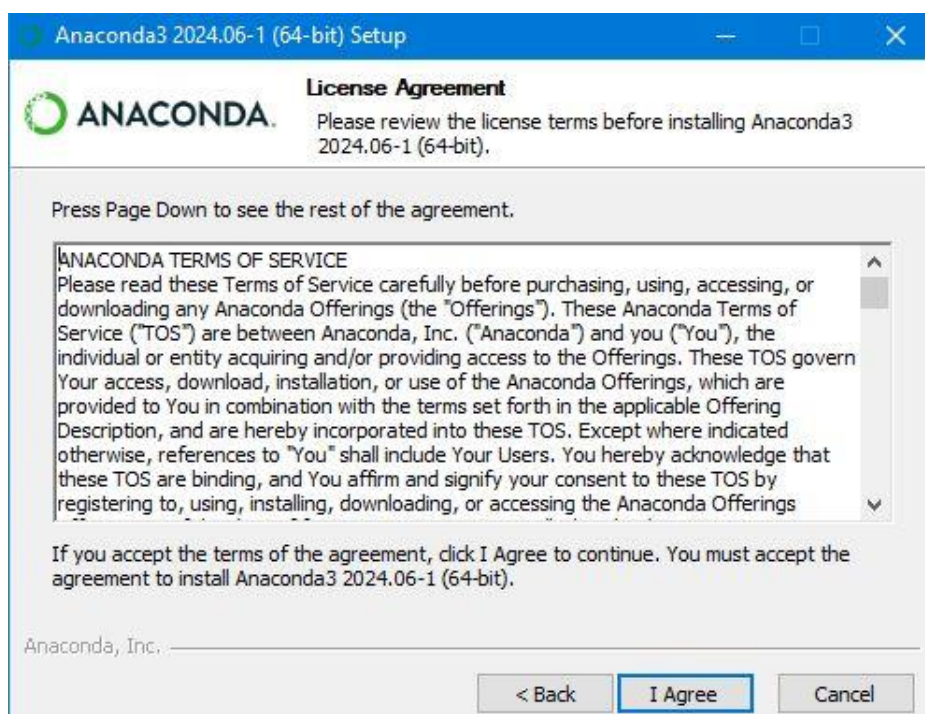
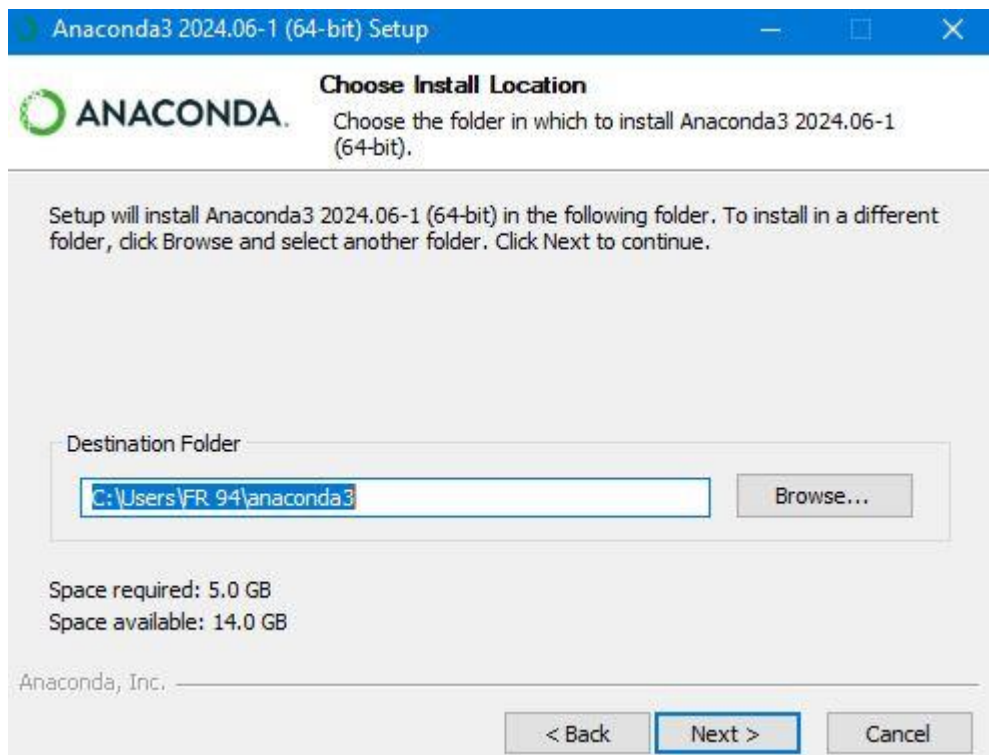
