Nama : Suci Maria

Kelas : TIF-A1

Npm : 41155050210005

TUGAS PERTEMUAN 1

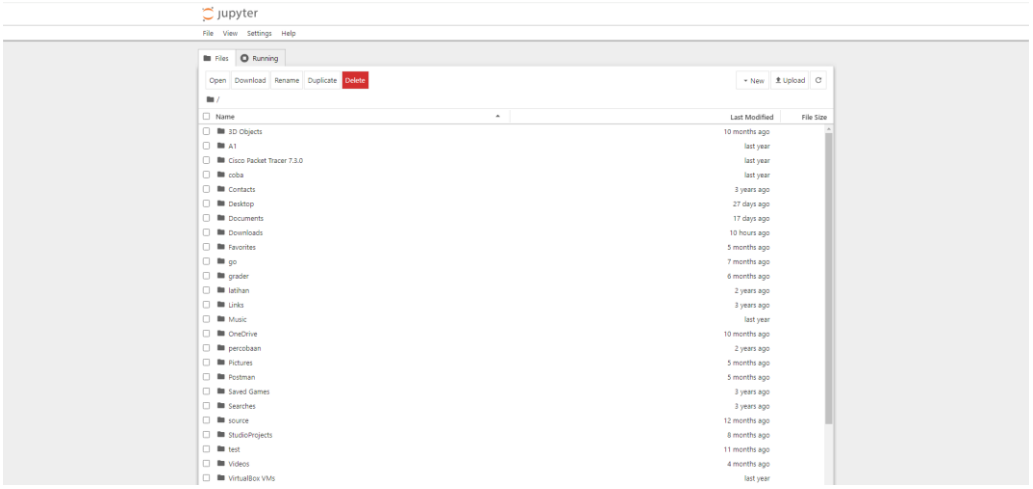
1. Instalasi Jupyter Notebook, Lakukan download dan instalasi :

1.1.

- Instalasi Python

```
C:\Users\DELL>python --version
Python 3.10.6
```

- Instalasi Jupyter Notebook



1.1. Jupyter Notebook (https://jupyter.org/), dan Library python seperti NumPy, SciPy, Pandas, Matplotlib, Seaborn, Scikit-learn.

- **NumPy**

```
[1]: !pip install numpy

Collecting numpy
  Downloading numpy-2.1.2-cp310-cp310-win_amd64.whl (12.9 MB)
     ---------------------------------------- 12.9/12.9 MB 662.2 kB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-2.1.2
```

- **SciPy**

```
[5]: pip install scipy

Collecting scipy
  Downloading scipy-1.14.1-cp310-cp310-win_amd64.whl.metadata (60 kB)
Requirement already satisfied: numpy<2.3,>=1.23.5 in c:\laragon\bin\python\python-3.10\lib\site-packages (from scipy) (2.1.2)
Downloading scipy-1.14.1-cp310-cp310-win_amd64.whl (44.8 MB)

Installing collected packages: scipy
Successfully installed scipy-1.14.1
Note: you may need to restart the kernel to use updated packages.
```

