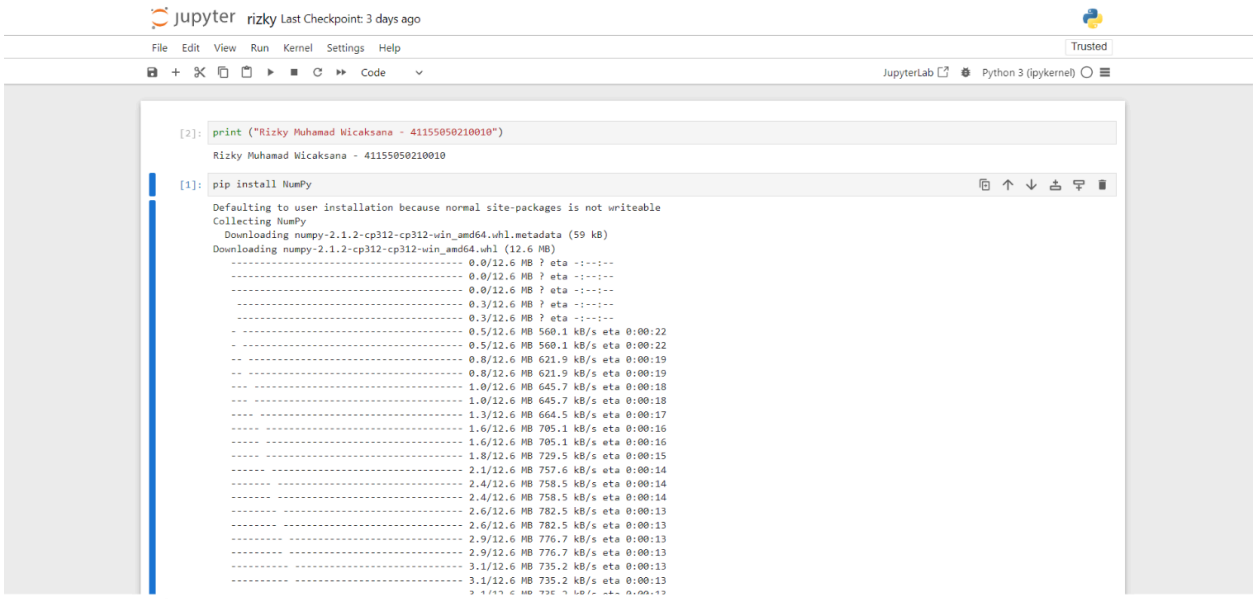
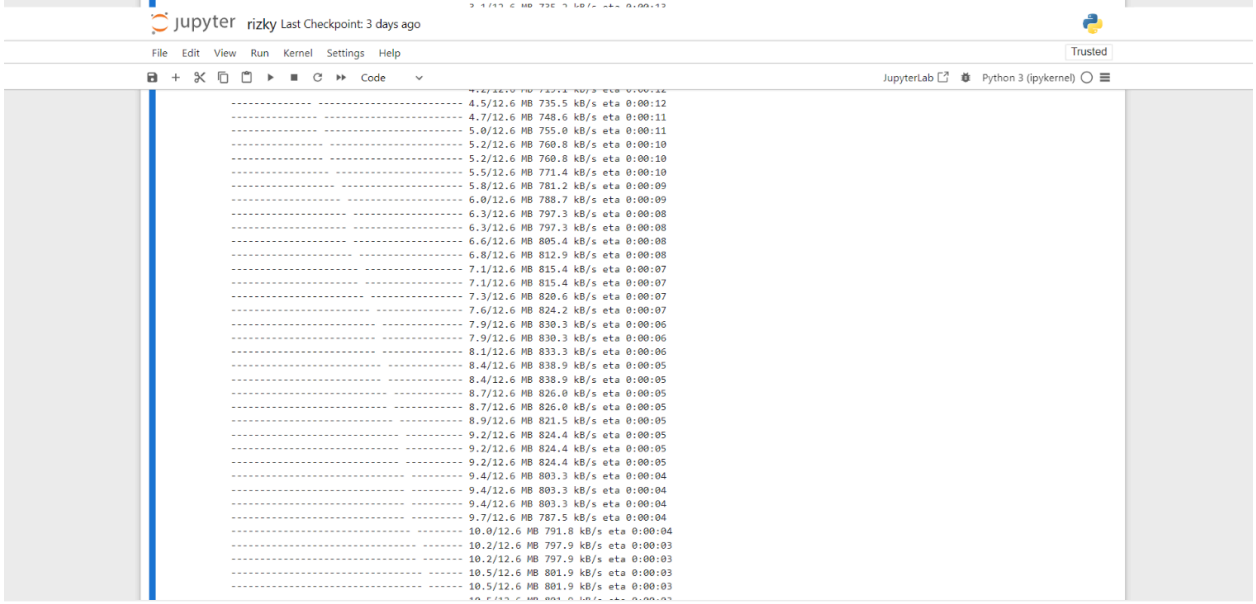