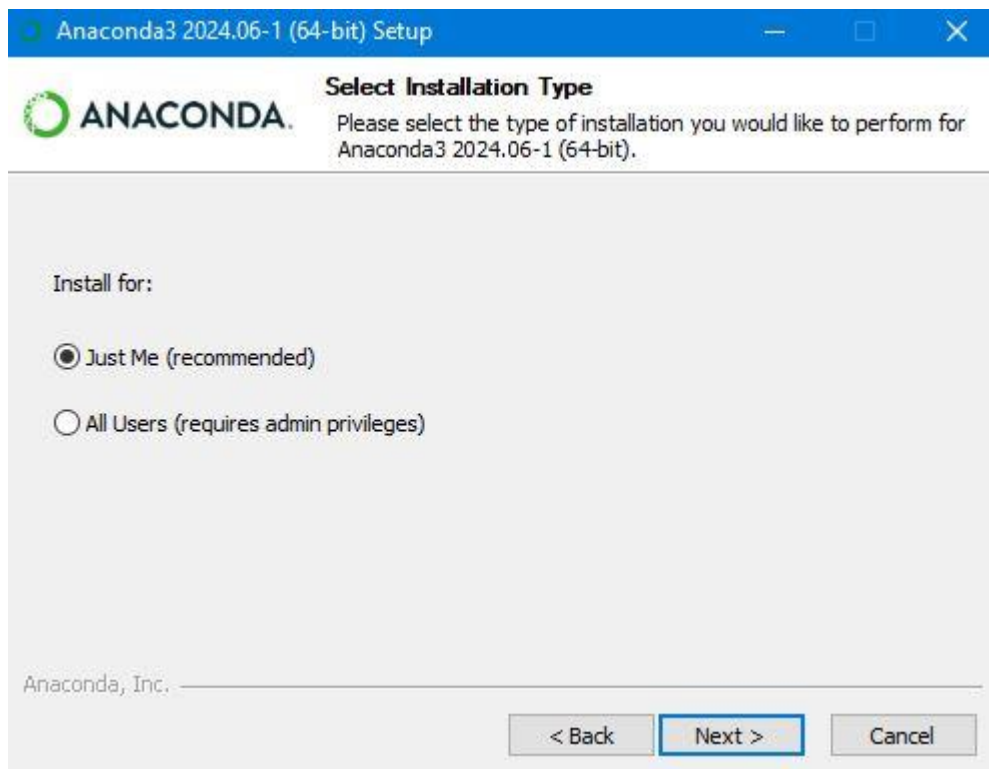