- **Pandas**

```
[4]: !pip install pandas

Collecting pandas
  Downloading pandas-2.2.3-cp310-cp310-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.22.4 in c:\laragon\bin\python\python-3.10\lib\site-packages (from pandas) (2.1.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\laragon\bin\python\python-3.10\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\laragon\bin\python\python-3.10\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading pandas-2.2.3-cp310-cp310-win_amd64.whl (11.6 MB)
   ---------------------------------------- 0.0/11.6 MB ? eta -:--:--
    --------------------------------------- 0.3/11.6 MB ? eta -:--:--
   -- ------------------------------------- 0.8/11.6 MB 2.0 MB/s eta 0:00:06
   --- ------------------------------------ 1.0/11.6 MB 2.2 MB/s eta 0:00:05
   ---- ----------------------------------- 1.3/11.6 MB 1.9 MB/s eta 0:00:06
   ------- -------------------------------- 2.1/11.6 MB 2.0 MB/s eta 0:00:05
   --------- ------------------------------ 2.6/11.6 MB 2.1 MB/s eta 0:00:05
   ---------- ----------------------------- 3.1/11.6 MB 2.2 MB/s eta 0:00:04
   ------------ --------------------------- 3.7/11.6 MB 2.2 MB/s eta 0:00:04
   -------------- ------------------------- 4.2/11.6 MB 2.2 MB/s eta 0:00:04
   ---------------- ----------------------- 5.0/11.6 MB 2.4 MB/s eta 0:00:03
   ------------------ --------------------- 5.5/11.6 MB 2.4 MB/s eta 0:00:03
   -------------------- ------------------- 6.0/11.6 MB 2.4 MB/s eta 0:00:03
   ---------------------- ----------------- 6.8/11.6 MB 2.5 MB/s eta 0:00:02
   ------------------------ --------------- 7.3/11.6 MB 2.5 MB/s eta 0:00:02
   --------------------------- ------------ 7.6/11.6 MB 2.4 MB/s eta 0:00:02
   ----------------------------- ---------- 8.7/11.6 MB 2.5 MB/s eta 0:00:02
   ------------------------------- -------- 9.4/11.6 MB 2.6 MB/s eta 0:00:01
   --------------------------------- ----- 10.0/11.6 MB 2.7 MB/s eta 0:00:01
   ----------------------------------- -- 10.7/11.6 MB 2.7 MB/s eta 0:00:01
   ------------------------------------- - 11.3/11.6 MB 2.7 MB/s eta 0:00:01
   ---------------------------------------  11.5/11.6 MB 2.7 MB/s eta 0:00:01
   ---------------------------------------- 11.6/11.6 MB 2.5 MB/s eta 0:00:00
Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
Downloading tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.3 pytz-2024.2 tzdata-2024.2
```

- **Matplotlib**

[7]:
```
pip install matplotlib
```

```
Collecting matplotlib
  Downloading matplotlib-3.9.2-cp310-cp310-win_amd64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.0-cp310-cp310-win_amd64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.54.1-cp310-cp310-win_amd64.whl.metadata (167 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.7-cp310-cp310-win_amd64.whl.metadata (6.4 kB)
Requirement already satisfied: numpy>=1.23 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib) (2.1.2)
Requirement already satisfied: packaging>=20.0 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib) (24.1)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-10.4.0-cp310-cp310-win_amd64.whl.metadata (9.3 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.1.4-py3-none-any.whl.metadata (5.1 kB)
Requirement already satisfied: python-dateutil>=2.7 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in c:\laragon\bin\python\python-3.10\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Downloading matplotlib-3.9.2-cp310-cp310-win_amd64.whl (7.8 MB)
Downloading pyparsing-3.1.4-py3-none-any.whl (104 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, contourpy, matplotlib
Successfully installed contourpy-1.3.0 cycler-0.12.1 fonttools-4.54.1 kiwisolver-1.4.7 matplotlib-3.9.2 pillow-10.4.0 pyparsing-3.1.4
Note: you may need to restart the kernel to use updated packages.
```