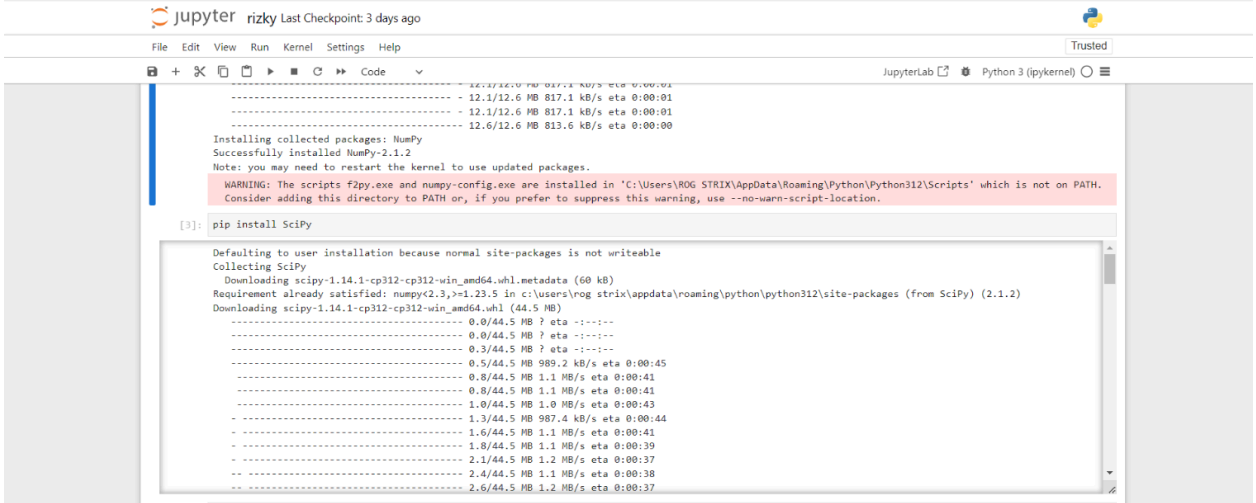
Nama : Rizky Muhamad Wicaksana
NPM : 41155050210060
Kelas : A1