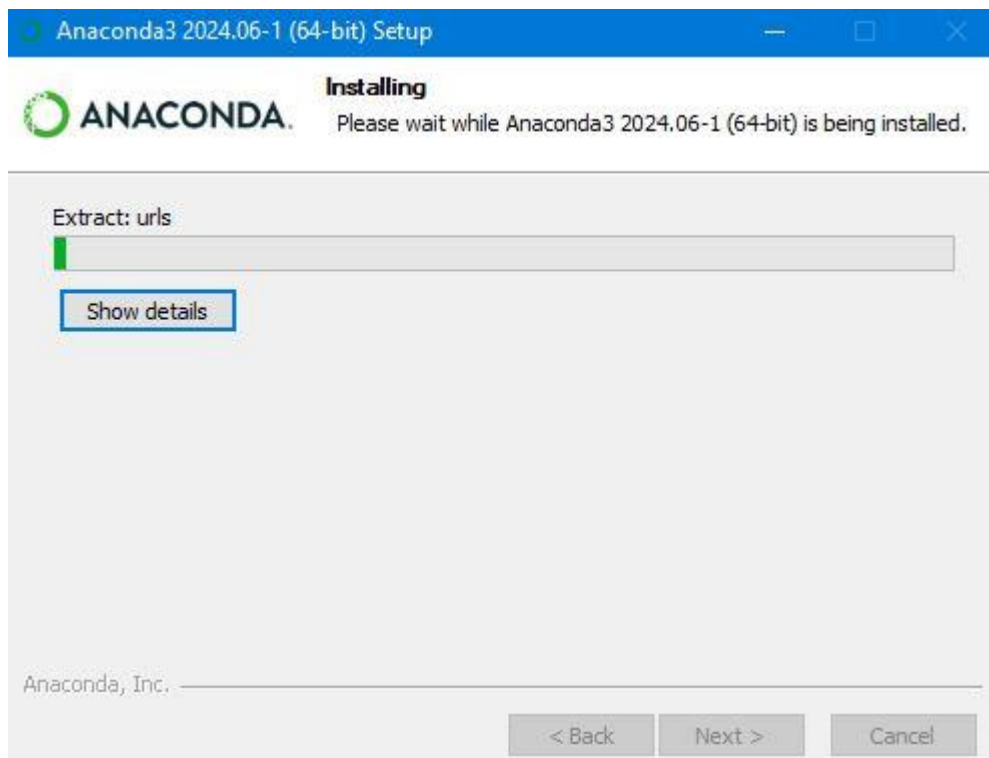
Tugas Machine Learning 1

Nama : Afif Naufal Hafidz
NPM : 41155050210067
Kelas : TIF A2

1. Instalasi Jupyter Notebook









Completing Anaconda3 2024.06-1 (64-bit) Setup

Thank you for installing Anaconda Distribution.

Here are some helpful resources to get you started. We recommend you bookmark the 'Getting Started with Anaconda Distribution' link so you can refer back to it later.

- ☒ Launch Anaconda Navigator
- ☒ Getting Started with Anaconda Distribution

< Back

Finish

Cancel

Anaconda Navigator

File Help

ANACONDA.NAVIGATOR

Update Now Connect

Home

Environments

Learning

Community

Anaconda Toolbox
Supercharged local notebooks.
Click the Toolbox tile to install.
Read the Docs

