```
In [9]: import numpy as np
        import pandas as pd
        from sklearn.cluster import KMeans as km
        import matplotlib.pyplot as plt
        import seaborn as sns
```

```
In [18]: df=pd.read_csv(r"C:\Users\lahar\Downloads\archive (7)\IRIS.csv")
         df.head()
```

Out[18]:

	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

```
In [19]: df['species'],categories = pd.factorize(df['species'])
         df.head()
```

Out[19]:

	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

```
In [20]: df.describe
```

```
Out[20]: <bound method NDFrame.describe of      sepal_length  sepal_width  petal_le
         ngth  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]>
```

In [21]: `df.isna().sum()`

Out[21]:
```
sepal_length     0
sepal_width      0
petal_length     0
petal_width      0
species          0
dtype: int64
```

In [22]:
```python
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()
```
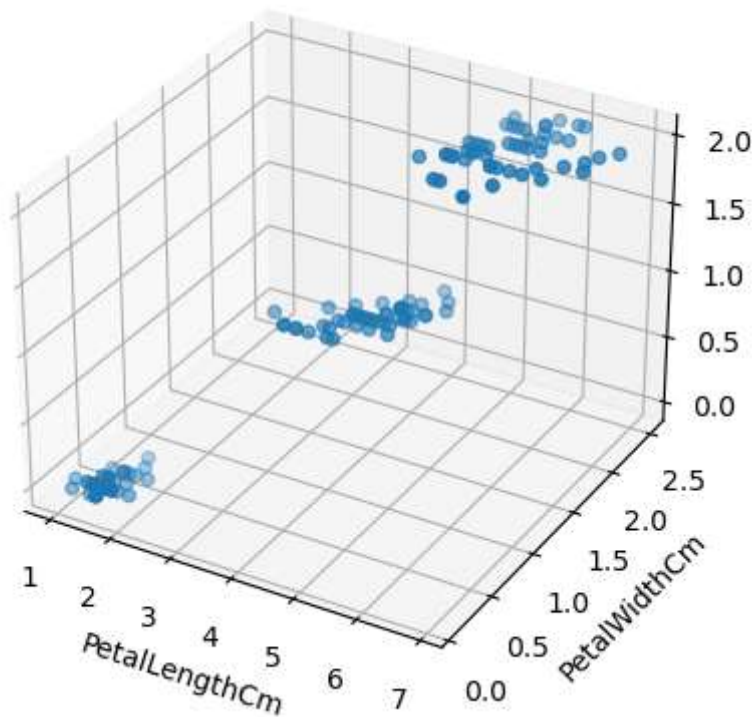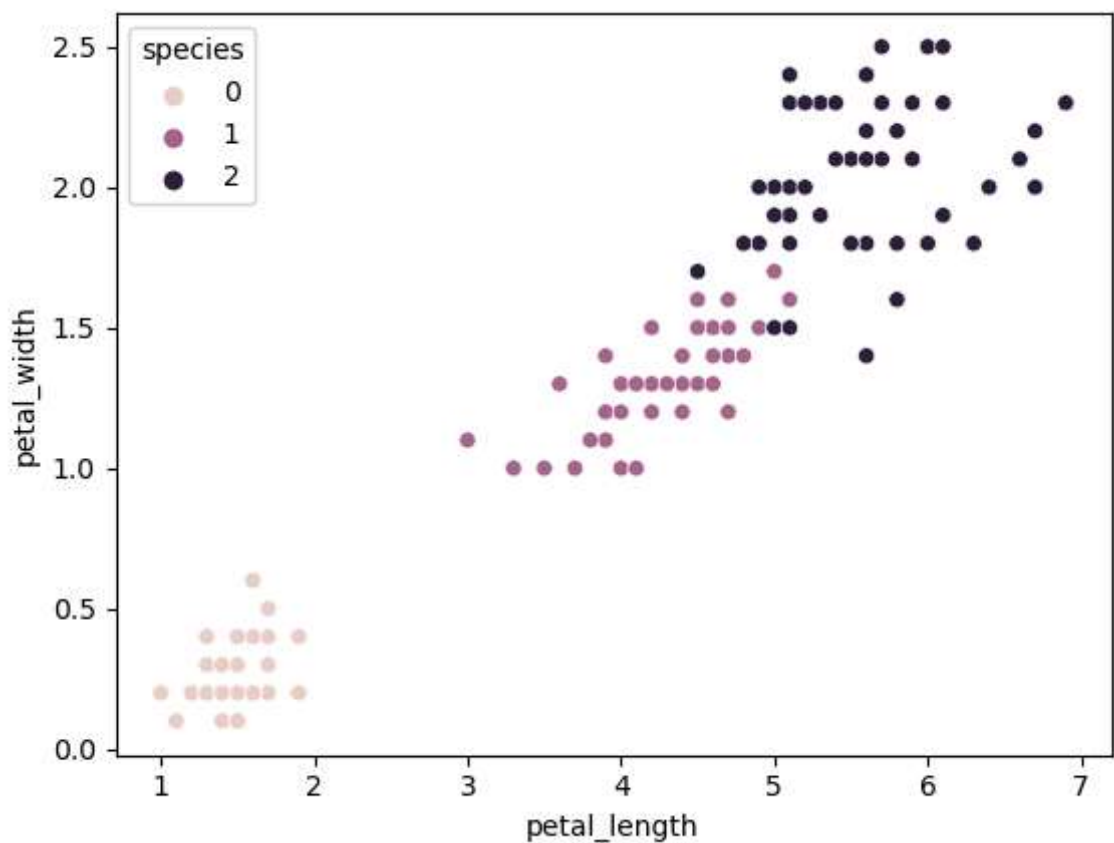
In [23]: `sns.scatterplot(data=df, x="sepal_length", y="sepal_width", hue="species");`
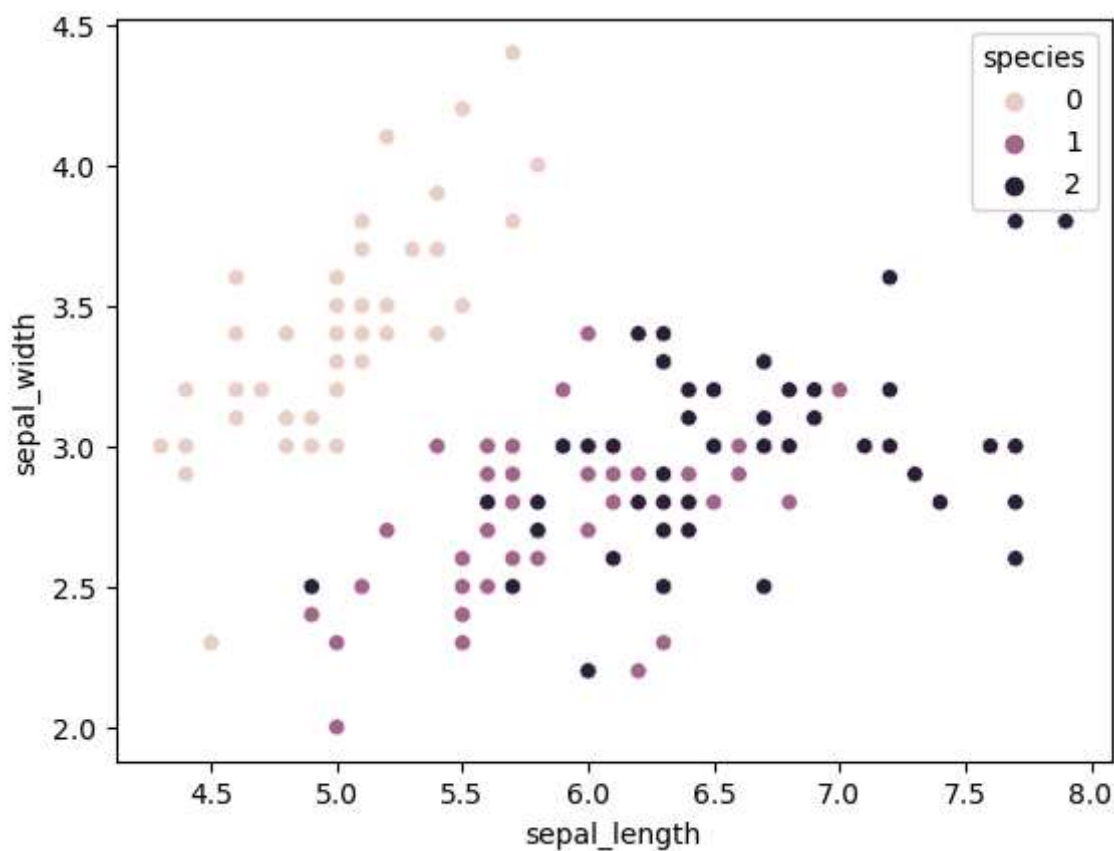
In [24]: `sns.scatterplot(data=df, x="petal_length", y="petal_width", hue="species");`

```
In [29]: k_rrg = range(1,10)
         sse=[]

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