Machine Learning

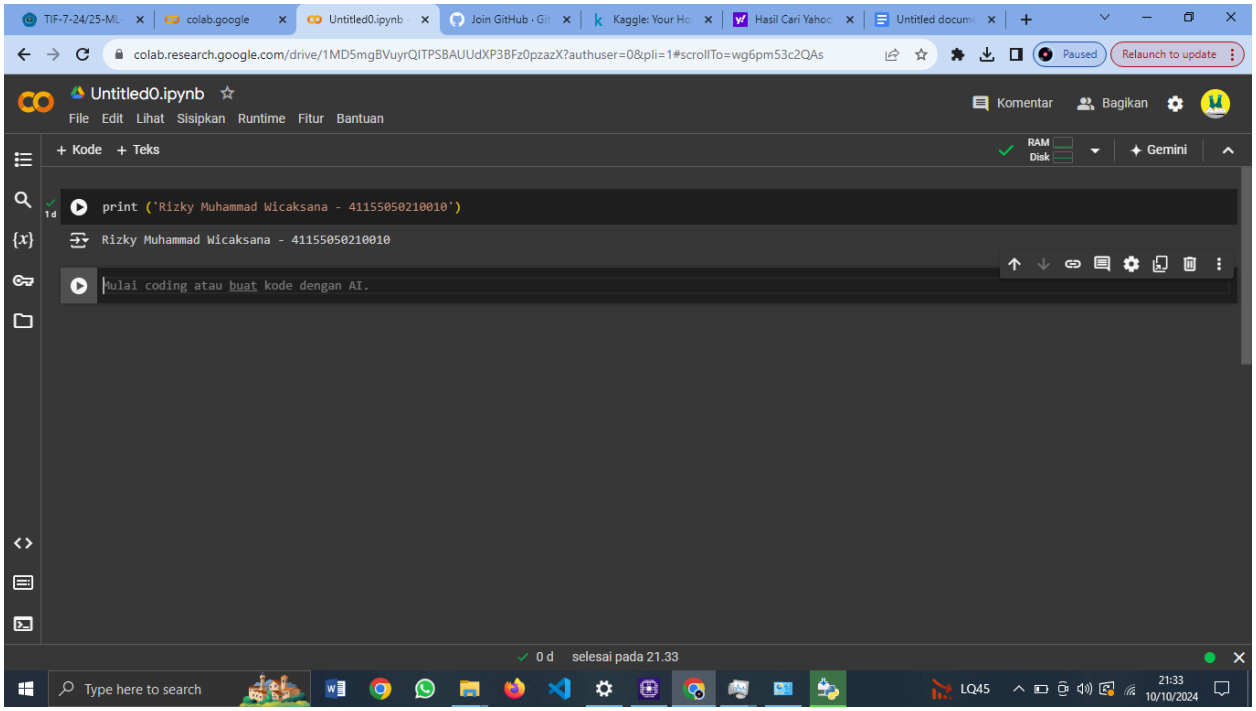
1. Instal Jupyter


jupyter rizky Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[2]: print ("Rizky Muhamad Wicaksana - 41155050210010")
Rizky Muhamad Wicaksana - 41155050210010
[1]: pip install NumPy
Defaulting to user installation because normal site-packages is not writeable
Collecting NumPy
Downloading numpy-2.1.2-cp312-cp312-win_amd64.whl.metadata (59 kB)
Downloading numpy-2.1.2-cp312-cp312-win_amd64.whl (12.6 MB)
0.0/12.6 MB ? eta -:--:--
0.0/12.6 MB ? eta -:--:--
0.0/12.6 MB ? eta -:--:--
0.3/12.6 MB ? eta -:--:--
0.3/12.6 MB ? eta -:--:--
0.5/12.6 MB 560.1 kB/s eta 0:00:22
0.5/12.6 MB 560.1 kB/s eta 0:00:22
0.8/12.6 MB 621.9 kB/s eta 0:00:19
0.8/12.6 MB 621.9 kB/s eta 0:00:19
1.0/12.6 MB 645.7 kB/s eta 0:00:18
1.0/12.6 MB 645.7 kB/s eta 0:00:18
1.3/12.6 MB 664.5 kB/s eta 0:00:17
1.6/12.6 MB 705.1 kB/s eta 0:00:16
1.6/12.6 MB 705.1 kB/s eta 0:00:16
1.8/12.6 MB 729.5 kB/s eta 0:00:15
2.1/12.6 MB 757.6 kB/s eta 0:00:14
2.4/12.6 MB 758.5 kB/s eta 0:00:14
2.4/12.6 MB 758.5 kB/s eta 0:00:14
2.6/12.6 MB 782.5 kB/s eta 0:00:13
2.6/12.6 MB 782.5 kB/s eta 0:00:13
2.9/12.6 MB 776.7 kB/s eta 0:00:13
2.9/12.6 MB 776.7 kB/s eta 0:00:13
3.1/12.6 MB 735.2 kB/s eta 0:00:13
3.1/12.6 MB 735.2 kB/s eta 0:00:13