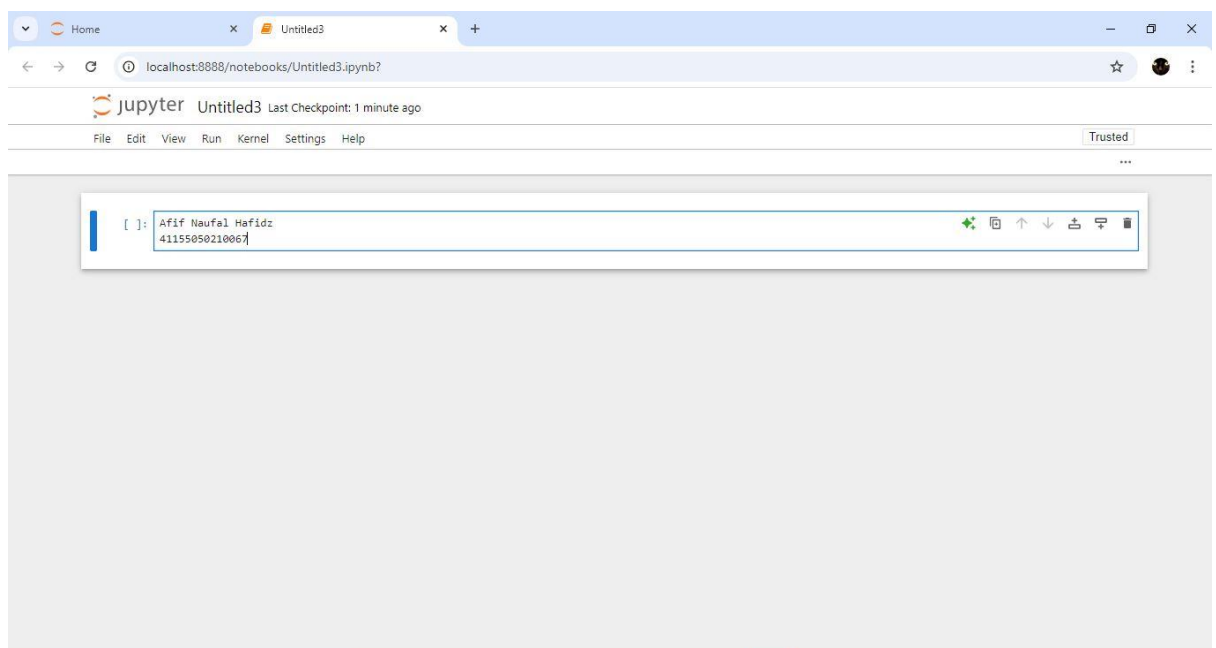
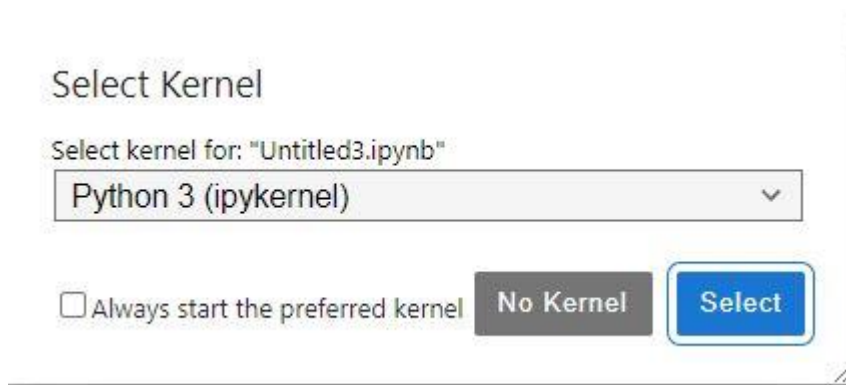
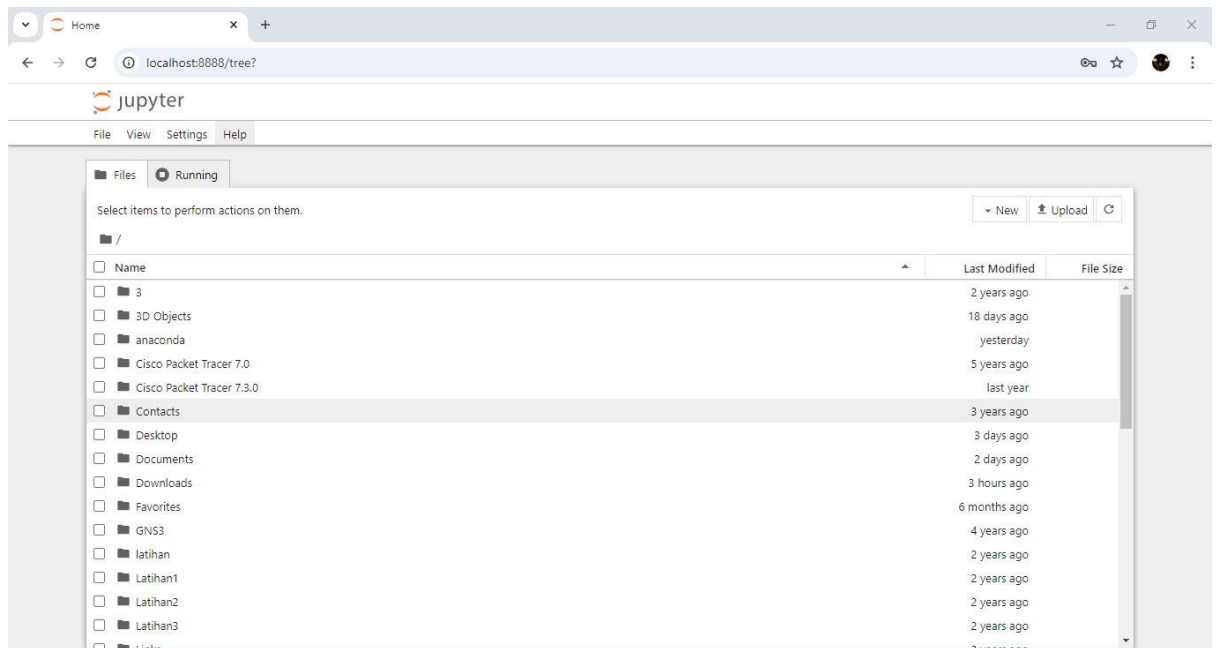
Documentation

