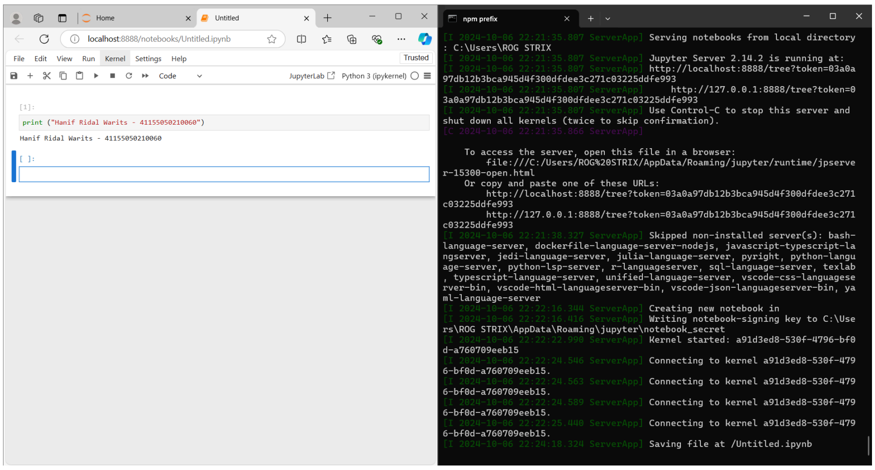
MACHINE LEARNING

Nama : Hanif Ridal Warits

NPM : 41155050210060

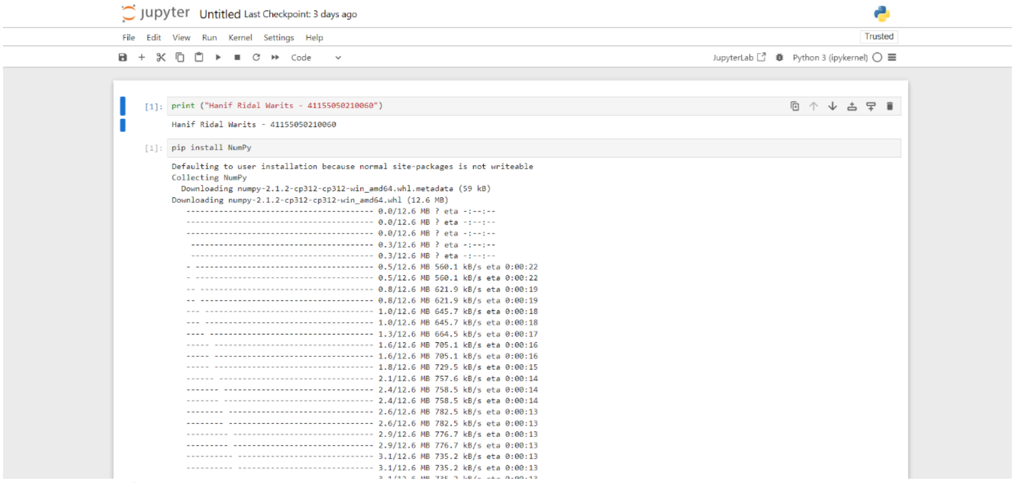
Kelas : Informatika A2 - 2021

1. Install Jupyter Notebook

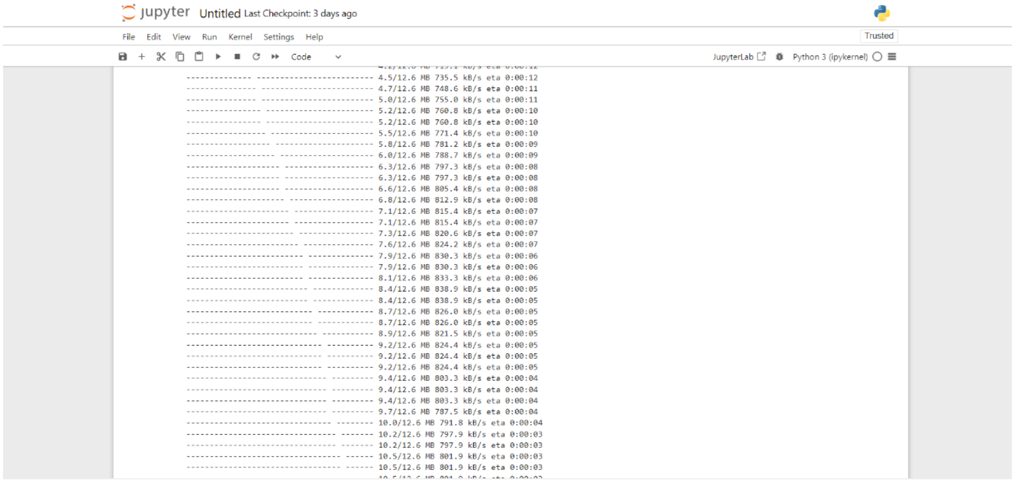


```
[1]:
print ("Hanif Ridal Warits - 41155050210060")
Hanif Ridal Warits - 41155050210060

[ ]:
```