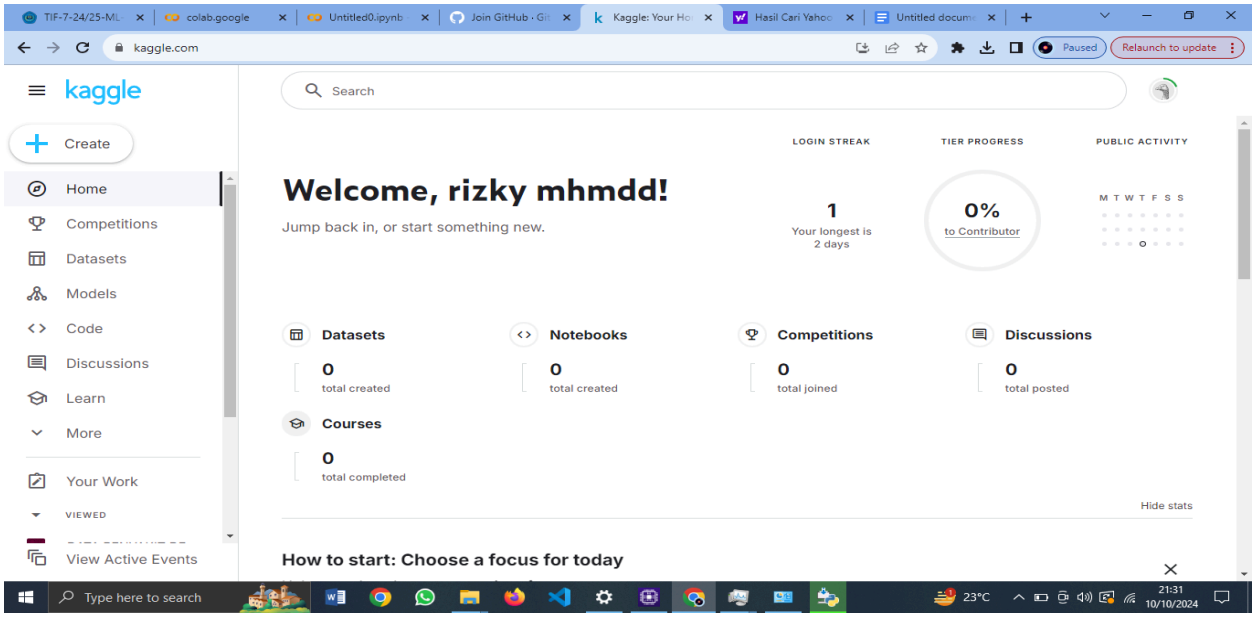

jupyter rizky Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
4.5/12.6 MB 735.5 kB/s eta 0:00:12
4.7/12.6 MB 748.6 kB/s eta 0:00:11
5.0/12.6 MB 755.0 kB/s eta 0:00:11
5.2/12.6 MB 760.8 kB/s eta 0:00:10
5.2/12.6 MB 760.8 kB/s eta 0:00:10
5.5/12.6 MB 771.4 kB/s eta 0:00:10
5.8/12.6 MB 781.2 kB/s eta 0:00:09
6.0/12.6 MB 788.7 kB/s eta 0:00:09
6.3/12.6 MB 797.3 kB/s eta 0:00:08
6.3/12.6 MB 797.3 kB/s eta 0:00:08
6.6/12.6 MB 805.4 kB/s eta 0:00:08
6.8/12.6 MB 812.9 kB/s eta 0:00:08
7.1/12.6 MB 815.4 kB/s eta 0:00:07
7.1/12.6 MB 815.4 kB/s eta 0:00:07
7.3/12.6 MB 820.6 kB/s eta 0:00:07
7.6/12.6 MB 824.2 kB/s eta 0:00:07
7.9/12.6 MB 830.3 kB/s eta 0:00:06
7.9/12.6 MB 830.3 kB/s eta 0:00:06
8.1/12.6 MB 833.3 kB/s eta 0:00:06
8.4/12.6 MB 838.9 kB/s eta 0:00:05
8.4/12.6 MB 838.9 kB/s eta 0:00:05
8.7/12.6 MB 826.0 kB/s eta 0:00:05
8.7/12.6 MB 826.0 kB/s eta 0:00:05
8.9/12.6 MB 821.5 kB/s eta 0:00:05
9.2/12.6 MB 824.4 kB/s eta 0:00:05
9.2/12.6 MB 824.4 kB/s eta 0:00:05
9.2/12.6 MB 824.4 kB/s eta 0:00:05
9.4/12.6 MB 803.3 kB/s eta 0:00:04
9.4/12.6 MB 803.3 kB/s eta 0:00:04
9.4/12.6 MB 803.3 kB/s eta 0:00:04
9.7/12.6 MB 787.5 kB/s eta 0:00:04
10.0/12.6 MB 791.8 kB/s eta 0:00:04
10.2/12.6 MB 797.9 kB/s eta 0:00:03
10.2/12.6 MB 797.9 kB/s eta 0:00:03
10.5/12.6 MB 801.9 kB/s eta 0:00:03
10.5/12.6 MB 801.9 kB/s eta 0:00:03


