Nama : Triansyah Amarullah Ahmad Prayoga

NPM : 41155050210034

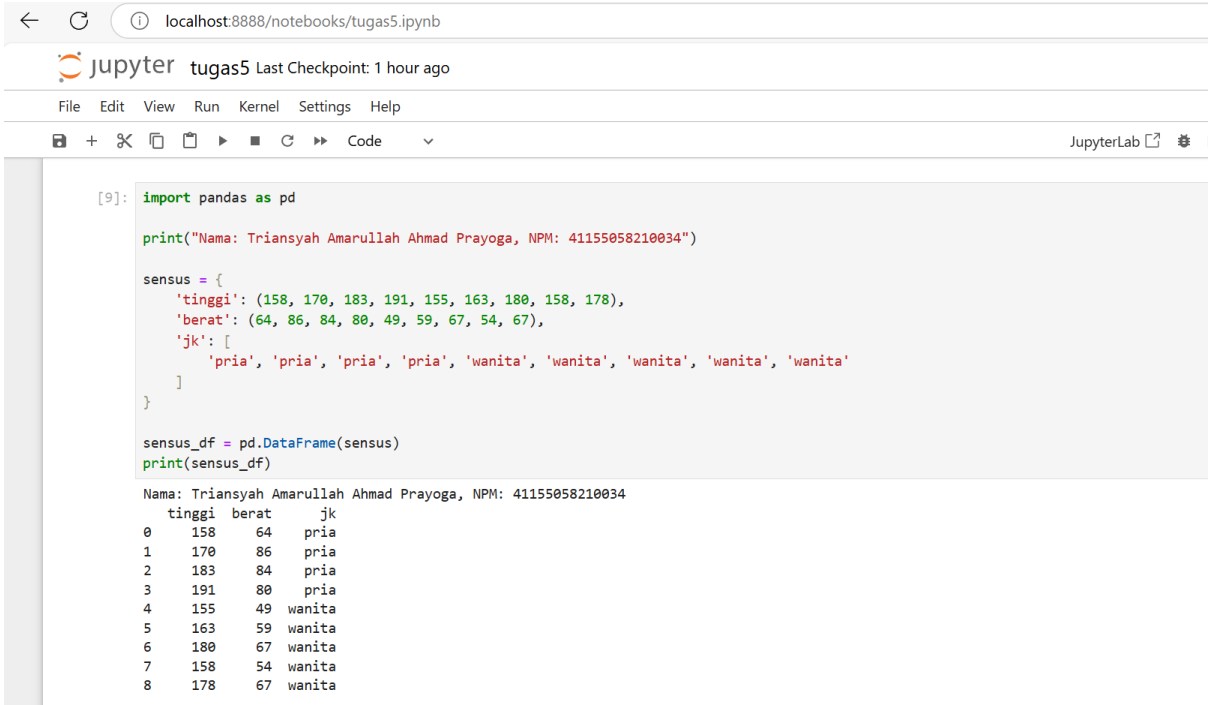
Kelas : TIF-A2 (2021)

Matkul : Machine Learning

# TUGAS 5

1.0. K-Nearest Neighbours (KNN). Lakukan praktik dari https://youtu.be/4zARMcgc7hA?si=x6RoHQXFF4NY76X8 , buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf, dari kegiatan ini:

1.1. Persiapan sample dataset




1.2. Visualisasi dataset

jupyter tugas5 Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Code

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
for jk, d in sensus_df.groupby('jk'):
    ax.scatter(d['tinggi'], d['berat'], label=jk)