- Seaborn

[8]:
```
pip install seaborn
```

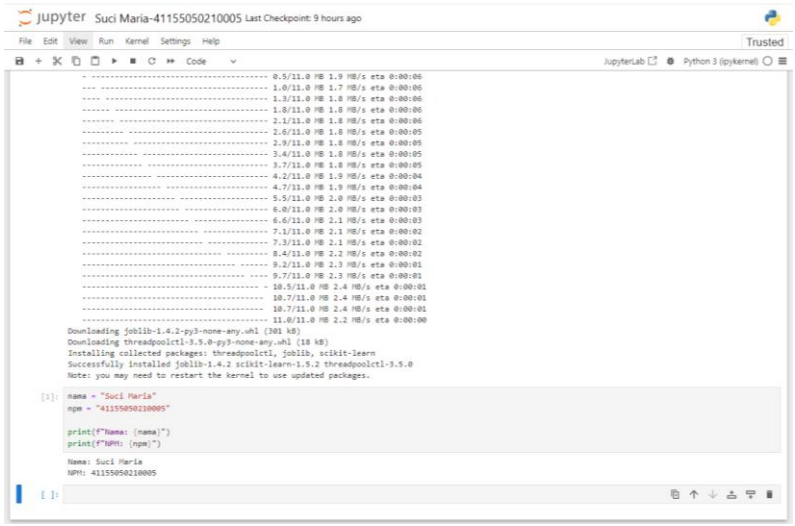
```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\laragon\bin\python\python-3.10\lib\site-packages (from seaborn) (2.1.2)
Requirement already satisfied: pandas>=1.2 in c:\laragon\bin\python\python-3.10\lib\site-packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\laragon\bin\python\python-3.10\lib\site-packages (from seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (4.54.1)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\laragon\bin\python\python-3.10\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.9.
0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\laragon\bin\python\python-3.10\lib\site-packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in c:\laragon\bin\python\python-3.10\lib\site-packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: six>=1.5 in c:\laragon\bin\python\python-3.10\lib\site-packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seabo
rn) (1.16.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
Note: you may need to restart the kernel to use updated packages.
```

- Scikit-learn

[9]:
```
pip install scikit-learn
```

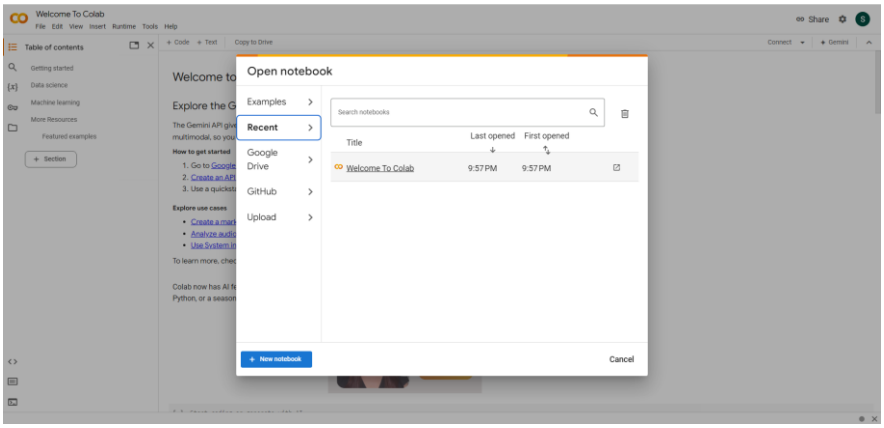
```
Collecting scikit-learn
  Downloading scikit_learn-1.5.2-cp310-cp310-win_amd64.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.19.5 in c:\laragon\bin\python\python-3.10\lib\site-packages (from scikit-learn) (2.1.2)
Requirement already satisfied: scipy>=1.6.0 in c:\laragon\bin\python\python-3.10\lib\site-packages (from scikit-learn) (1.14.1)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Downloading threadpoolctl-3.5.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.5.2-cp310-cp310-win_amd64.whl (11.0 MB)
   ---------------------------------------- 0.0/11.0 MB ? eta -:--:--
    --------------------------------------- 0.3/11.0 MB ? eta -:--:--
   - -------------------------------------- 0.5/11.0 MB 1.9 MB/s eta 0:00:06