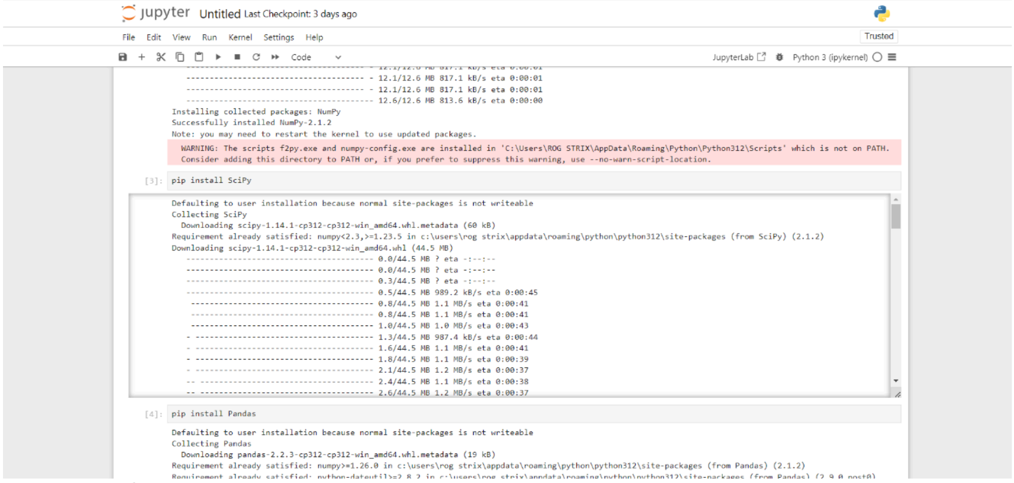
```
[I 2024-10-06 22:21:35.807 ServerApp] Serving notebooks from local directory
: C:\Users\ROG STRIX
[I 2024-10-06 22:21:35.807 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-10-06 22:21:35.807 ServerApp] http://localhost:8888/tree?token=03a0a
97db12b3bca945d4f300dfdee3c271c03225ddfe993
[I 2024-10-06 22:21:35.807 ServerApp]     http://127.0.0.1:8888/tree?token=0
3a0a97db12b3bca945d4f300dfdee3c271c03225ddfe993
[I 2024-10-06 22:21:35.807 ServerApp] Use Control-C to stop this server and
shut down all kernels (twice to skip confirmation).
[C 2024-10-06 22:21:35.866 ServerApp]

    To access the server, open this file in a browser:
        file:///C:/Users/ROG%20STRIX/AppData/Roaming/jupyter/runtime/jpserve
r-15300-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/tree?token=03a0a97db12b3bca945d4f300dfdee3c271
c03225ddfe993
        http://127.0.0.1:8888/tree?token=03a0a97db12b3bca945d4f300dfdee3c271
c03225ddfe993
[I 2024-10-06 22:21:38.327 ServerApp] Skipped non-installed server(s): bash-
language-server, dockerfile-language-server-nodejs, javascript-typescript-la
ngserver, jedi-language-server, julia-language-server, pyright, python-langu
age-server, python-lsp-server, r-languageserver, sql-language-server, texlab
, typescript-language-server, unified-language-server, vscode-css-languagese
rver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, ya
ml-language-server
[I 2024-10-06 22:22:16.344 ServerApp] Creating new notebook in
[I 2024-10-06 22:22:16.416 ServerApp] Writing notebook-signing key to C:\Use
rs\ROG STRIX\AppData\Roaming\jupyter\notebook_secret
[I 2024-10-06 22:22:22.990 ServerApp] Kernel started: a91d3ed8-530f-4796-bf0
d-a760709eeb15
[I 2024-10-06 22:22:24.546 ServerApp] Connecting to kernel a91d3ed8-530f-479
6-bf0d-a760709eeb15.
[I 2024-10-06 22:22:24.563 ServerApp] Connecting to kernel a91d3ed8-530f-479
6-bf0d-a760709eeb15.
[I 2024-10-06 22:22:24.589 ServerApp] Connecting to kernel a91d3ed8-530f-479
6-bf0d-a760709eeb15.
[I 2024-10-06 22:22:25.440 ServerApp] Connecting to kernel a91d3ed8-530f-479
6-bf0d-a760709eeb15.
[I 2024-10-06 22:24:18.324 ServerApp] Saving file at /Untitled.ipynb
```


jupyter Untitled Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

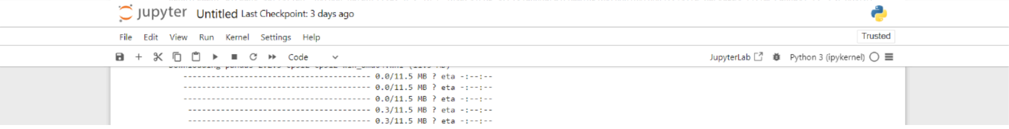
```
[1]: print ("Hanif Ridal Warits - 41155050210060")

     Hanif Ridal Warits - 41155050210060

[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 Untitled 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
```

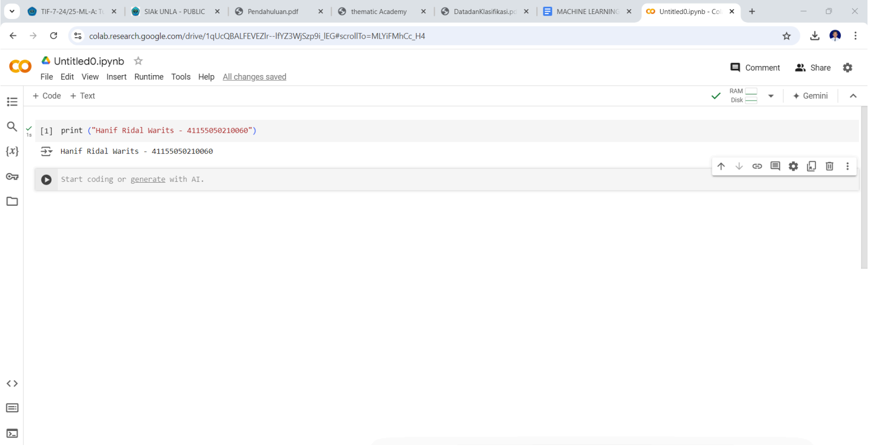

jupyter Untitled Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
   -------------------------------------- - 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

[4]: pip install Pandas

Defaulting to user installation because normal site-packages is not writeable
Collecting Pandas
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\rog strix\appdata\roaming\python\python312\site-packages (from Pandas) (2.1.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\rog strix\appdata\roaming\python\python312\site-packages (from Pandas) (2.9.0.post0)
```


jupyter Untitled Last Checkpoint: 3 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
   ---------------------------------------- 0.0/11.5 MB ? eta -:--:--
   ---------------------------------------- 0.0/11.5 MB ? eta -:--:--
   ---------------------------------------- 0.0/11.5 MB ? eta -:--:--
    --------------------------------------- 0.3/11.5 MB ? eta -:--:--
    --------------------------------------- 0.3/11.5 MB ? eta -:--:--
```