plt.legend(loc='upper left')
plt.title('Sebaran Data Tinggi Badan, Berat Badan, dan Jenis Kelamin')
plt.xlabel('Tinggi Badan (cm)')
plt.ylabel('Berat Badan (kg)')
plt.grid(True)
plt.show()
```

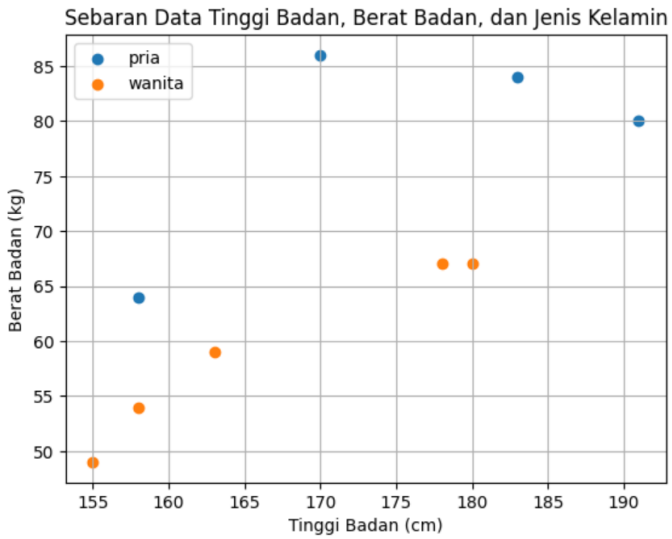
localhost:8888/notebooks/tugas5.ipynb

jupyter tugas5 Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Code

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```




1.3. Pengantar classification dengan K-Nearest Neighbours | KNN

K-Nearest Neighbors (KNN) adalah salah satu algoritma klasifikasi yang paling sederhana dan intuitif dalam machine learning. Algoritma ini bekerja berdasarkan prinsip bahwa data yang serupa cenderung berada di dekat satu sama lain dalam ruang fitur.

## 1.4. Preprocessing dataset dengan Label Binarizer

jupyter tugas5 Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Code JupyterLab Pyth

[11]:
```
import numpy as np

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")

X_train = np.array(sensus_df[['tinggi', 'berat']])
y_train = np.array(sensus_df['jk'])

print(f'X_train:\n{X_train}\n')
print(f'y_train:\n{y_train}\n')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
X_train:
[[158  64]
 [170  86]
 [183  84]
 [191  80]
 [155  49]
 [163  59]
 [180  67]
 [158  54]
 [178  67]]

y_train:
['pria' 'pria' 'pria' 'pria' 'wanita' 'wanita' 'wanita' 'wanita' 'wanita']
```

localhost:8888/notebooks/tugas5.ipynb

jupyter tugas5 Last Checkpoint: 28 minutes ago

File Edit View Run Kernel Settings Help

Code Jupy

```
y_train:
['pria' 'pria' 'pria' 'pria' 'wanita' 'wanita' 'wanita' 'wanita' 'wanita']
```

[10]:
```
from sklearn.preprocessing import LabelBinarizer

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")

lb = LabelBinarizer()
y_train = lb.fit_transform(y_train)

print(f'y_train:\n{y_train}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
y_train:
[[0]
 [0]
 [0]
 [0]
 [1]
 [1]
 [1]
 [1]
 [1]]
```

[ ]:

```
[11]: y_train = y_train.flatten()

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")

print(f'y_train:\n{y_train}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
y_train:
[0 0 0 0 1 1 1 1 1]
```

1.5. Training KNN Classification Model

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

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")

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

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
[12]: KNeighborsClassifier
KNeighborsClassifier(n_neighbors=3)
```

1.6. Prediksi dengan KNN Classification Model

```
[13]: tinggi_badan = 155
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")
berat_badan = 70
X_new = np.array([tinggi_badan, berat_badan]).reshape(1, -1)
X_new
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
[13]: array([[155,  70]])
```

```
[15]: y_new = model.predict(X_new)
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034")
y_new
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
[15]: array([1])
```

```
lb.inverse_transform(y_new)
```

```
array(['wanita'], dtype='<U6')
```

1.7. Visualisasi Nearest Neighbours

```
]:
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

fig, ax = plt.subplots()
for jk, d in sensus_df.groupby('jk'):
    ax.scatter(d['tinggi'], d['berat'], label=jk)


plt.scatter(tinggi_badan,
            berat_badan,
            marker='s',
            color='red',
            label='misterius')

plt.legend(loc='upper left')
plt.title('Sebaran Data Tinggi Badan, Berat Badan, dan Jenis Kelamin')
plt.xlabel('Tinggi Badan (cm)')
plt.ylabel('Berat Badan (kg)')
plt.grid(True)
plt.show()
```