   --- ------------------------------------ 1.0/11.0 MB 1.7 MB/s eta 0:00:06
   ---- ----------------------------------- 1.3/11.0 MB 1.8 MB/s eta 0:00:06
   ------ --------------------------------- 1.8/11.0 MB 1.8 MB/s eta 0:00:06
   ------- -------------------------------- 2.1/11.0 MB 1.8 MB/s eta 0:00:06
   --------- ------------------------------ 2.6/11.0 MB 1.8 MB/s eta 0:00:05
   ---------- ----------------------------- 2.9/11.0 MB 1.8 MB/s eta 0:00:05
   ------------ --------------------------- 3.4/11.0 MB 1.8 MB/s eta 0:00:05
   ------------- -------------------------- 3.7/11.0 MB 1.8 MB/s eta 0:00:05
   --------------- ------------------------ 4.2/11.0 MB 1.9 MB/s eta 0:00:04
   ---------------- ----------------------- 4.7/11.0 MB 1.9 MB/s eta 0:00:04
   -------------------- ------------------- 5.5/11.0 MB 2.0 MB/s eta 0:00:03
   --------------------- ------------------ 6.0/11.0 MB 2.0 MB/s eta 0:00:03
   ----------------------- ---------------- 6.6/11.0 MB 2.1 MB/s eta 0:00:03
   ------------------------- -------------- 7.1/11.0 MB 2.1 MB/s eta 0:00:02
   -------------------------- ------------- 7.3/11.0 MB 2.1 MB/s eta 0:00:02
   ------------------------------ --------- 8.4/11.0 MB 2.2 MB/s eta 0:00:02
   ---------------------------------- ------ 9.2/11.0 MB 2.3 MB/s eta 0:00:01
   ----------------------------------- ---- 9.7/11.0 MB 2.3 MB/s eta 0:00:01
   ---------------------------------------- - 10.5/11.0 MB 2.4 MB/s eta 0:00:01
   ---------------------------------------  10.7/11.0 MB 2.4 MB/s eta 0:00:01
   ---------------------------------------  10.7/11.0 MB 2.4 MB/s eta 0:00:01
   ---------------------------------------- 11.0/11.0 MB 2.2 MB/s eta 0:00:00
Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Installing collected packages: threadpoolctl, joblib, scikit-learn
Successfully installed joblib-1.4.2 scikit-learn-1.5.2 threadpoolctl-3.5.0
Note: you may need to restart the kernel to use updated packages.
```

1.2. Tuliskan nama dan nomor NPM anda pada Jupiter Notebook.

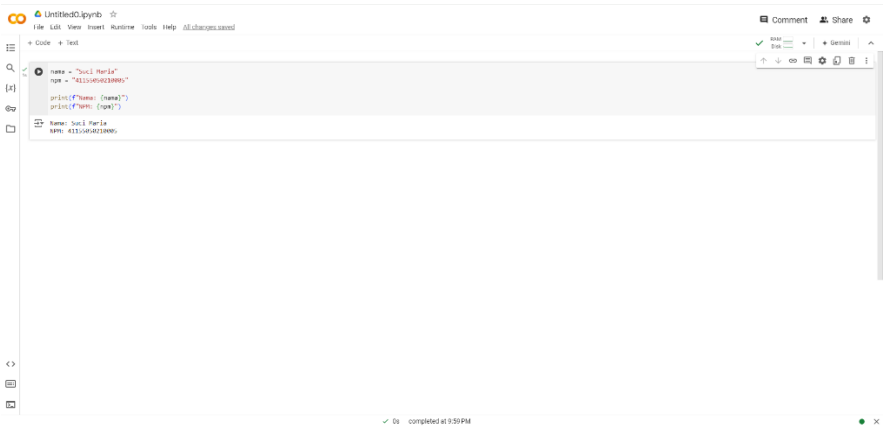


2. Menggunakan Google Collab, Lakukan

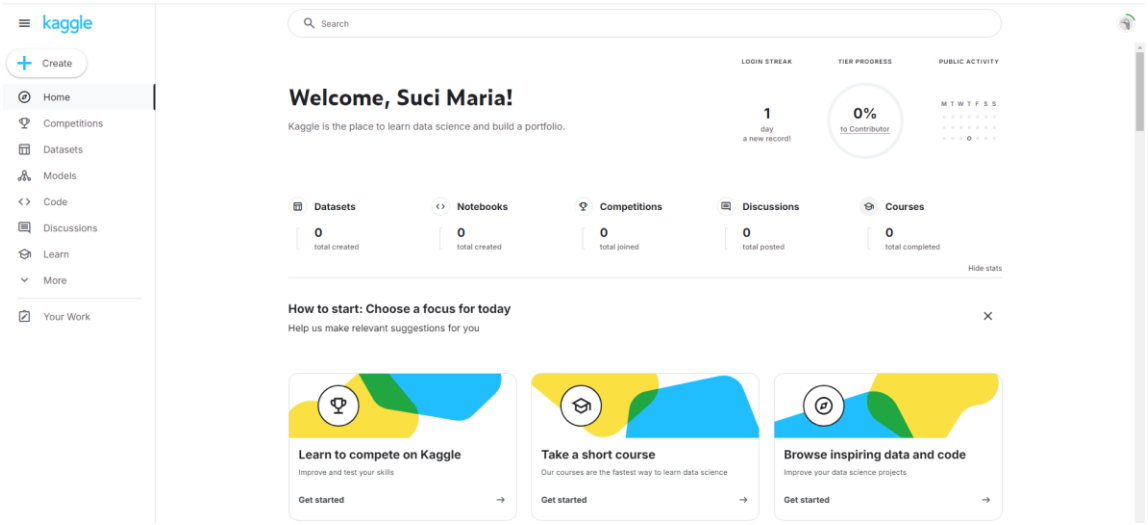
2.1. Gunakan Google Colab (https://colab.research.google.com/).



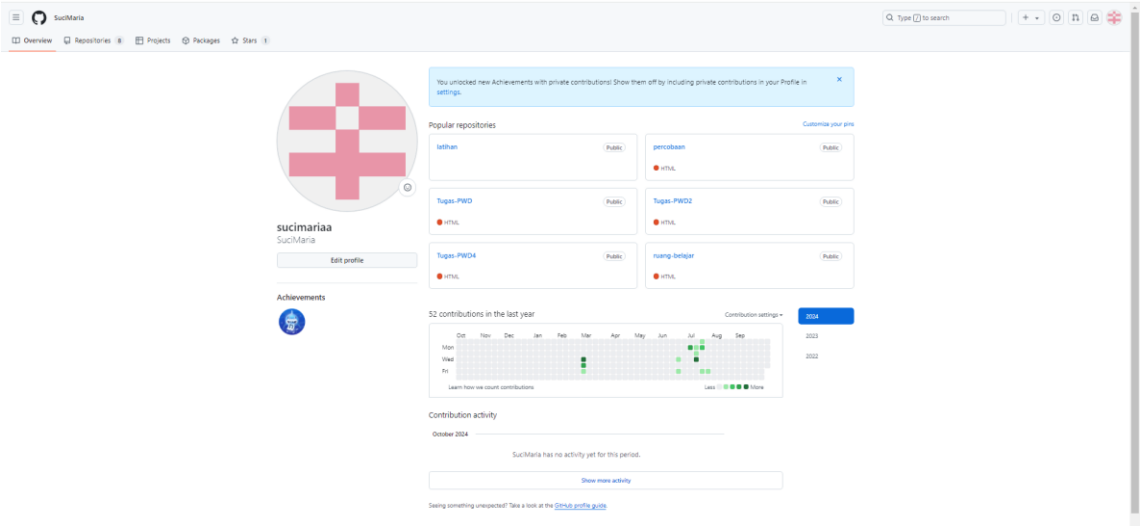
2.2. Tuliskan nama dan nomor NPM anda pada Google Colab.



3. **Buatlah akun di https://www.kaggle.com/ .**



4. **Buatlah akun di https://github.com/ .**



5. **Lakukan praktek dari https://youtu.be/mSO2hJln0OY?feature=shared . Praktek tersebut yaitu:**

5.1. Load sample dataset

- **From sklearn.datasets import load_iris**

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

     iris = load_iris()
     iris
```

```
[3]: {'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],
```

- **Iris.keys()**

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

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

5.2. Metadata | Deskripsi dari sample dataset

```
[6]: 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.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

    The famous Iris database, first used by Sir R.A. Fisher. The dataset is taken
    from Fisher's paper. Note that it's the same as in R, but not as in the UCI
    Machine Learning Repository, which has two wrong data points.

    This is perhaps the best known database to be found in the
    pattern recognition literature.  Fisher's paper is a classic in the field and
    is referenced frequently to this day.  (See Duda & Hart, for example.)  The
    data set contains 3 classes of 50 instances each, where each class refers to a
    type of iris plant.  One class is linearly separable from the other 2; the
```

5.3. Explanatory & Response Variables | Features & Target

➢ Explanatory Variables (Features)

```
[7]: X = iris.data
     X.shape
     # X
```

```
[7]: (150, 4)
```

```
[8]: X = iris.data
     # X.shape
     X
```

```
[8]: 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],
```

➢ Response Variable (Target)

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

```
[9]: (150,)
```

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

```
[10]: 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, 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, 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, 2])
```

5.4. Feature & Target Names

➢ Feature

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

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

➢ Target

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

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

5.5. Visualisasi Data

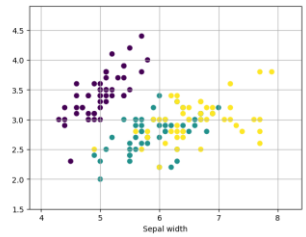
```
[13]: import matplotlib.pyplot as plt

      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.xlabel('Sepal width')

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



5.6. Training Set & Testing Set

```
[15]: 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 train: {y_test.shape}')
```

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

5.7. Load sample dataset sebagai Pandas Data Frame

```
[16]: iris = load_iris(as_frame=True)

iris_features_df = iris.data
iris_features_df
```

[16]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
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

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

6.1.Persiapan dataset | Loading & splitting dataset

- **Load Sample Dataset Iris Dataset**

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

iris = load_iris()

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

- **Splitting Dataset Training & Testing Set**

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

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.4,
                                                    random_state=1)
```

6.2. Training model Machine Learning

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

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

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

6.3. Evaluasi model Machine Learning

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

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

     Accuracy: 0.9833333333333333
```

6.4. Pemanfaatan trained model machine learning

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

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

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

- **Dumping Model Machine Learning menjadi file joblib**

```
[8]: import joblib

     joblib.dump(model, 'iris_classifier_knn.joblib')

[8]: ['iris_classifier_knn.joblib']
```

Suci Maria-41155050210005.ipynb	17 minutes ago	103.8 KB
TugasNo6.1.ipynb	1 minute ago	21 KB
Untitled.ipynb	11 hours ago	72 B
Untitled1.ipynb	11 hours ago	72 B
Untitled2.ipynb	16 minutes ago	617 B
cobain	last year	3 B
go.mod	5 months ago	43 B
iris_classifier_knn.joblib	2 minutes ago	8.7 KB
Workspace.sws	2 years ago	682 B

- **Loading Model Machine Learning dari file joblib**

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

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

7.1. Persiapan sample dataset

```
[16]: 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

[16]: 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]])
```

```
[17]: sample_data.shape
```

```
[17]: (4, 3)
```

7.2. Teknik data preprocessing 1: binarisation

```
[18]: sample_data
```

```
[18]: 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]])
```

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

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

7.3. Teknik data preprocessing 2: scaling

```
[23]: sample_data
```

```
[23]: 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]])
```

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

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

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

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

7.4. Teknik data preprocessing 3: normalisation

- **L1 Normalisation: Least Absolute Deviations**

```
[26]: sample_data
```

```
[26]: 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]])
```

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

```
[27]: 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**

```
[28]: sample_data
```

```
[28]: 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]])
```

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

```
[29]: 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 ]])
```