Anaconda Blog

Twitter YouTube GitHub

All applications on base (root) Channels

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 JupyterLab 4.0.11 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 Jupyter Notebook 7.0.8 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated Launch
 Qt Console 5.5.1 PyQt GUI that supports inline figures. Launch	 Spyder 5.5.1 Scientific Python Development Launch	 VS Code 1.91.1 Streamlined code editor with support for Launch	 watsonx IBM watsonx IBM watsonx is an enterprise-ready AI Launch



2. Menggunakan Google Collab

The image displays two screenshots of the Google Colaboratory (Colab) web interface.

Top Screenshot: The browser address bar shows `colab.research.google.com/#scrollTo=C4HZx7Gndbrh`. The page title is "Selamat Datang di Colaboratory". The sidebar on the left contains a "Daftar isi" (Table of Contents) with links to "Memulai" (Getting started), "Ilmu data" (Data science), "Machine learning", "Referensi Lainnya" (Other references), and "Contoh bagus" (Good examples). The main content area displays "Selamat Datang di Colab!" and "(Baru) Coba Gemini API" (New! Try Gemini API), followed by a list of links: "Generate a Gemini API key", "Talk to Gemini with the Speech-to-Text API", "Gemini API: Quickstart with Python", "Gemini API code sample", "Compare Gemini with ChatGPT", and "More notebooks". Below the links, a message states: "Jika Anda sudah terbiasa dengan Colab, tonton video ini untuk mempelajari tabel interaktif, tampilan histori kode yang dieksekusi, dan palet perintah." (If you are already familiar with Colab, watch this video to learn about interactive tables, the code execution history view, and the command palette). A video thumbnail titled "3 Cool Google Colab Features" is visible.

Bottom Screenshot: The browser address bar shows `colab.research.google.com/drive/1R8CRoT-o8l0k_E4wgYi5NDriaWEW48HL#scrollTo=KypCOPnIE60Z`. The page title is "Untitled0.ipynb". The sidebar on the left contains a "Daftar isi" (Table of Contents) with links to "Kode" (Code) and "Teks" (Text). The main content area displays a single code cell with the text "Afif Naufal Hafidz" and the user ID "41155050210067".

3. Akun Kaggle

The screenshot shows the Kaggle user dashboard for 'Afif Naufal H'. The interface includes a left sidebar with navigation links: Home, Competitions, Datasets, Models, Code, Discussions, Learn, More, Your Work, and VIEWED. The main content area features a welcome message, a search bar, and a summary of user statistics. A 'LOGIN STREAK' section shows a streak of 1 day. A 'TIER PROGRESS' section shows 0% progress towards the next tier. A 'PUBLIC ACTIVITY' section shows a calendar for the current month. Below these, there are four cards for 'Datasets', 'Notebooks', 'Competitions', and 'Discussions', each showing 0 total created or joined. A 'Courses' card shows 0 total completed. A 'Hide stats' link is located at the bottom right of the dashboard.

kaggle.com

Search

Welcome, Afif Naufal H!