jupyter rizky Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
12.1/12.6 MB 817.1 kB/s eta 0:00:01
12.1/12.6 MB 817.1 kB/s eta 0:00:01
12.6/12.6 MB 813.6 kB/s eta 0:00:00
Installing collected packages: NumPy
Successfully installed NumPy-2.1.2
Note: you may need to restart the kernel to use updated packages.
WARNING: The scripts f2py.exe and numpy-config.exe are installed in 'C:\Users\ROG STRIX\AppData\Roaming\Python\Python312\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
[3]: pip install SciPy
Defaulting to user installation because normal site-packages is not writeable
Collecting SciPy
Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.3,>=1.23.5 in c:\users\rog strix\appdata\roaming\python\python312\site-packages (from SciPy) (2.1.2)
Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
0.0/44.5 MB ? eta -:--:--
0.0/44.5 MB ? eta -:--:--
0.3/44.5 MB ? eta -:--:--
0.5/44.5 MB 989.2 kB/s eta 0:00:45
0.8/44.5 MB 1.1 MB/s eta 0:00:41
0.8/44.5 MB 1.1 MB/s eta 0:00:41
1.0/44.5 MB 1.0 MB/s eta 0:00:43
1.3/44.5 MB 987.4 kB/s eta 0:00:44
1.6/44.5 MB 1.1 MB/s eta 0:00:41
1.8/44.5 MB 1.1 MB/s eta 0:00:39
2.1/44.5 MB 1.2 MB/s eta 0:00:37
2.4/44.5 MB 1.1 MB/s eta 0:00:38
2.6/44.5 MB 1.2 MB/s eta 0:00:37