```
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
```

```
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```
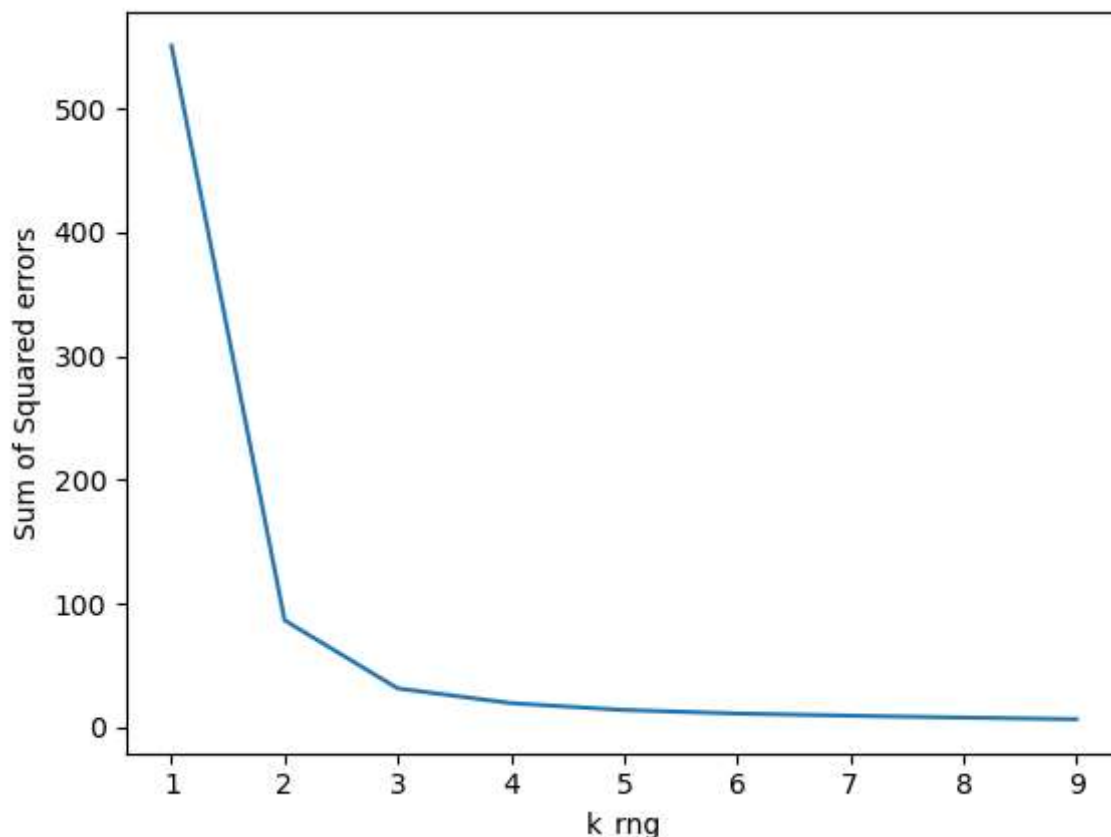
In [30]:
```
sse
```

Out[30]:
```
[550.6434666666668,
 86.40394533571005,
 31.387758974358974,
 19.49352336396547,
 13.933308757908756,
 11.10362366260479,
 9.344727985739752,
 7.836495238095239,
 6.496659206692711]
```

In [33]:
```python
plt.xlabel('k_rng')
plt.ylabel("Sum of Squared errors")
plt.plot(k_rrg, sse)
```

Out[33]: [<matplotlib.lines.Line2D at 0x24f97d09a50>]