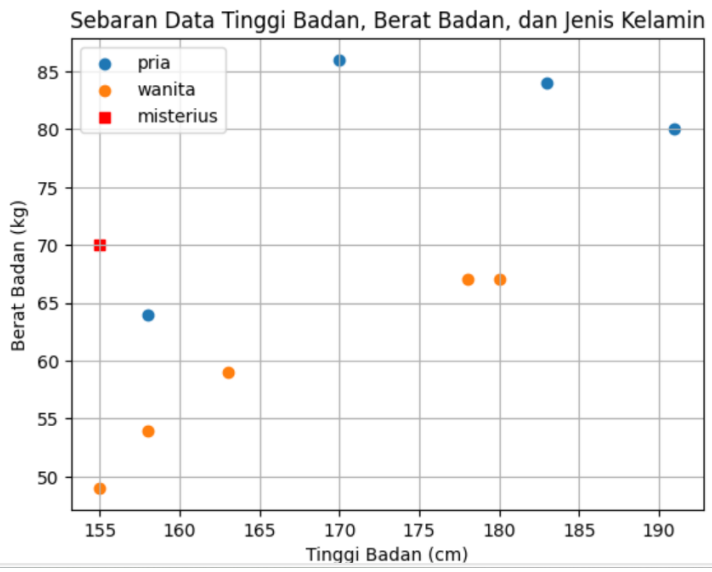
```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

jupyter tugas5 Last Checkpoint: 4 hours ago

File Edit View Run Kernel Settings Help

Code Jup

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```




1.8. Kalkulasi jarak dengan Euclidean Distance

31]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")
misterius = np.array([tinggi_badan, berat_badan])
misterius
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

31]:
```
array([155,  70])
```

[ ]:

]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

X_train
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

]:
```
array([[158,  64],
       [170,  86],
       [183,  84],
       [191,  80],
       [155,  49],
       [163,  59],
       [180,  67],
       [158,  54],
       [178,  67]], dtype=int64)
```

TugasPertemuan5[411550502100 | tugas5 | Gemini

localhost:8888/notebooks/tugas5.ipynb

jupyter tugas5 Last Checkpoint: 4 hours ago

File Edit View Run Kernel Settings Help

Code

```
       [158,  54],
       [178,  67]], dtype=int64)
```

[34]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

from scipy.spatial.distance import euclidean

data_jarak = [euclidean(misterius, d) for d in X_train]
data_jarak
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[34]:
```
[6.708203932499369,
 21.93171219946131,
 31.304951684997057,
 37.36308338453881,
 21.0,
 13.601470508735444,
 25.179356624028344,
 16.278820596099706,
 23.194827009486403]
```

localhost:8888/notebooks/tugas5.ipynb

jupyter tugas5 Last Checkpoint: 5 hours ago

File Edit View Run Kernel Settings Help

Code

```
23.194827009486403]
```

[36]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

sensus_df['jarak'] = data_jarak
sensus_df.sort_values(['jarak'])
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[36]:

| | tinggi | berat | jk | jarak |
|---|---|---|---|---|
| **0** | 158 | 64 | pria | 6.708204 |
| **5** | 163 | 59 | wanita | 13.601471 |
| **7** | 158 | 54 | wanita | 16.278821 |
| **4** | 155 | 49 | wanita | 21.000000 |
| **1** | 170 | 86 | pria | 21.931712 |
| **8** | 178 | 67 | wanita | 23.194827 |
| **6** | 180 | 67 | wanita | 25.179357 |
| **2** | 183 | 84 | pria | 31.304952 |
| **3** | 191 | 80 | pria | 37.363083 |

1.9. Evaluasi KNN Classification Model | Persiapan testing set

localhost:8888/notebooks/tugas5.ipynb

jupyter tugas5 Last Checkpoint: 5 hours ago

File Edit View Run Kernel Settings Help

Code

[40]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

X_test = np.array([[168, 65], [180, 96], [160, 52], [169, 67]])
y_test = lb.transform(np.array(['pria', 'pria', 'wanita', 'wanita'])).flatten()

print(f'x_test: \n{X_test}\n')
print(f'y_test:\n{y_test}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

x_test:
[[168  65]
 [180  96]
 [160  52]
 [169  67]]

y_test:
[0 0 1 1]
```

[ ]:

Prediksi terhadap testing

```
y_test:
[0 0 1 1]
```

[46]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

y_pred = model.predict(X_test)
y_pred
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[46]:
```
array([1, 0, 1, 1])
```

[ ]:

1.9. Evaluasi model dengan accuracy score

**Accuracy**

Accuracy is the proportion of test instances that were classified correctly.