Jump back in, or start something new.

LOGIN STREAK: 1 Your longest is 5 days

TIER PROGRESS: 0% to Contributor

PUBLIC ACTIVITY: M T W T F S S

Datasets: 0 total created

Notebooks: 0 total created

Competitions: 0 total joined

Discussions: 0 total posted

Courses: 0 total completed

Hide stats

4. Akun Github

The screenshot shows the GitHub user profile for 'AfifNH'. The profile includes a circular profile picture, a bio, and a list of popular repositories. A notification banner at the top states: 'You unlocked new Achievements with private contributions! Show them off by including private contributions in your Profile in settings.' The repositories listed are: 'Latihan' (HTML), 'Tugas-PWD' (HTML), 'Aplikasi-Login' (HTML), 'Aplikasi-Login1' (Forked from dicodingacademy/Aplikasi-Login), 'App-Profile' (Sebuah aplikasi untuk menampilkan halaman profile pribadi dan ini juga proyek pertama untuk di deploy ke google cloud), and 'back-end-rute' (Sebuah kerangka back-end rute). The profile also shows an 'Edit profile' button and an 'Achievements' section.

github.com/AfifNH

AfifNH

Overview Repositories 8 Projects Packages Stars

You unlocked new Achievements with private contributions! Show them off by including private contributions in your Profile in settings.

Popular repositories

Latihan (Public) HTML

Tugas-PWD (Public) Kumpulan Tugas-Tugas Mata Kuliah Pemrograman Web Dasar HTML

Aplikasi-Login (Public) Sebuah aplikasi dengan fitur login HTML

Aplikasi-Login1 (Public) Forked from dicodingacademy/Aplikasi-Login

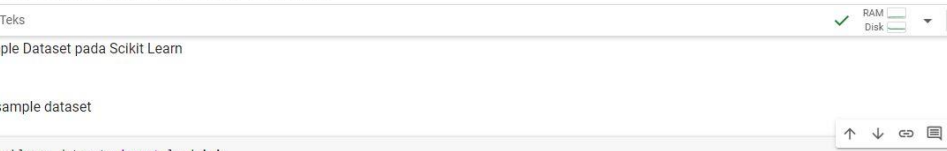
App-Profile (Public) Sebuah aplikasi untuk menampilkan halaman profile pribadi dan ini juga proyek pertama untuk di deploy ke google cloud.

back-end-rute (Public) Sebuah kerangka back-end rute

Edit profile

Achievements

5.1. Load Sample Dataset



The screenshot shows a JupyterLab environment with a file named 'Praktik.ipynb'. The interface includes a top bar with 'File', 'Edit', 'Lihat', 'Sisipkan', 'Runtime', 'Fitur', 'Bantuan', and 'Menyimpan...'. The main area displays a code cell with the following Python code:

```
from sklearn.datasets import load_iris

iris = load_iris()
iris
```

The output of the code is a list of 150 data points, each represented as a list of four values (sepal length, sepal width, petal length, petal width). The data is as follows:

```
[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. ],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
```

The bottom status bar indicates '0 d selesai pada 21.16'.