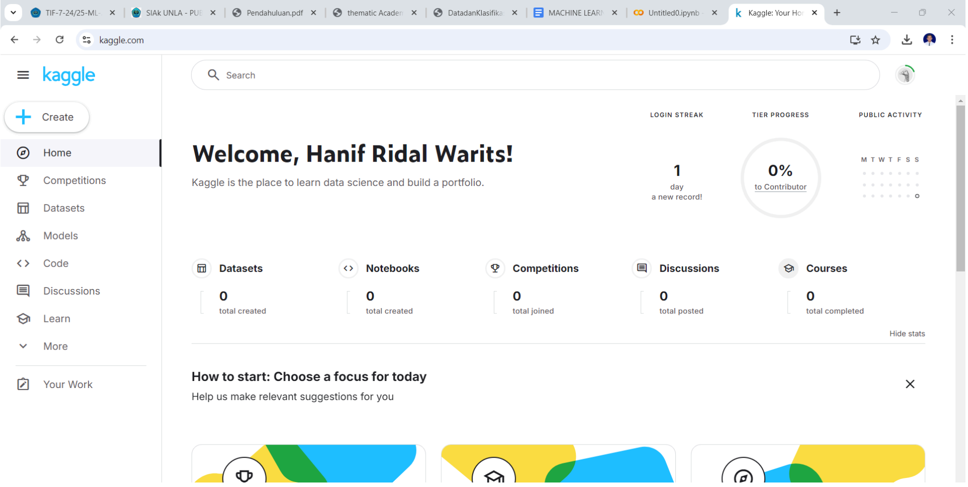
2. Google Collab



```
[1] print ("Hanif Ridal Warits - 41155050210060")
```

```
Hanif Ridal Warits - 41155050210060
```

3. Akun Kaggle : https://www.kaggle.com/hanifkrong



Welcome, Hanif Ridal Warits!

Kaggle is the place to learn data science and build a portfolio.

1 day a new record!

0% to Contributor

Datasets 0 total created

Notebooks 0 total created

Competitions 0 total joined

Discussions 0 total posted

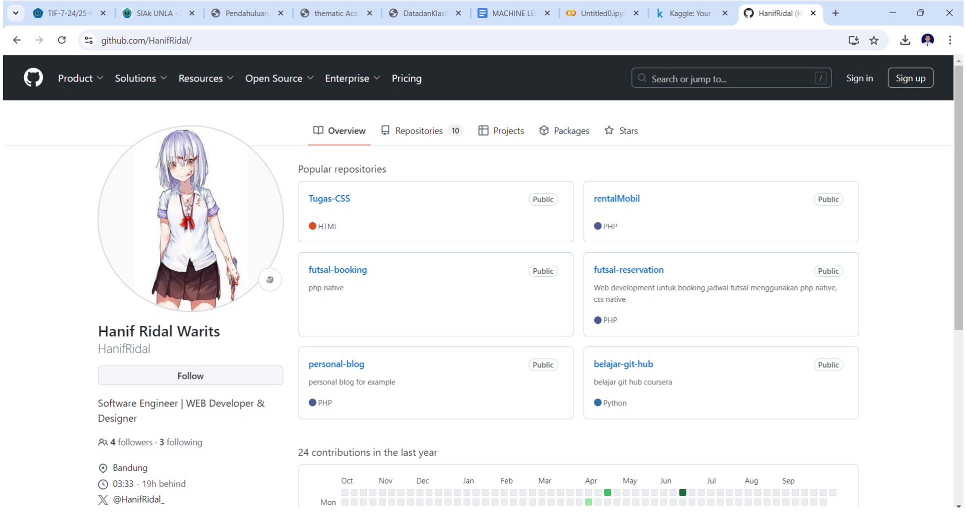
Courses 0 total completed

Hide stats

How to start: Choose a focus for today

Help us make relevant suggestions for you

4. Akun GitHub : https://github.com/HanifRidal/



Hanif Ridal Warits

HanifRidal

Software Engineer | WEB Developer & Designer

4 followers · 3 following

Bandung

03:33 - 19h behind

@HanifRidal_

Popular repositories

Tugas-CSS Public
HTML

rentalMobil Public
PHP

futsal-booking Public
php native

futsal-reservation Public
Web development untuk booking jadwal futsal menggunakan php native, css native
PHP

personal-blog Public
personal blog for example
PHP

belajar-git-hub Public
belajar git hub coursera
Python

24 contributions in the last year

5.0. Lakukan praktek dari https://youtu.be/mSO2hJIn0OY?feature=shared . Praktek tersebut yaitu:

5.1. Load sample dataset

5.2. Metadata | Deskripsi dari sample dataset

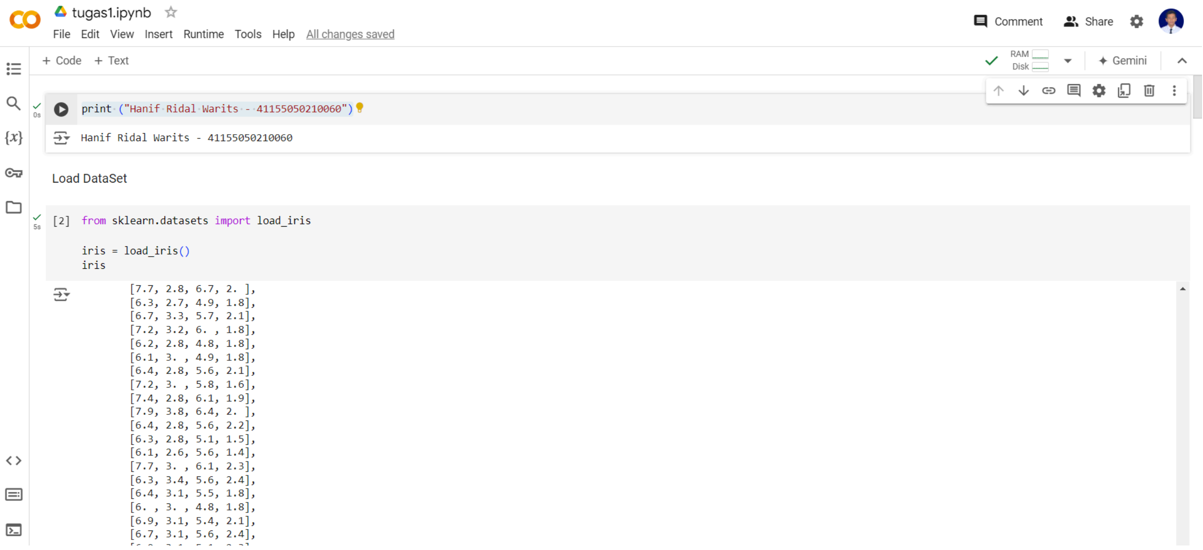
5.3. Explanatory & Response Variables | Features & Target