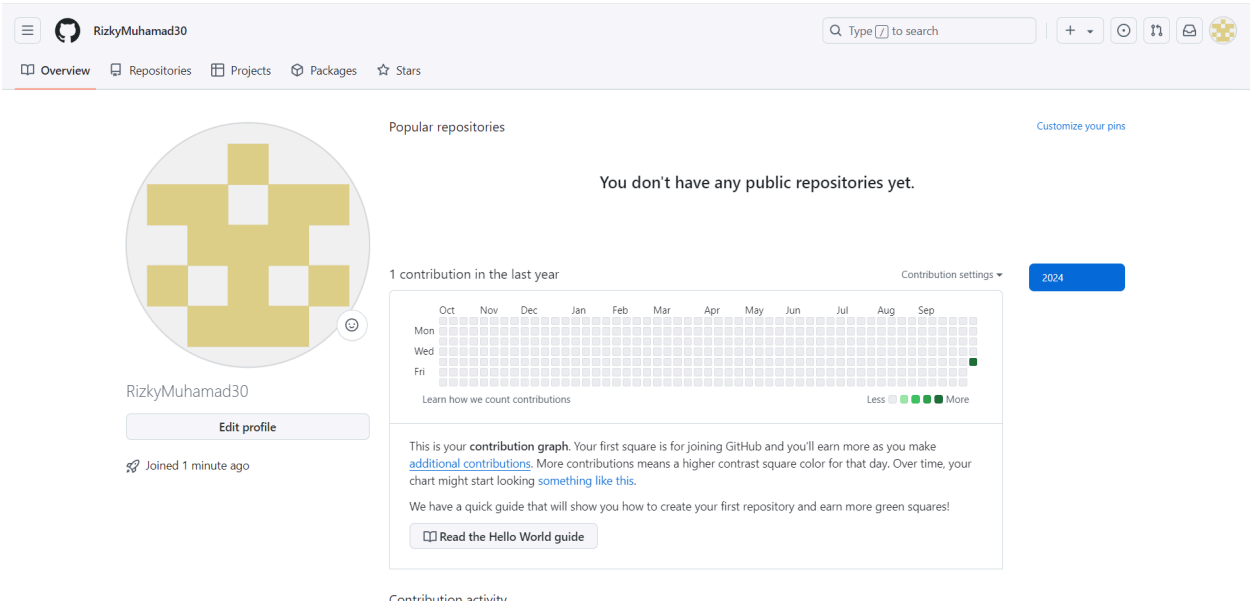
2. Google Collab



3. Kaggle



4. Github



5. Prak

Load Dataset

[5]:
```
from sklearn.datasets import load_iris

iris = load_iris()
iris
```

[5]:
```
{'data': array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
```

[7]:
```
iris.keys()
```

[7]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])
```

Deskripsi dari sampel Dataset

[8]:
```
print(iris.DESCR)
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

:Number of Instances: 150 (50 in each of three classes)
:Number of Attributes: 4 numeric, predictive attributes and the class
:Attribute Information:
    - sepal length in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
    - class:
            - Iris-Setosa
            - Iris-Versicolour
            - Iris-Virginica

:Summary Statistics:

============== ==== ==== ======= ===== ====================
                Min  Max   Mean    SD   Class Correlation
============== ==== ==== ======= ===== ====================
sepal length:   4.3  7.9   5.84   0.83    0.7826
sepal width:    2.0  4.4   3.05   0.43   -0.4194
petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
============== ==== ==== ======= ===== ====================

:Missing Attribute Values: None
:Class Distribution: 33.3% for each of 3 classes.
```

Explanatory & Response Variable (Feature & Target)

[10]:
```
x = iris.data
x .shape
# x
```

[10]:
```
(150, 4)
```

[11]:
```
y = iris.target
# y.shape
y
```

[11]:
```
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, 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])
```

[ ]:

Feature & Target Names

[12]:
```
feature_names = iris.feature_names
feature_names
```

[12]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

[13]:
```
target_names = iris.target_names
target_names
```

[13]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

Vasulisasi Data
Visualisasi Sepal Lenght & Width

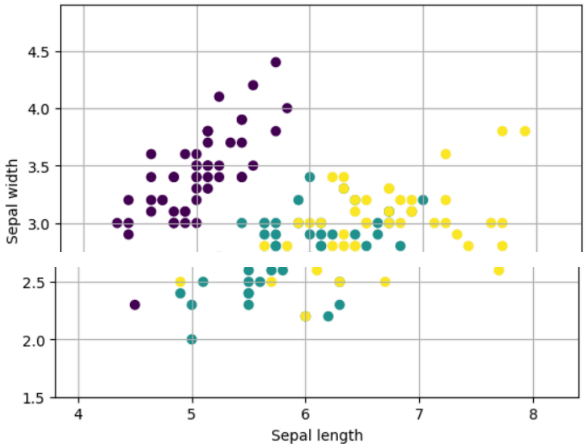
[24]:
```
import matplotlib.pyplot as plt

X = x[:, :2]

x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5
y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5

plt.scatter(X[:,0], x[:, 1], c=y)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')

plt.xlim(x_min, x_max)
plt.ylim(y_min, y_max)
plt.grid(True)
plt.show()
```



Training & Testing Dataset

[28]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=1)
print(f'X train : {X_train.shape}')
print(f'X test : {X_test.shape}')
print(f'y train : {y_train.shape}')
print(f'y test : {y_test.shape}')
```

```
X train : (105, 2)
X test : (45, 2)
y train : (105,)
y test : (45,)
```

Load Iris Dataset sebagai Pandas DataFrame

[ ]:

03 Worklow Dengan Scikit Learn

Persiapan Dataset

[2]:
```
from sklearn.datasets import load_iris

iris = load_iris()

X = iris.data
y = iris.target
```