[illegible]

5.2. Metadata | Deskripsi Dari Sample Dataset

The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The code editor displays the following text:

```
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
0.0  5.4    3.76    1.76    0.9247
5.0  14.7   9.59    5.48    0.9392
15.0 30.9   18.01   7.92    0.9810
=====
```

The interface includes a top bar with a file explorer, a menu bar (File, Edit, View, Insert, Cell, Help), and a toolbar with icons for running, saving, and other actions. The status bar at the bottom indicates the execution time as 0.0 seconds.

```
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
latter are NOT linearly separable from each other.

.. dropdown:: References
```

```
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
latter are NOT linearly separable from each other.

.. dropdown:: References

- Fisher, R.A. "The use of multiple measurements in taxonomic problems"
  Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
  Mathematical Statistics" (John Wiley, NV, 1950).
- Duda, R.O., & Hart, P.E. (1973) Pattern Classification and Scene Analysis.
  (Q327.D83) John Wiley & Sons. ISBN 0-471-22361-1. See page 218.
- Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
  Structure and Classification Rule for Recognition in Partially Exposed
  Environments". IEEE Transactions on Pattern Analysis and Machine
  Intelligence, Vol. PAMI-2, No. 1, 67-71.
- Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule". IEEE Transactions
  on Information Theory, May 1972, 431-433.
- See also: 1988 MLC Proceedings, 54-64. Cheeseman et al's AUTOCLASS II
  conceptual clustering system finds 3 classes in the data.
- Many, many more ...
```

5.3. Explanatory & Response Variables | Features & Target

```
5.3. Explanatory & Response Variables | Features & Target

Explanatory Variables (Features)

[10] X = iris.data
X.shape
# X
(150, 4)

Response Variables (Target)

y = iris.target
y.shape
# y
(150,)
```

5.4. Feature & Target Names

```
Prak1.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan

+ Kode + Teks

feature_names = iris.feature_names
feature_names

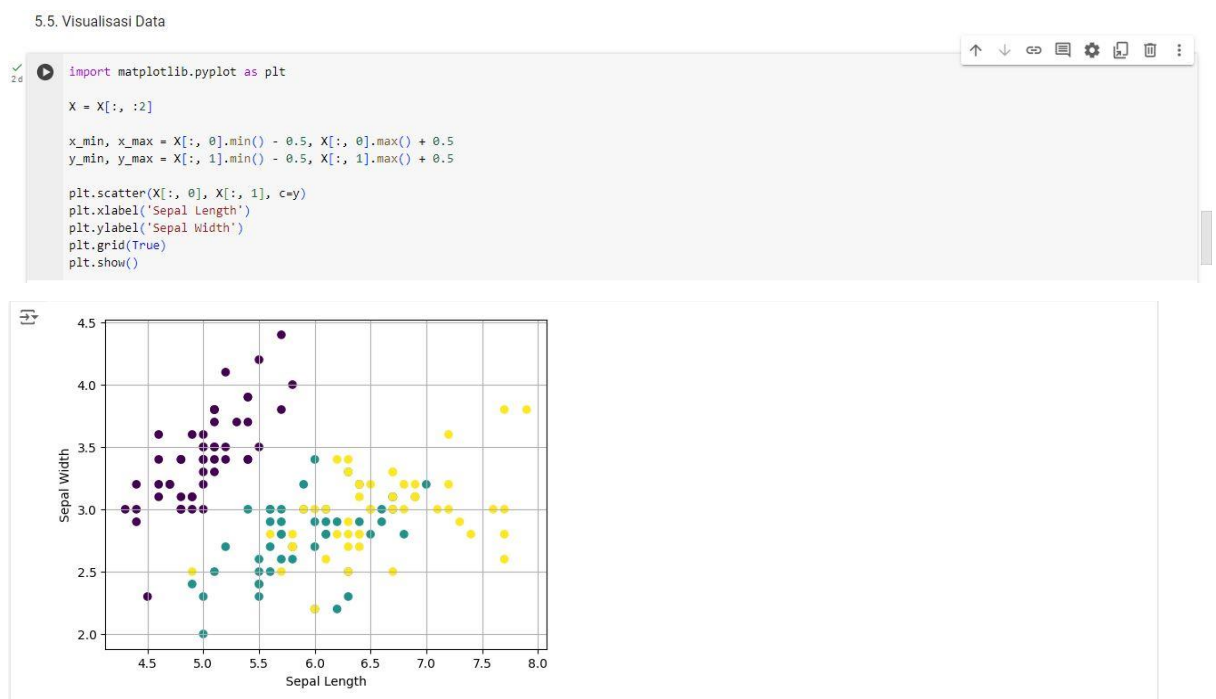
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']

target_names = iris.target_names
target_names

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

0 d selesai pada 21:38
```

