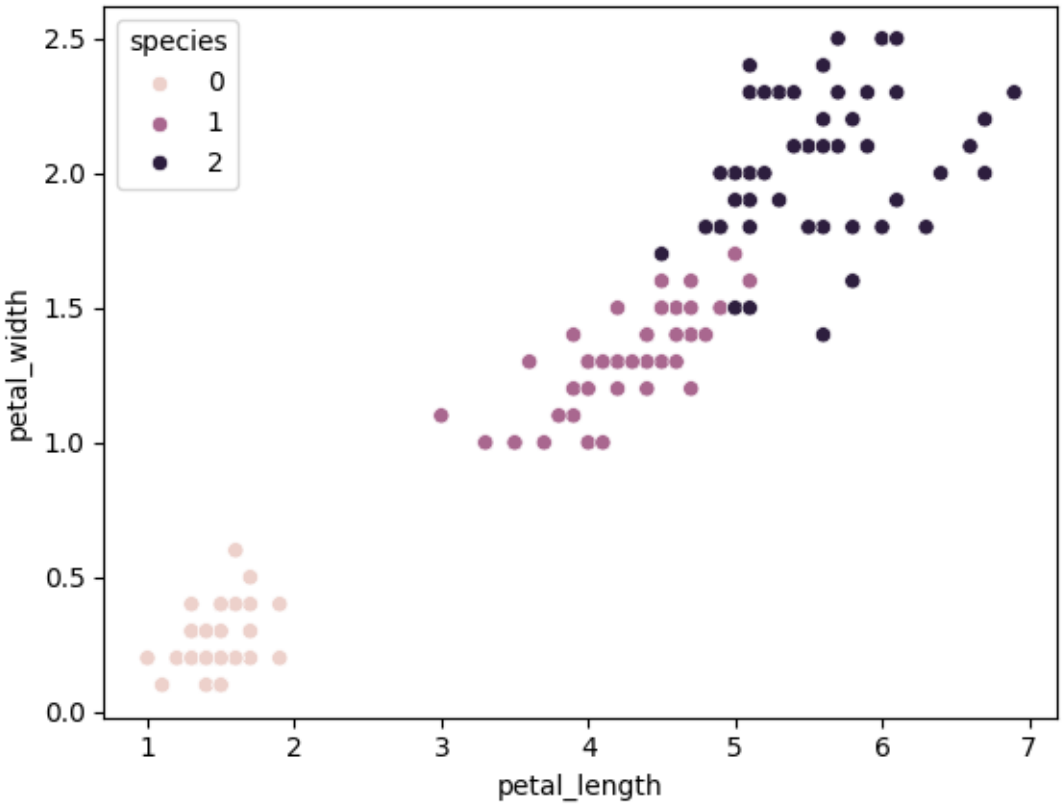
[11]:
```
sns.scatterplot(data=df, x='sepal_length',y='sepal_width',hue='species')
```

[11]: <Axes: xlabel='sepal_length', ylabel='sepal_width'>



[12]:
```
sns.scatterplot(data=df, x='petal_length',y='petal_width',hue='species')
```