In [34]:
```python
km = KMeans(n_clusters=3,random_state=0,)
y_predicted = km.fit_predict(df[['petal_length','petal_width']])
y_predicted
```

```
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:141
2: FutureWarning: The default value of `n_init` will change from 10 to 'au
to' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
C:\Users\lahar\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:143
6: UserWarning: KMeans is known to have a memory leak on Windows with MKL,
when there are less chunks than available threads. You can avoid it by set
ting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[34]:
```
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, 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, 0, 2, 0, 0, 0, 0, 0, 2, 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])
```

In [37]:
```python
df['cluster']=y_predicted
df.head(150)
```

Out[37]:

	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 × 6 columns

In [41]:
```python
from  sklearn.metrics import confusion_matrix
cm =confusion_matrix(df.species, df.cluster)
cm
```

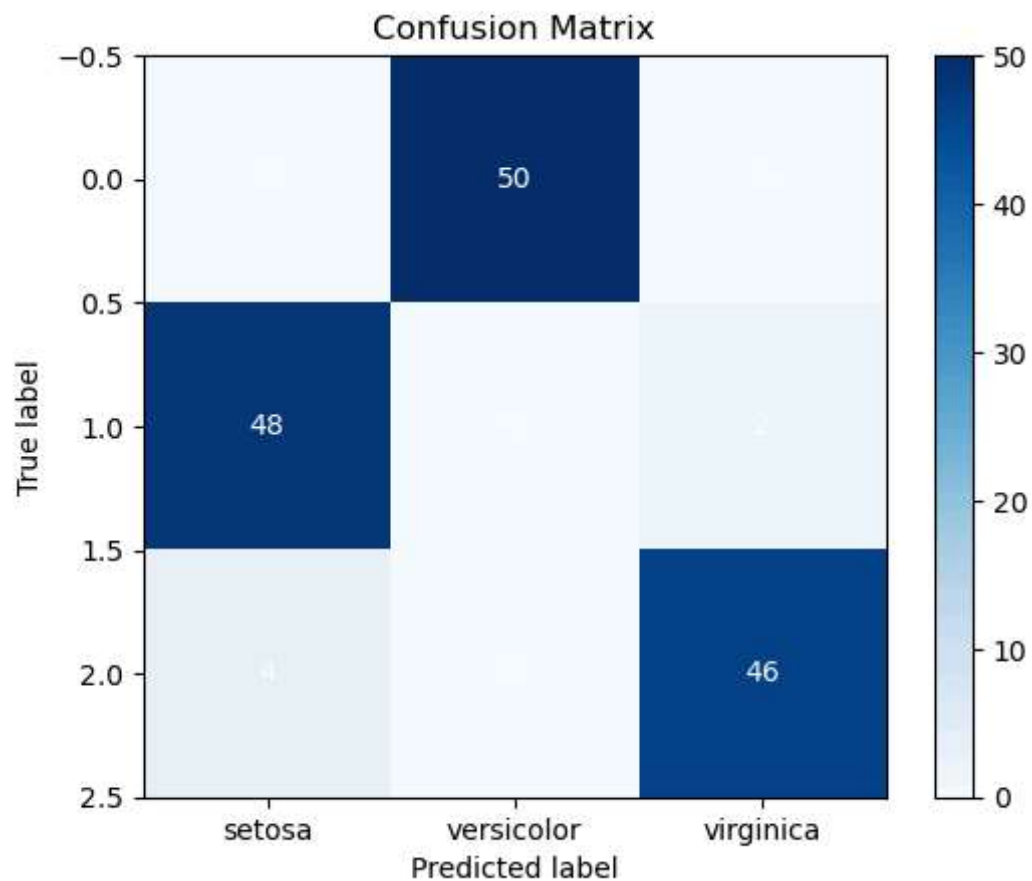Out[41]:
```
array([[ 0, 50,  0],
       [48,  0,  2],
       [ 4,  0, 46]], dtype=int64)
```

```python
In [42]: true_labels = df.species
         predicted_labels = df.cluster

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

         plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
         plt.title('Confusion Matrix')
         plt.colorbar()
         tick_marks = np.arange(len(class_labels))
         plt.xticks(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()
```



```python
In [ ]:
```