5.4. Feature & Target Names

5.5. Visualisasi Data

5.6. Training Set & Testing Set

5.7. Load sample dataset sebagai Pandas Data Frame



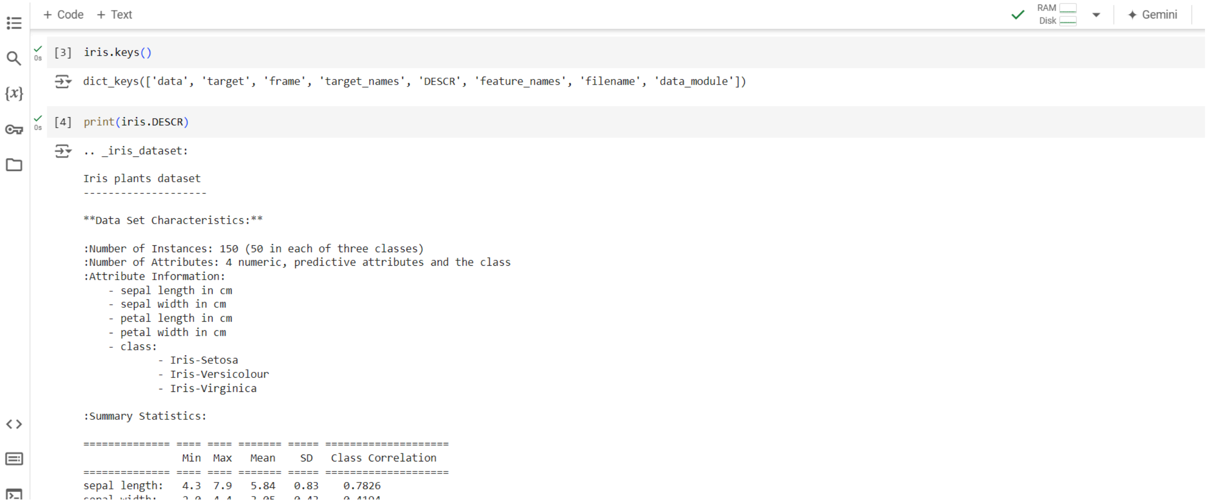
```
print ("Hanif Ridal Warits - 41155050210060")
```

```
Hanif Ridal Warits - 41155050210060
```

Load DataSet

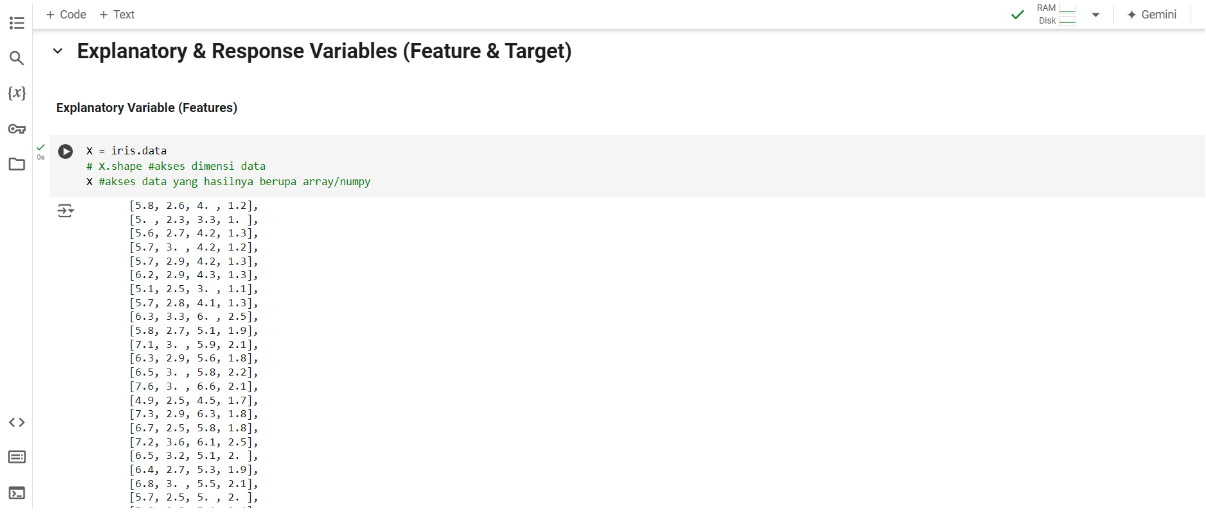
```
[2] from sklearn.datasets import load_iris

    iris = load_iris()
    iris
```

```
        [7.7, 2.8, 6.7, 2. ],
        [6.3, 2.7, 4.9, 1.8],
        [6.7, 3.3, 5.7, 2.1],
        [7.2, 3.2, 6. , 1.8],
        [6.2, 2.8, 4.8, 1.8],
        [6.1, 3. , 4.9, 1.8],
        [6.4, 2.8, 5.6, 2.1],
        [7.2, 3. , 5.8, 1.6],
        [7.4, 2.8, 6.1, 1.9],
        [7.9, 3.8, 6.4, 2. ],
        [6.4, 2.8, 5.6, 2.2],
        [6.3, 2.8, 5.1, 1.5],
        [6.1, 2.6, 5.6, 1.4],
        [7.7, 3. , 6.1, 2.3],
        [6.3, 3.4, 5.6, 2.4],
        [6.4, 3.1, 5.5, 1.8],
        [6. , 3. , 4.8, 1.8],
        [6.9, 3.1, 5.4, 2.1],
        [6.7, 3.1, 5.6, 2.4],
```



```
[3] iris.keys()

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

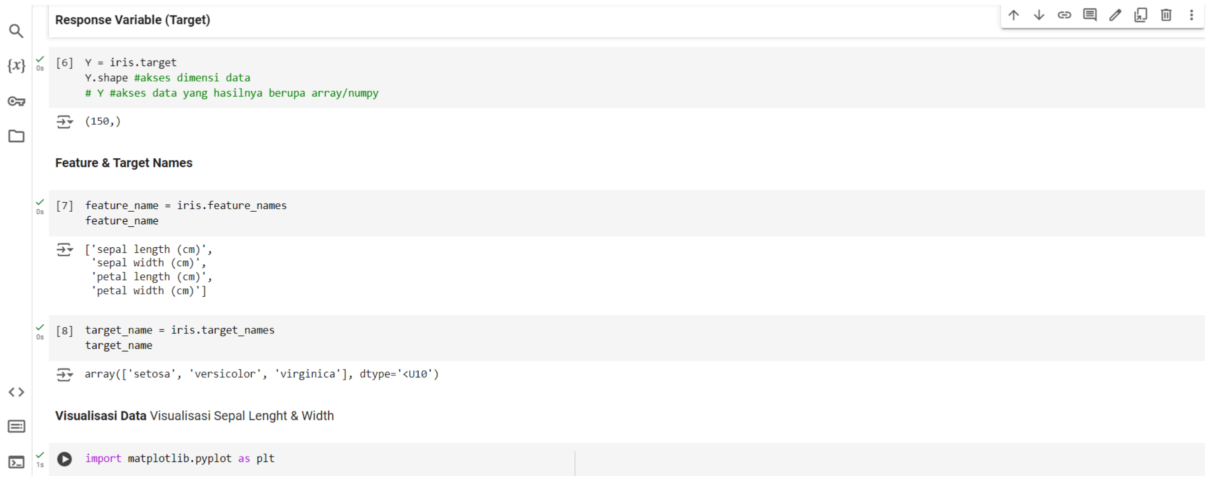
```
[4] print(iris.DESCR)