[12]:
```
<Axes: xlabel='petal_length', ylabel='petal_width'>
```



Applying Elbow Technique

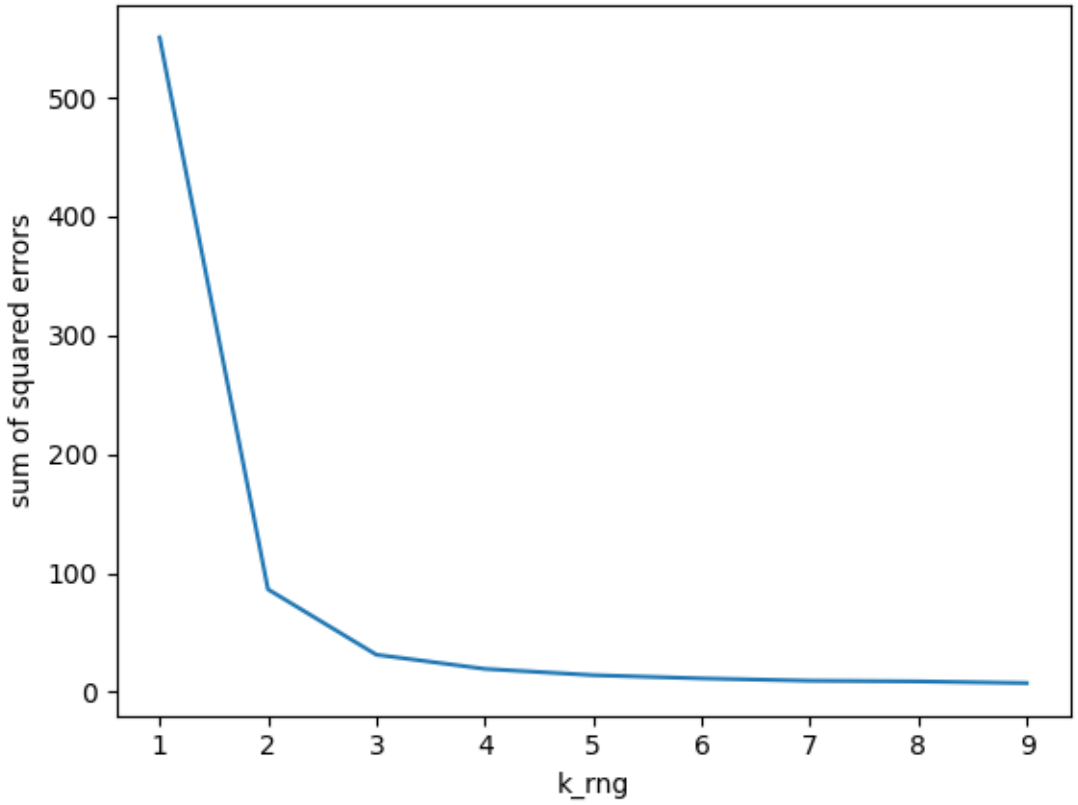
[13]:
```
k_rng = range(1,10)
sse=[]

for k in k_rng:
  km = KMeans(n_clusters=k)
  km.fit(df[['petal_length','petal_width']])
  sse.append(km.inertia_)
```

```
d:\Path\Lib\site-packages\joblib\externals\loky\backend\context.py:136:
UserWarning: Could not find the number of physical cores for the following
reason:
[WinError 2] The system cannot find the file specified
Returning the number of logical cores instead. You can silence this warning by
setting LOKY_MAX_CPU_COUNT to the number of cores you want to use.
  warnings.warn(
  File "d:\Path\Lib\site-packages\joblib\externals\loky\backend\context.py",
line 257, in _count_physical_cores
    cpu_info = subprocess.run(
               ^^^^^^^^^^^^^^^

  File "d:\Path\Lib\subprocess.py", line 548, in run
```

```
  with Popen(*popenargs, **kwargs) as process:
       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "d:\Path\Lib\subprocess.py", line 1026, in __init__
  self._execute_child(args, executable, preexec_fn, close_fds,
File "d:\Path\Lib\subprocess.py", line 1538, in _execute_child
  hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

[14]:
```
plt.xlabel('k_rng')
plt.ylabel('sum of squared errors')
plt.plot(k_rng,sse)
```

[14]:
```
[<matplotlib.lines.Line2D at 0x171ff5182f0>]
```



[15]:
```
km=KMeans(n_clusters=3,random_state=0,)
y_predicted=km.fit_predict(df[['petal_length','petal_width']])
y_predicted
```

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

[16]:
```
df['cluster']=y_predicted
df.head(150)
```

[16]:

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

[150 rows x 6 columns]

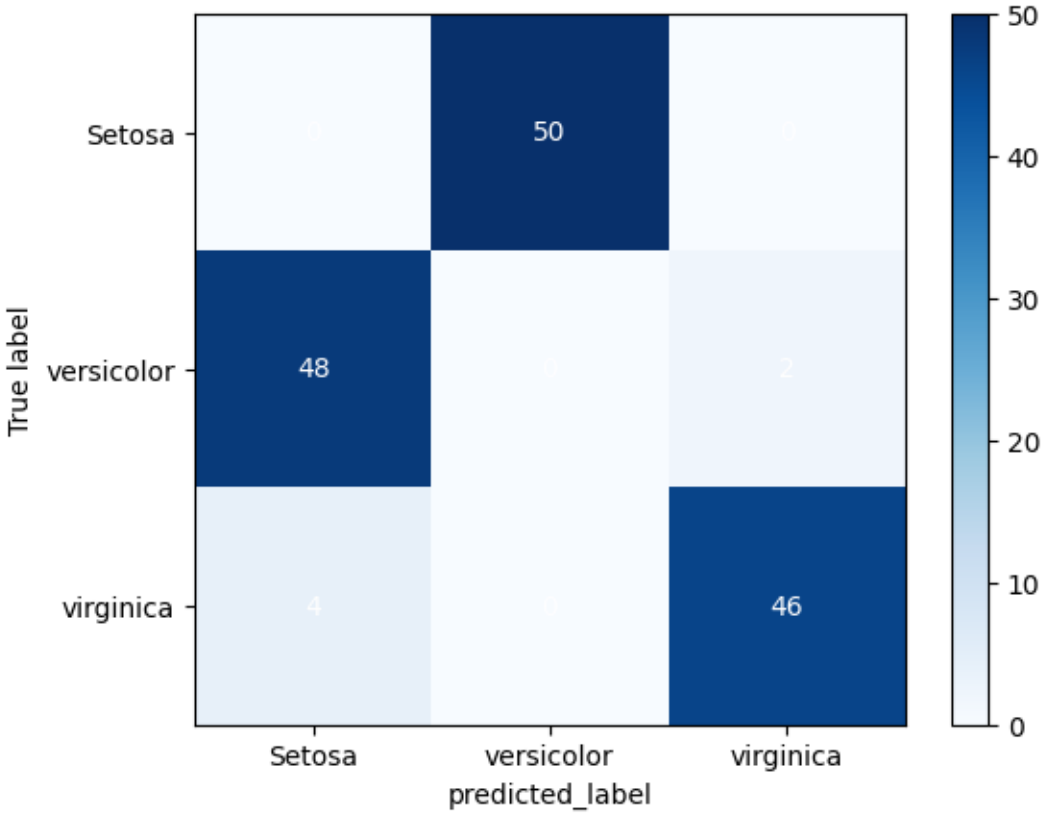
Accuracy measure

[17]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(df.species,df.cluster)
cm
```

[17]:
```
array([[ 0, 50,  0],
       [48,  0,  2],
       [ 4,  0, 46]], dtype=int64)
```

[18]:
```
true_labels = df.species
predicted_labels=df.cluster

cm=confusion_matrix(true_labels,predicted_labels)
class_labels=['Setosa','versicolor','virginica']

# plt confusin matrix
plt.imshow(cm,interpolation='nearest',cmap=plt.cm.Blues)
plt.colorbar()
tick_marks=np.arange(len(class_labels))
plt.xticks(tick_marks,class_labels)
plt.yticks(tick_marks,class_labels)

for i in range(len(class_labels)):
```

```
  for j in range(len(class_labels)):
    plt.text(j,i,str(cm[i][j]),ha='center',va='center',color='white')

plt.xlabel("predicted_label")
plt.ylabel("True label")
plt.show()
```



[ ]: