

Nama : Hafsah Islamiati Ayuningtyas

NPM : 41155050210039

Kelas : INF-A2

Mata Kuliah : Machine Learning

Tugas Pertemuan 1

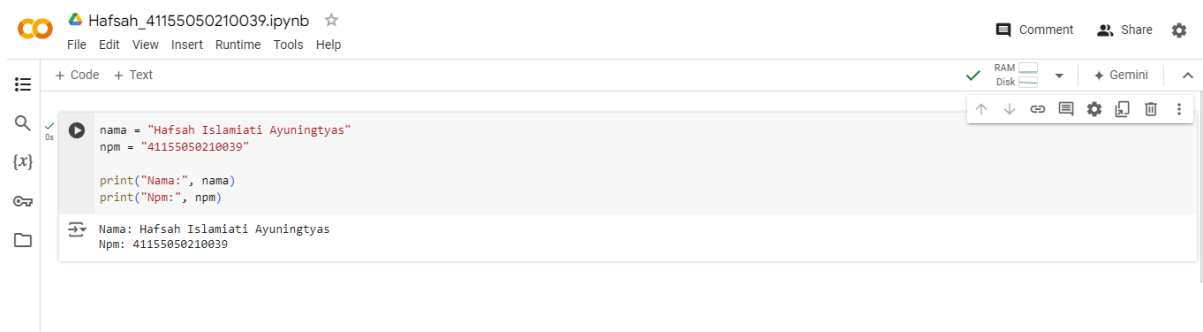
1.2. Tuliskan nama dan nomor NPM anda pada Jupiter Notebook

```
nama = "Hafsah Islamiati Ayuningtyas"
npm = "41155050210039"

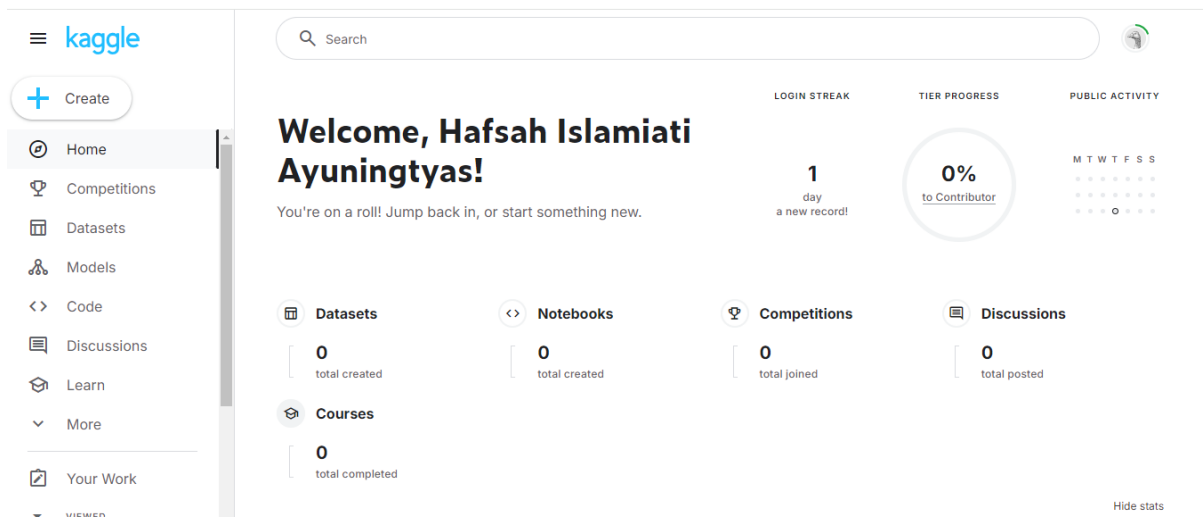
print("Nama:", nama)
print("Npm:", npm)
```

Nama: Hafsah Islamiati Ayuningtyas
Npm: 41155050210039

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



3.0 Akun Kaggle



4.0 Akun GitHub

The screenshot shows the GitHub profile of user hafsahislamiati. The profile includes a circular avatar with a blue and white geometric pattern, the username hafsahislamiati, and an 'Edit profile' button. Below the profile information, there are sections for 'Popular repositories' and '2 contributions in the last year'.

Popular repositories:

- latihan (Public)
- tugas2-pwd-tag-html (Public) - HTML
- tugas2-pwd-html (Public) - HTML
- template-website (Public) - HTML
- msib5_latihan_git (Public) - PHP
- msib5-latihan-web (Public) - HTML

2 contributions in the last year:

The contribution calendar shows activity for the year 2023, with a focus on the month of September. The calendar grid shows contributions for each day of the month.

5.1 Load Sample Dataset

The screenshot shows a JupyterLab notebook interface. The notebook is titled 'Hafsah_41155050210039' and shows the following code and output:

```
[2]: from sklearn.datasets import load_iris
iris = load_iris()
iris
```

The output of the code is a dictionary containing the Iris dataset:

```
[2]: {'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]]),
'target': array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2]),
'feature_names': ('sepal_length', 'sepal_width', 'petal_length', 'petal_width'),
'target_names': ('species'),
'DESCRIPTION': 'The Iris dataset is a classic dataset for machine learning. It contains 150 samples of Iris flowers, each with four features: sepal length, sepal width, petal length, and petal width. The target variable is the species of the flower, which can be 'setosa', 'versicolour', or 'virginica'.
```

The notebook also shows the keys of the dictionary and the keys of the 'data' dictionary:

```
[3]: iris.keys()
[3]: dict_keys(['data', 'target', 'feature_names', 'target_names', 'DESCR', 'filename', 'data_module'])
```

