Home | Untitled2 - Jupyter Notebook

localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3

jupyter Untitled2 Last Checkpoint: 11 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

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

In [2]:
```
columns = ['Sepal length', 'Sepal width', 'Petal length', 'Petal width', 'Class_labels']
```

In [5]:
```
df = pd.read_csv('iris.data', names=columns)
```

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

Out[6]:

	Sepal length	Sepal width	Petal length	Petal width	Class_labels
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

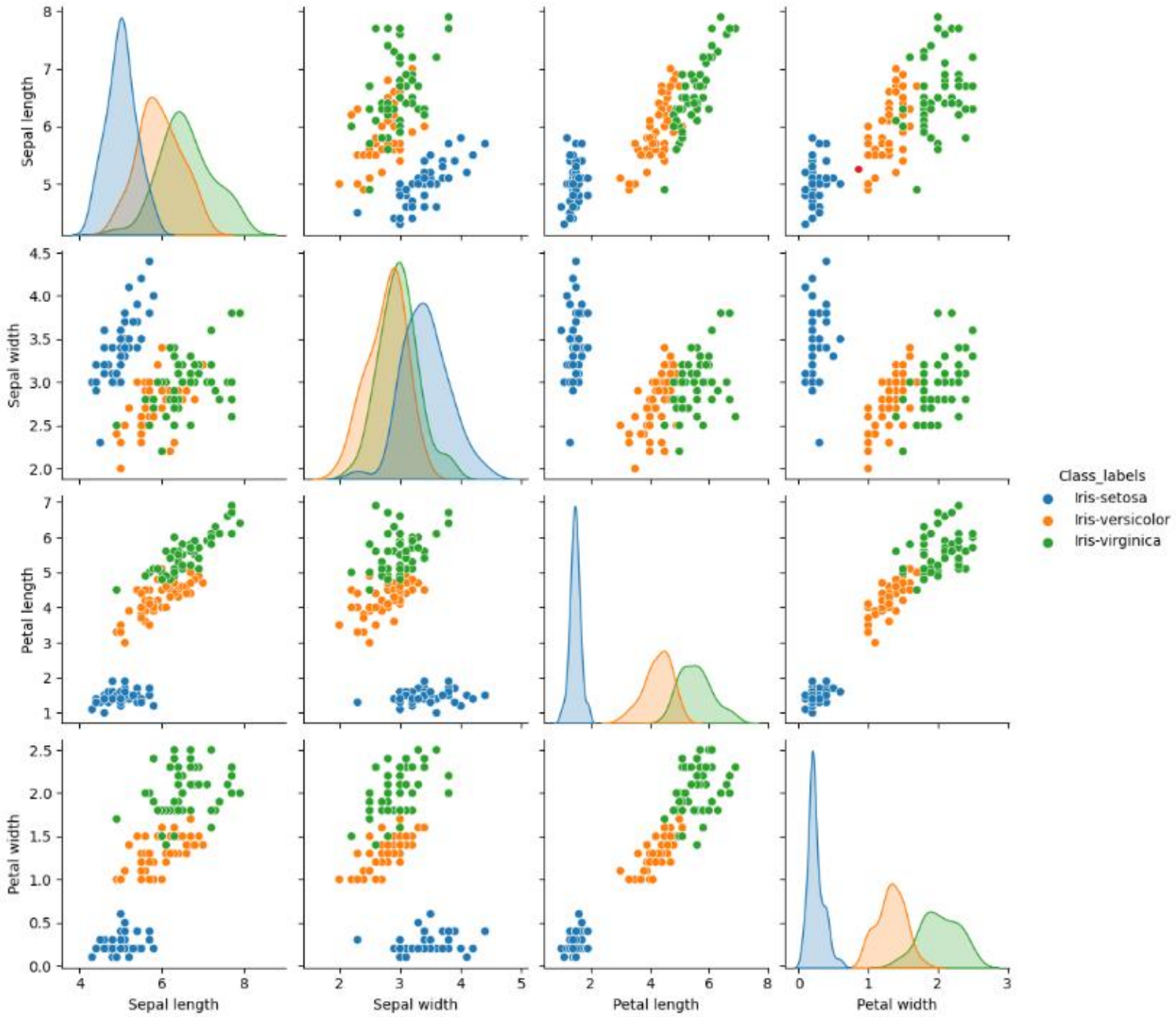
In [7]:
```
df.describe()
```

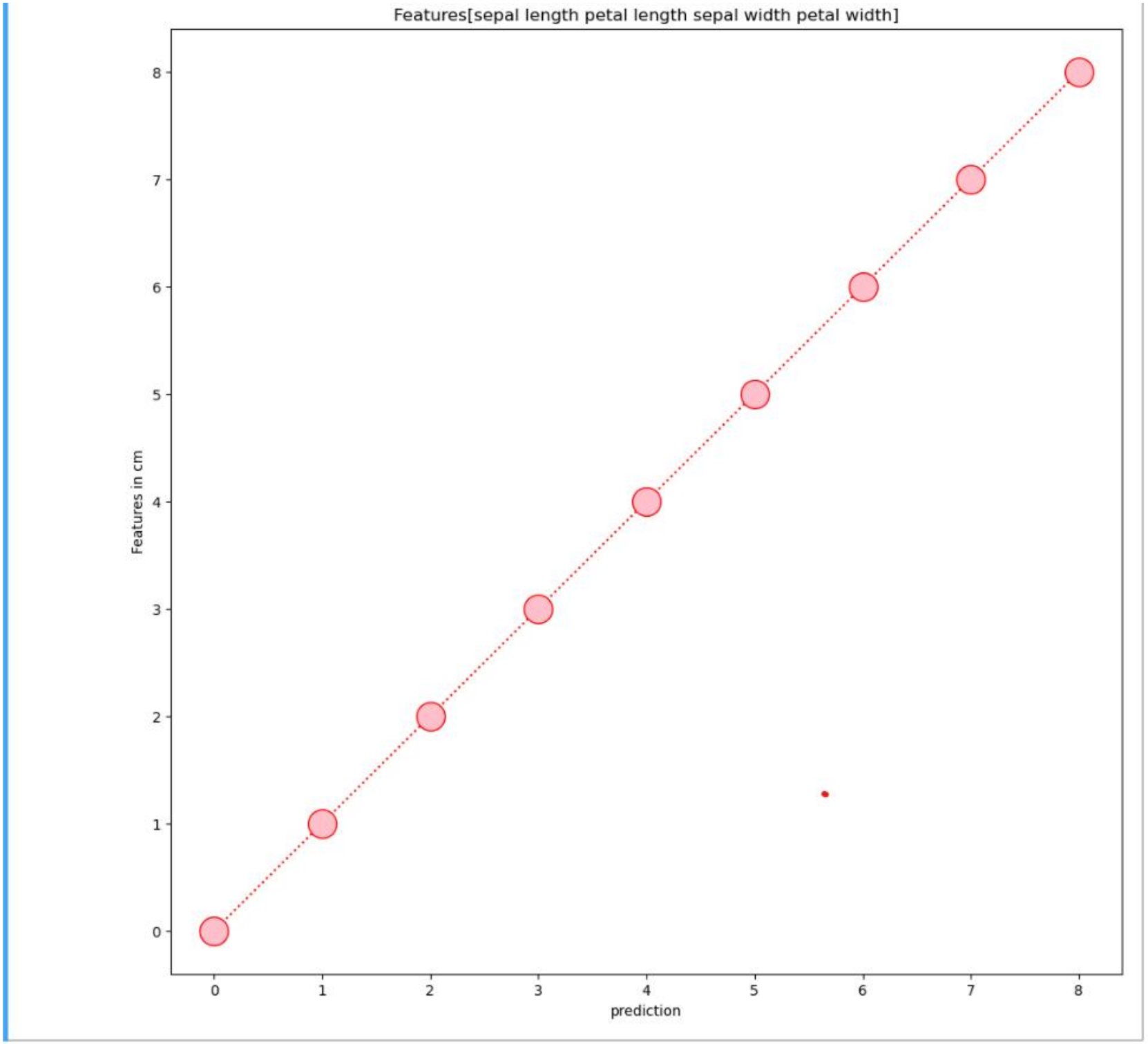
Out[7]:

	Sepal length	Sepal width	Petal length	Petal width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161

In [8]:
```
sns.pairplot(df, hue='Class_labels')
```

Out[8]:
```
<seaborn.axisgrid.PairGrid at 0x2b9bf06bd00>
```




Features[sepal length petal length sepal width petal width]
Features in cm
prediction
0
1
2
3
4
5
6
7
8

In [12]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2)
```

In [13]:
```
from sklearn.svm import SVC
svn = SVC()
svn.fit(X_train, y_train)
```

Out[13]:
SVC

SVC()

In [14]:
```
predictions = svn.predict(X_test)
```

In [15]:
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predictions)
```

Out[15]:
```
0.9666666666666667
```

In [18]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00         8
Iris-versicolor       0.92      1.00      0.96        12
 Iris-virginica       1.00      0.90      0.95        10