.. _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
```



Explanatory & Response Variables (Feature & Target)

Explanatory Variable (Features)

```
X = iris.data
# X.shape #akses dimensi data
X #akses data yang hasilnya berupa array/numpy

       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
```



Response Variable (Target)

```
[6] Y = iris.target
    Y.shape #akses dimensi data
    # Y #akses data yang hasilnya berupa array/numpy

(150,)
```

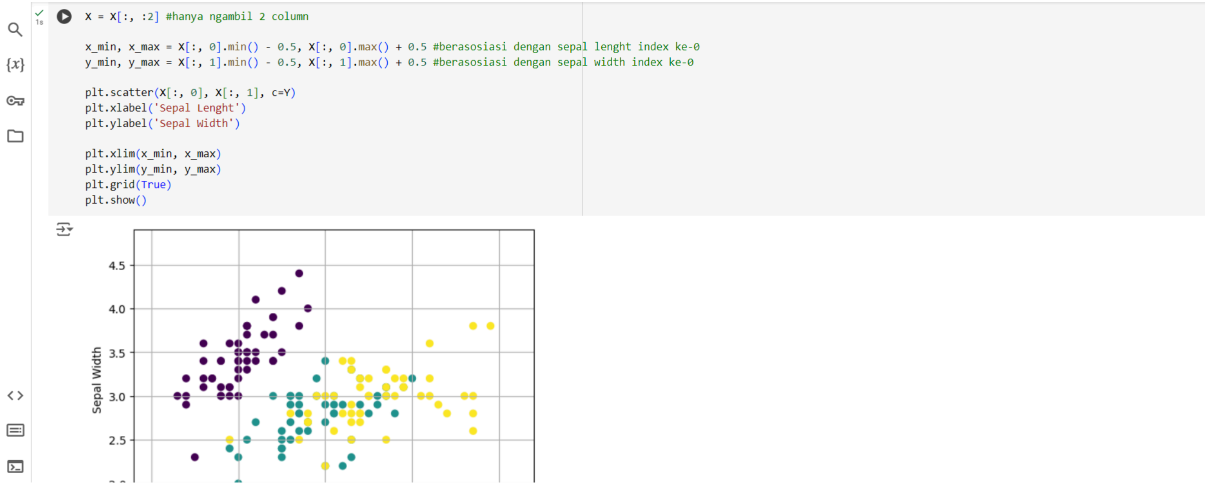
Feature & Target Names

```
[7] feature_name = iris.feature_names
    feature_name

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

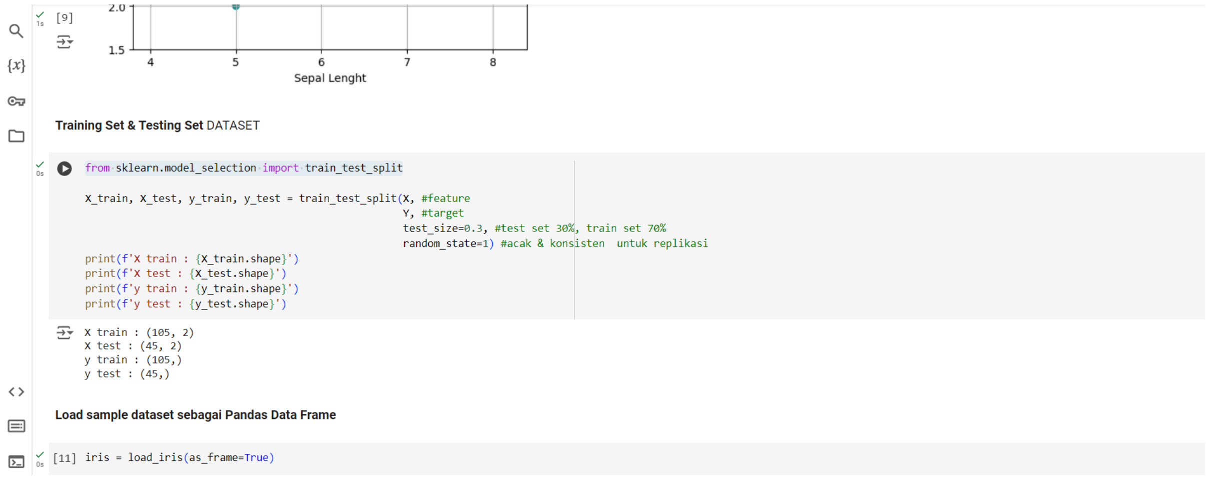
```
[8] target_name = iris.target_names
    target_name

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

Visualisasi Data Visualisasi Sepal Lenght & Width

```
import matplotlib.pyplot as plt
```



```
X = X[:, :2] #hanya ngambil 2 column

x_min, x_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5 #berasosiasi dengan sepal lenght index ke-0
y_min, y_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5 #berasosiasi dengan sepal width index ke-0

plt.scatter(X[:, 0], X[:, 1], c=Y)
plt.xlabel('Sepal Lenght')
plt.ylabel('Sepal Width')

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



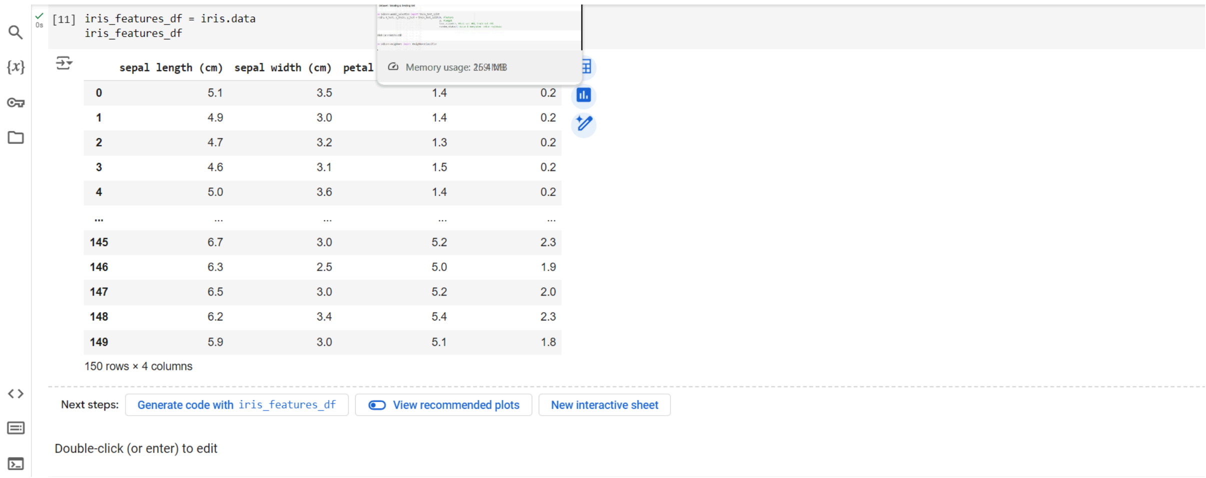
Training Set & Testing Set DATASET

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, #feature
                                                    Y, #target
                                                    test_size=0.3, #test set 30%, train set 70%
                                                    random_state=1) #acak & konsisten  untuk replikasi

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 sample dataset sebagai Pandas Data Frame

```
[11] iris = load_iris(as_frame=True)
```



```
[11] iris_features_df = iris.data
     iris_features_df
```

	sepal length (cm)	sepal width (cm)	petal	
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

Double-click (or enter) to edit

6.0. Lakukan praktek dari https://youtu.be/tiREcHrtDLo?feature=shared . Praktek tersebut yaitu:

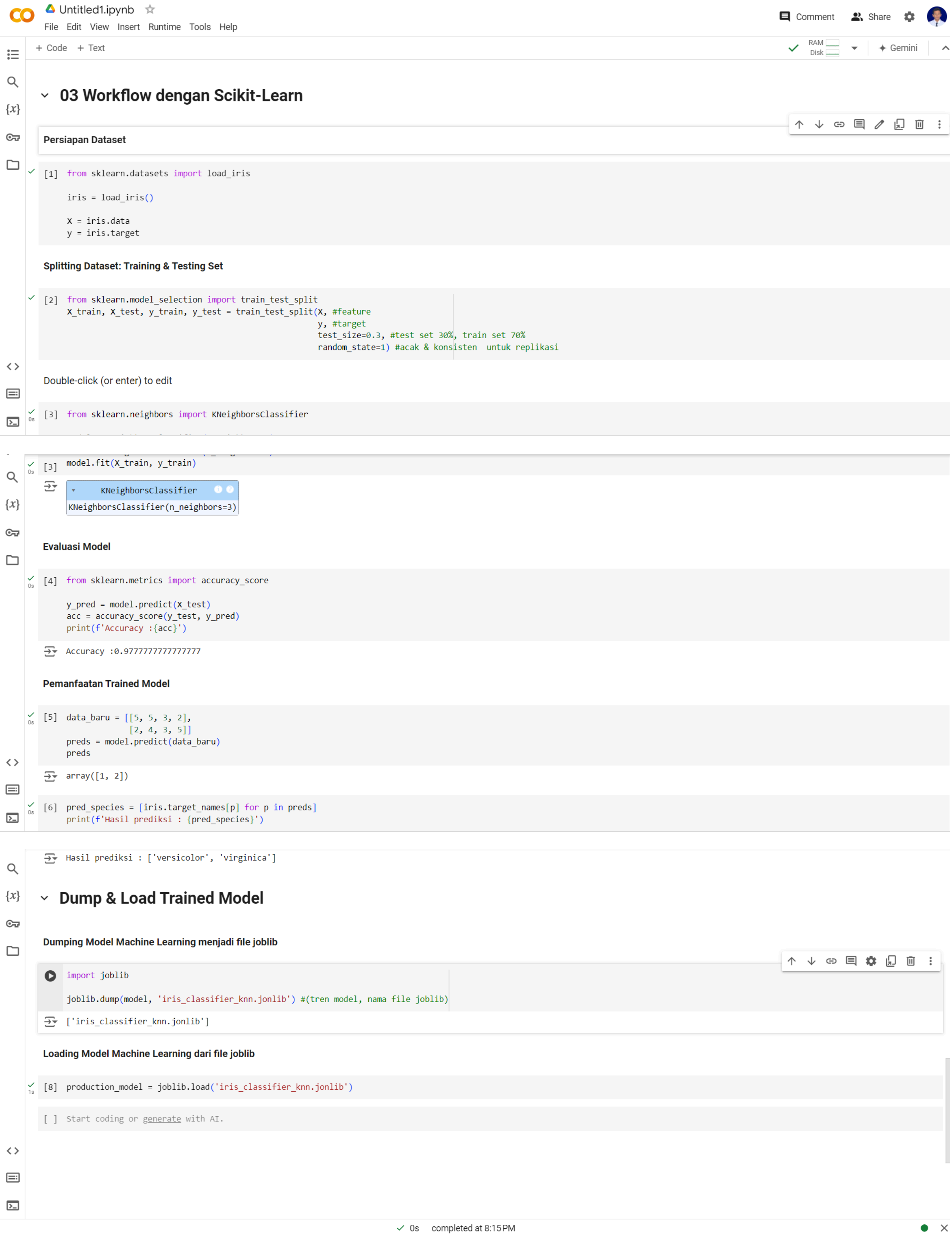
6.1. Persiapan dataset | Loading & splitting dataset

6.2. Training model Machine Learning

6.3. Evaluasi model Machine Learning

6.4. Pemanfaatan trained model machine learning

6.5. Deploy model Machine Learning | Dumping dan Loading model Machine Learning



03 Workflow dengan Scikit-Learn

Persiapan Dataset

```
[1] from sklearn.datasets import load_iris

    iris = load_iris()

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

Splitting Dataset: Training & Testing Set

```
[2] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, #feature
                                                        y, #target
                                                        test_size=0.3, #test set 30%, train set 70%
                                                        random_state=1) #acak & konsisten  untuk replikasi
```

Double-click (or enter) to edit

```
[3] from sklearn.neighbors import KNeighborsClassifier

    model.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

Evaluasi Model

```
[4] 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

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

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

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

```
Hasil prediksi : ['versicolor', 'virginica']
```

Dump & Load Trained Model

Dumping Model Machine Learning menjadi file joblib

```
import joblib

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

```
['iris_classifier_knn.jonlib']
```

Loading Model Machine Learning dari file joblib

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

```
[ ] Start coding or generate with AI.
```

✓ 0s completed at 8:15 PM

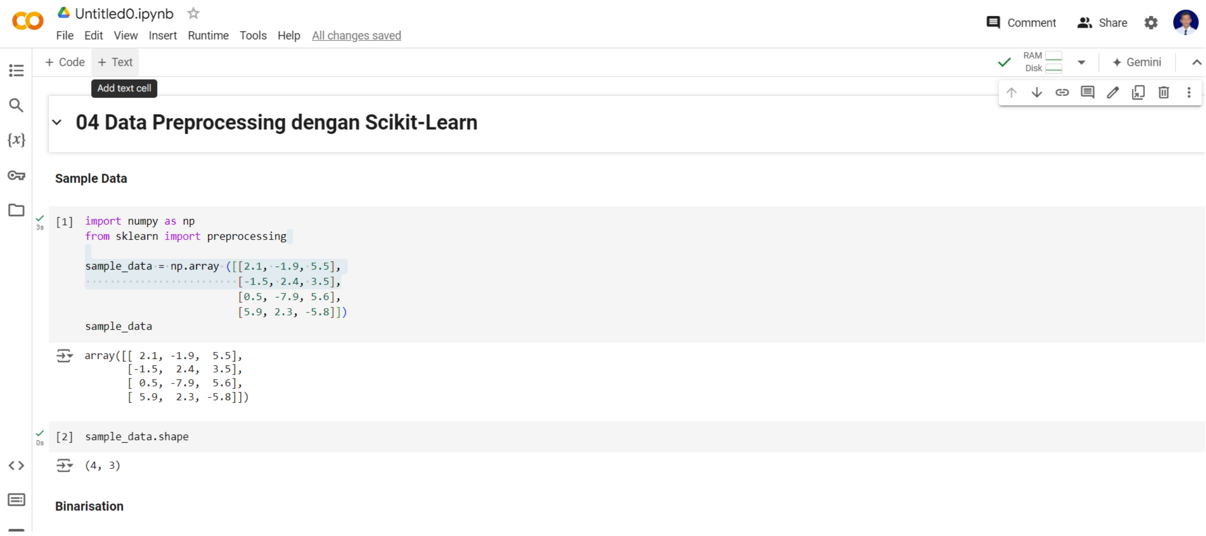
7.0. Lakukan praktek dari https://youtu.be/smNnhEd26Ek?feature=shared . Praktek tersebut yaitu:

7.1. Persiapan sample dataset

7.2. Teknik data preprocessing 1: binarisation

7.3. Teknik data preprocessing 2: scaling

7.4. Teknik data preprocessing 3: normalisation



04 Data Preprocessing dengan Scikit-Learn

Sample Data

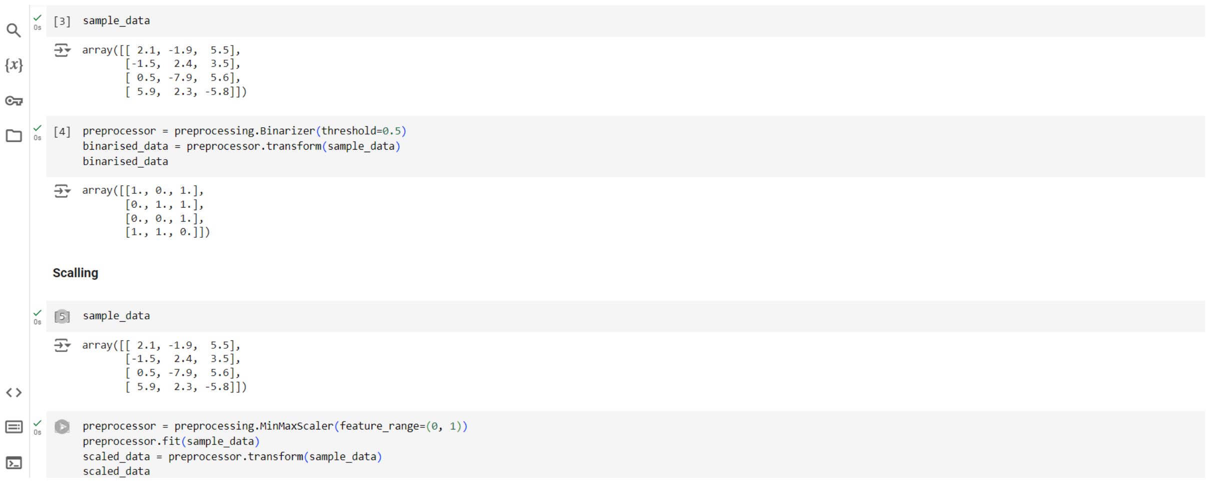
```
[1] import numpy as np
    from sklearn import preprocessing

    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
```

```
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
```

```
(4, 3)
```

Binarisation



```
[3] sample_data
```

```
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
```

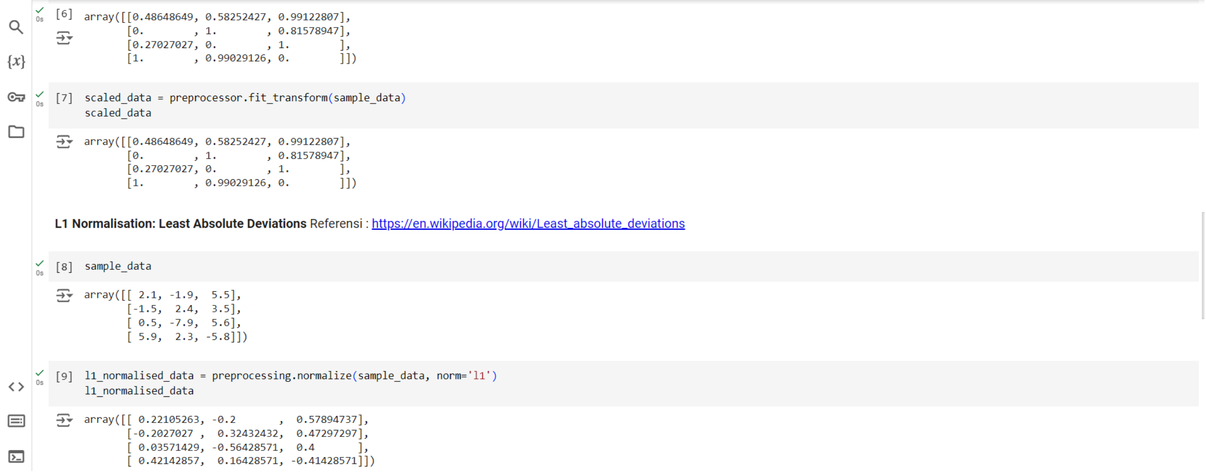
```
array([[1., 0., 1.],
       [0., 1., 1.],
       [0., 0., 1.],
       [1., 1., 0.]])
```

Scalling

```
[5] sample_data
```

```
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]])
```

```
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
```

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

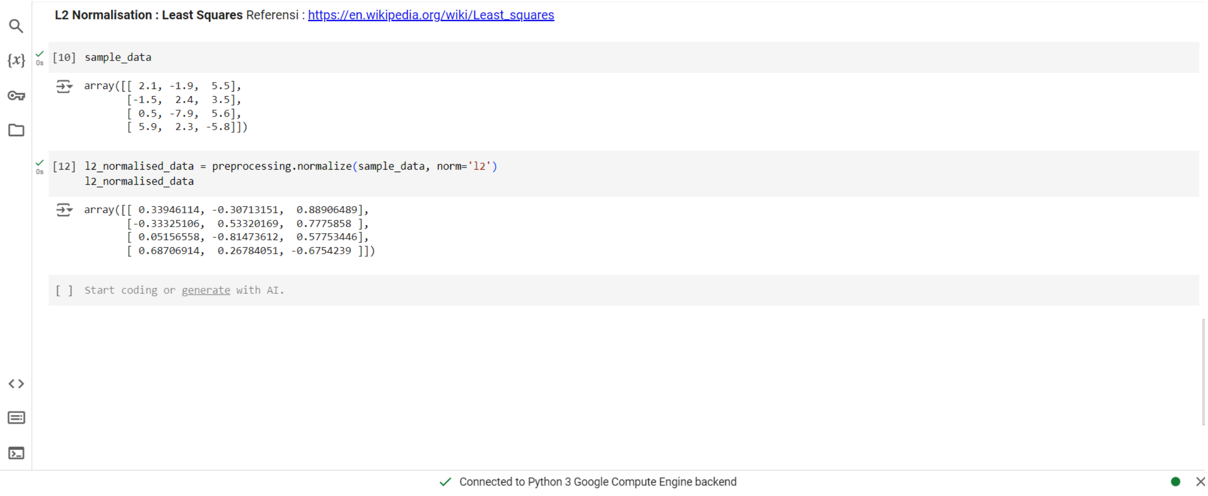
L1 Normalisation: Least Absolute Deviations Referensi : https://en.wikipedia.org/wiki/Least_absolute_deviations

```
[8] sample_data
```

```
    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
```

```
    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
```

```
    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]])
```

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

```
    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 ]])
```

```
[ ] Start coding or generate with AI.
```

Connected to Python 3 Google Compute Engine backend