Splitting Dataset:Training & Testing Set

[4]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=1)
```

Double-click (or enter) to edit

[6]:
```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors=3)
model.fit(X_train, y_train)
```

[6]:
```
KNeighborsClassifier(n_neighbors=3)
```

Evaluasi Model

[7]:
```
from sklearn.metrics import accuracy_score

y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print(f'Accuracy :{acc}')
```

```
Accuracy :0.9777777777777777
```

Pemanfaatan Trained Model

[8]:
```
data_baru = [[5, 5, 3, 2],
             [2, 4, 3, 5]]
preds = model.predict(data_baru)
preds
```

[8]:
```
array([1, 2])
```

[9]:
```
pred_species = [iris.target_names[p] for p in preds]
print(f'Hasil prediksi : {pred_species}')
```

```
Hasil prediksi : [np.str_('versicolor'), np.str_('virginica')]
```

Dump & Load Trained Model

Dumping Model Machine Learning menjadi File Joblib

[10]:
```
import joblib

joblib.dump(model, 'iris_classifier_knn.jonlib') #(tren model, nama file joblib)
```

[10]:
```
['iris_classifier_knn.jonlib']
```

Loading Model Machine Learning dari File Joblib

[11]:
```
production_model = joblib.load('iris_classifier_knn.jonlib')
```

04 Data Preprocessing dengan Scikit-Learn

Sampel Data

[1]:
```
import numpy as np
from sklearn import preprocessing

sample_data = np.array ([[2.1, -1.9, 5.5],
                         [-1.5, 2.4, 3.5],
                         [0.5, -7.9, 5.6],
                         [5.9, 2.3, -5.8]])
sample_data
```

[1]:
```
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

[2]:
```
sample_data.shape
```

[2]:
```
(4, 3)
```

Binarisation

[3]:
```
sample_data
```

[3]:
```
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

[4]:
```
preprocessor = preprocessing.Binarizer(threshold=0.5)
binarised_data = preprocessor.transform(sample_data)
binarised_data
```

[4]:
```
array([[1., 0., 1.],
       [0., 1., 1.],
       [0., 0., 1.],
       [1., 1., 0.]])
```

Scalling

[5]:
```
sample_data
```

[5]:
```
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

[6]:
```
preprocessor = preprocessing.MinMaxScaler(feature_range=(0, 1))
preprocessor.fit(sample_data)
scaled_data = preprocessor.transform(sample_data)
scaled_data
```

[6]:
```
array([[0.48648649, 0.58252427, 0.99122807],
       [0.        , 1.        , 0.81578947],
       [0.27027027, 0.        , 1.        ],
       [1.        , 0.99029126, 0.        ]])
```

[7]:
```
scaled_data = preprocessor.fit_transform(sample_data)
scaled_data
```

[7]:
```
array([[0.48648649, 0.58252427, 0.99122807],
       [0.        , 1.        , 0.81578947],
       [0.27027027, 0.        , 1.        ],
       [1.        , 0.99029126, 0.        ]])
```

L1 Normalisation: Least Absolute Deviations Referensi : https://en.wikipedia.org/wiki/Least_absolute_deviations

[8]:
```
sample_data
```

[8]:
```
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

[9]:
```
l1_normalised_data = preprocessing.normalize(sample_data, norm='l1')
l1_normalised_data
```

[9]:
```
array([[ 0.22105263, -0.2       ,  0.57894737],
       [-0.2027027 ,  0.32432432,  0.47297297],
       [ 0.03571429, -0.56428571,  0.4       ],
       [ 0.42142857,  0.16428571, -0.41428571]])
```

L2 Normalisation : Least Squares Referensi : https://en.wikipedia.org/wiki/Least_squares

[10]:
```
sample_data
```

[10]:
```
array([[ 2.1, -1.9,  5.5],
       [-1.5,  2.4,  3.5],
       [ 0.5, -7.9,  5.6],
       [ 5.9,  2.3, -5.8]])
```

[11]:
```
l2_normalised_data = preprocessing.normalize(sample_data, norm='l2')
l2_normalised_data
```

[11]:
```
array([[ 0.33946114, -0.30713151,  0.88906489],
       [-0.33325106,  0.53320169,  0.7775858 ],
       [ 0.05156558, -0.81473612,  0.57753446],
       [ 0.68706914,  0.26784051, -0.6754239 ]])
```