5.2 Metadata | Deskripsi dari sample dataset

```
JupyterLab
Python 3 (ipykernel)

[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
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 (MARSHALLXPL@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.
```

5.3. Explanatory & Response Variables | Features & Target

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

[5]: (150, 4)
```

Jupyter Hafsah_41155050210039 Last Checkpoint: 17 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[6]: X = iris.data
      X.shape
      X

[6]: 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]]

[8]: y = iris.target
      y.shape
      # y

[8]: (150,)
```

[]:

Activate
Go to Settings

5.6. Training Set & Testing Set

```
[16]: 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,)
```

5.7. Load sample dataset sebagai Pandas Data Frame

jupyter Hafsah_41155050210039 Last Checkpoint: 46 minutes ago

File Edit View Run Kernel Settings Help

Code

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

iris_features_df = iris.data
iris_features_df
```

	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

6.1. Persiapan dataset | Loading & splitting dataset

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

```
iris = load_iris()
```

```
x = iris.data
```

```
y = iris.target
```

```
[23]: 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

```
[25]:
```

```
from sklearn.neighbors import KNeighborsClassifier
```

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

```
model.fit(X_train, y_train)
```

```
[25]:
```

```
▼ KNeighborsClassifier ⓘ ?  
KNeighborsClassifier(n_neighbors=3)
```

6.3. Evaluasi model Machine Learning

```
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

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

array([1, 2])

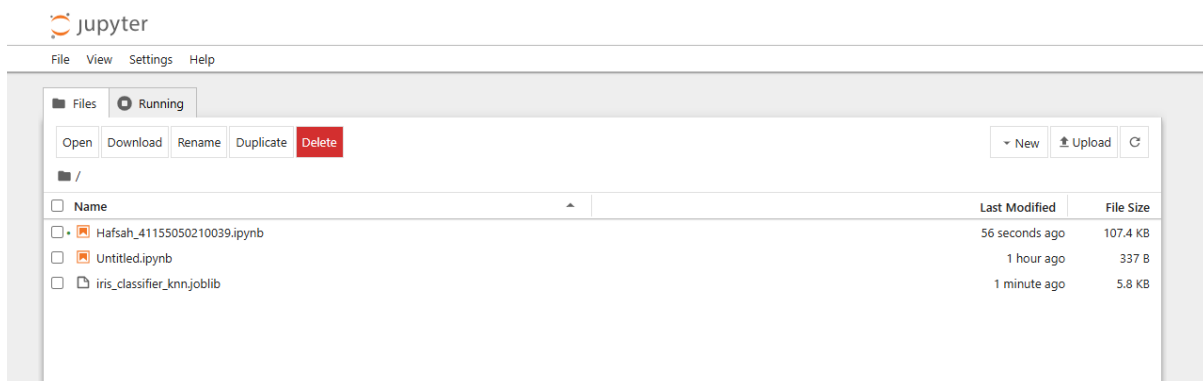
pred_species = [iris.target_names[p] for p in preds]
print(f'Hasil Prediksi : {pred_species}')

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

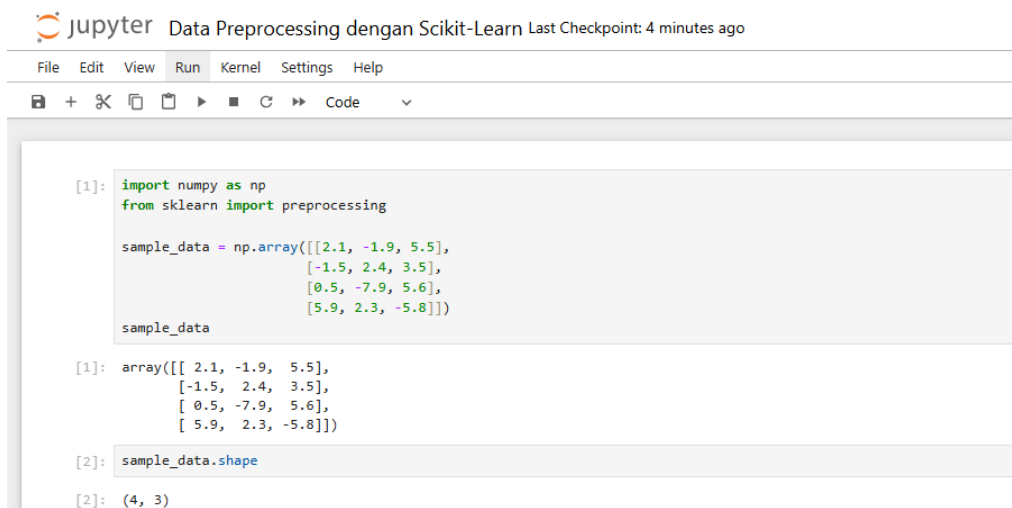
6.5. Deploy model Machine Learning | Dumping dan Loading model Machine Learning

```
[33]: import joblib
      joblib.dump(model, 'iris_classifier_knn.joblib')

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



7.1. Persiapan sample dataset



7.2. Teknik data preprocessing 1: 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]])

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

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

7.3. Teknik data preprocessing 2: scaling

```
[6]: sample_data

[6]: 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]])

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

[8]: scaled_data = preprocessor.fit_transform(sample_data)
      scaled_data

[8]: 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

```
[6]: sample_data

[6]: 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]])

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

[8]: scaled_data = preprocessor.fit_transform(sample_data)
      scaled_data

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


- L2 Normalisation: Least Squares

```
[11]: sample_data
```

```
[11]: 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
```

```
[12]: 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 ]])
```