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

```
[47]: print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")
      from sklearn.metrics import accuracy_score

      acc = accuracy_score(y_test, y_pred)

      print(f'Accuracy: {acc}')

      Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

      Accuracy: 0.75

[ ]:
```

1.10. Evaluasi model dengan precision score

**Precission**

Precision is the proportion of test instances that were predicted to be positive that are truly positive.

$$precission = \frac{tp}{tp + fp}$$

```
      Accuracy: 0.75

[48]: print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

      from sklearn.metrics import precision_score

      prec = precision_score(y_test, y_pred)

      print(f'Precision: {prec}')

      Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

      Precision: 0.6666666666666666
```

1.11. Evaluasi model dengan recall score

**Recall**

Recall is the proportion of truly positive test instances that were predicted to be positive.

$$recall = \frac{tp}{tp + fn}$$

[49]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

from sklearn.metrics import recall_score

rec = recall_score(y_test, y_pred)

print(f'Recall: {rec}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

Recall: 1.0
```

1.12. Evaluasi model dengan F1 score

**F1 Score**

The F1 score is the harmonic mean of precision and recall.

$$F1 = 2 \times \frac{precission \times recall}{precission + recall}$$

```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

from sklearn.metrics import f1_score

f1 = f1_score(y_test, y_pred)

print(f'F1-score: {f1}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

F1-score: 0.8
```

1.13. Evaluasi model dengan classification report

```
[51]: print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

from sklearn.metrics import classification_report

cls_report = classification_report(y_test, y_pred)

print(f'Classification Report:\n{cls_report}')
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       0.67      1.00      0.80         2

    accuracy                           0.75         4
   macro avg       0.83      0.75      0.73         4
weighted avg       0.83      0.75      0.73         4
```

1.14. Evaluasi model dengan Mathews Correlation Coefficient

**Matthews Correlation Coefficient (MCC)**

- MCC is an alternative to the F1 score for measuring the performance of binary classifiers.
- A perfect classifier's MCC is 1.
- A trivial classifier that predicts randomly will score 0, and a perfectly wrong classifier will score -1.

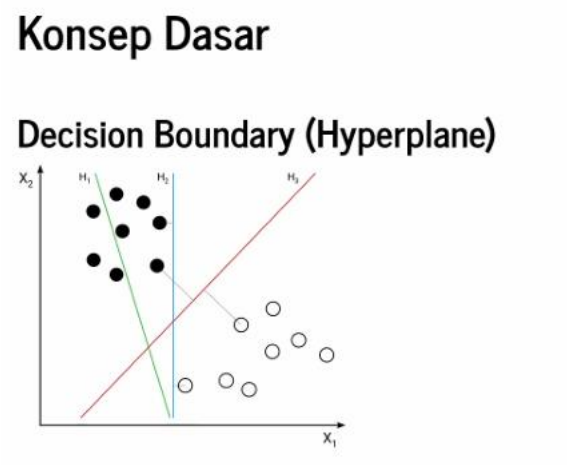
$$MCC = \frac{tp \times tn + fp \times fn}{\sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}}$$

```
[52]: print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

from sklearn.metrics import matthews_corrcoef

mcc = matthews_corrcoef(y_test, y_pred)

print(f'MCC: {mcc}')
```

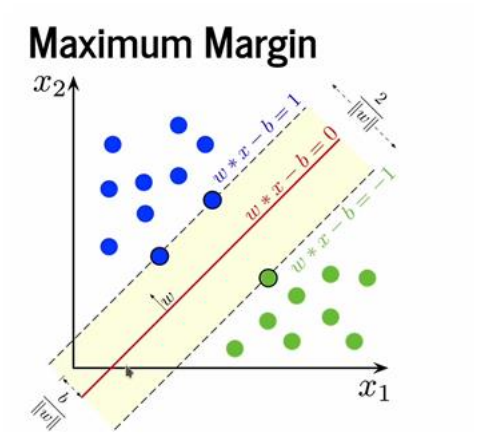
```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