       accuracy                           0.97        30
      macro avg       0.97      0.97      0.97        30
   weighted avg       0.97      0.97      0.97        30
```

In [19]:
```
X_new = np.array([[3, 2, 1, 0.2], [  4.9, 2.2, 3.8, 1.1 ], [  5.3, 2.5, 4.6, 1.9 ]])
#Prediction of the species from the input vector
prediction = svn.predict(X_new)
print("Prediction of Species: {}".format(prediction))
```

```
Prediction of Species: ['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor']
```

In [18]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, predictions))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00         8
Iris-versicolor       0.92      1.00      0.96        12
 Iris-virginica       1.00      0.90      0.95        10

       accuracy                           0.97        30
      macro avg       0.97      0.97      0.97        30
   weighted avg       0.97      0.97      0.97        30
```

In [19]:
```
X_new = np.array([[3, 2, 1, 0.2], [  4.9, 2.2, 3.8, 1.1 ], [  5.3, 2.5, 4.6, 1.9 ]])
#Prediction of the species from the input vector
prediction = svn.predict(X_new)
print("Prediction of Species: {}".format(prediction))
```

```
Prediction of Species: ['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor']
```

In [20]:
```
import pickle
with open('SVM.pickle', 'wb') as f:
    pickle.dump(svn, f)
```

In [21]:
```
with open('SVM.pickle', 'rb') as f:
    model = pickle.load(f)
```

In [22]:
```
model.predict(X_new)
```

Out[22]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor'], dtype=object)
```