5.5. Visualisasi Data



5.6. Training Set & Testing Set

```
5.6. Training Set & Testing Set

[17]: 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 shape: {X_train.shape}')
      print(f'X_test shape: {X_test.shape}')
      print(f'y_train shape: {y_train.shape}')
      print(f'y_test shape: {y_test.shape}')

      X_train shape: (105, 2)
      X_test shape: (45, 2)
      y_train shape: (105,)
      y_test shape: (45,)
```

5.7. Load Sample Dataset Sebagai Pandas Data Frame

```
iris = load_iris(as_frame=True)
iris_feature_df = iris.data
iris_feature_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 x 4 columns

0 d selesai pada 22:05

6. Machine Learning Workflow dengan Scikit Learn

6.1. Persiapan Dataset | Loading & Splitting Dataset

```
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
y = iris.target

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)
```

0 d selesai pada 22:41

6.2. Training Model Machine Learning

6.2. Training model Machine Learning

```
from sklearn.neighbors import KNeighborsClassifier

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

KNeighborsClassifier

KNeighborsClassifier(n_neighbors=3)

6.3. Evaluasi Model Machine Learning

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

```
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])

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

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

6.5. Deploy Model Machine Learning | Dumping Dan Loading Model Machine Learning

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

Dumping Model Machine Learning menjadi file joblib

import joblib

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

['iris_classifier_knn.joblib']

Loading Model Machine Learning dari file joblib

production_model = joblib.load('iris_classifier_knn.joblib')
```

7. Data Preprocessing dengan Scikit-Learn

7.1. Persiapan Sample Dataset

```
7. Data Preprocessing dengan Scikit-Learn

7.1. Persiapan sample dataset

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]])

sample_data.shape

(4, 3)
```

7.2. Teknik Data Preprocessing 1: Binarisation


```
prak3.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan
+ Kode + Teks
7.2. Teknik data preprocessing 1: binarisation
[x] [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]])
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.]])
0 d selesai pada 21.50
```

7.3. Teknik Data Preprocessing 2: Scaling

```
prak3.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan
+ Kode + Teks
7.3. Teknik data preprocessing 2: scaling
[x] [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
array([[0.48648649, 0.58252427, 0.99122807],
       [0.         ,  1.         , 0.81578947],
       [0.27027027, 0.         ,  1.         ],
       [1.         , 0.99029126, 0.         ]])
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.         ]])
7.4. Teknik data preprocessing 3: normalisation
0 d selesai pada 21.54
```

7.4. Teknik Data Preprocessing 3: Normalisation

```
prak3.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan telah disimpan
+ Kode + Teks
7.4. Teknik data preprocessing 3: normalisation
L1 Normalisation: Least Absolute Deviations
[x] [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]])
l1_normalised_data = preprocessing.normalize(sample_data, norm='l1')
l1_normalised_data
array([[ 0.22186263, -0.2       ,  0.57694737],
       [-0.2027027,  0.32432432,  0.47297297],
       [ 0.03971429, -0.56428571,  0.4       ],
       [ 0.42142857,  0.16428571, -0.41428571]])
L2 Normalisation: Least Squares
[x] [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]])
l2_normalised_data = preprocessing.normalize(sample_data, norm='l2')
l2_normalised_data
array([[ 0.33946114, -0.30713151,  0.88906489],
       [-0.33325106,  0.53320169,  0.775858  ],
       [ 0.05156558, -0.81478612,  0.57753446],
       [ 0.63786914,  0.26744651, -0.6754239  ]])
0 d selesai pada 22.02
```