MCC: 0.5773502691896258
```

2.0. Support Vector Machine (SVM). Lakukan praktik dari https://youtu.be/z69XYXpvVrE?si=KR_hDSlwjGIMcT0w , buat screenshot dengan nama kalian pada coding, kumpulkan dalam bentuk pdf, dari kegiatan ini:

2.1. Pengenalan Decision Boundary & Hyperplane



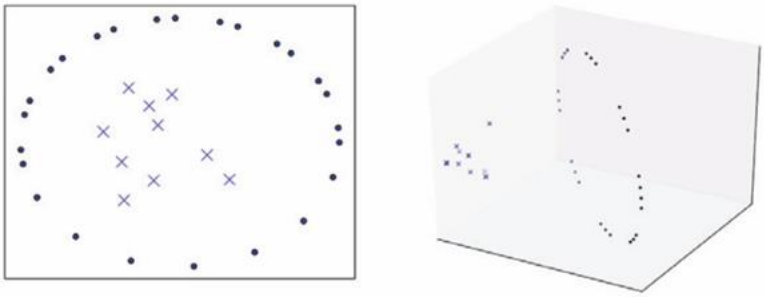

2.2. Pengenalan Support Vector & Maximum Margin




2.3. Pengenalan kondisi Linearly Inseparable dan Kernel Tricks

## Linearly Inseperable & Kernel Tricks

Referensi: https://www.quora.com/What-is-the-kernel-trick



2.4. Pengenalan MNIST Handwritten Digits Dataset

### Classification Task dengan Support Vector Machine (SVM)

Referensi: https://www.svm-tutorial.com/

**Dataset: The MNIST database of handwritten digits**

[*]:

```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")
from sklearn.datasets import fetch_openml
import pandas as pd


X, y = fetch_openml('mnist_784', data_home="./dataset/mnist", return_X_y=True, parser='auto')

# Menampilkan bentuk data
print(X.shape)
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[55]:

```
(70000, 784)
```

jupyter tugas5 Last Checkpoint: 5 hours ago

File Edit View Run Kernel Settings Help

Code

```
(70000, 784)
```

[59]:
```
import matplotlib.pyplot as plt
import matplotlib.cm as cm

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

pos = 1
for data in X.to_numpy()[:8]:
    plt.subplot(1, 8, pos)
    plt.imshow(data.reshape((28, 28)), cmap=cm.Greys_r)
    plt.axis('off')
    pos += 1

plt.show()
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[ ]:

[60]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

y[:8]
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[60]:
```
0    5
1    0
2    4
3    1
4    9
5    2
6    1
7    3
Name: class, dtype: category
Categories (10, object): ['0', '1', '2', '3', ..., '6', '7', '8', '9']
```

```
X_train = X[:60000]
y_train = y[:60000]

X_test = X[60000:]
y_test = y[60000:]
```

2.5. Klasifikasi dengan Support Vector Classifier | SVC

[26]:
```
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")
from sklearn.svm import SVC

model = SVC(random_state=0)
model.fit(X_train, y_train)
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034
```

[26]:
```
SVC
SVC(random_state=0)
```

[28]:
```
from sklearn.metrics import classification_report
print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       0.67      1.00      0.80         2

    accuracy                           0.75         4
   macro avg       0.83      0.75      0.73         4
weighted avg       0.83      0.75      0.73         4
```

2.6. Hyperparameter Tuning dengan Grid Search

[30]:
```python
from sklearn.model_selection import GridSearchCV

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

parameters = {
    'kernel': ['rbf', 'poly', 'sigmoid'],
    'C': [0.5, 1, 10, 100],
    'gamma': ['scale', 1, 0.1, 0.01, 0.001]
}

grid_search = GridSearchCV(estimator=SVC(random_state=0),
                           param_grid=parameters,
                           n_jobs=6,
                           verbose=1,
                           scoring='accuracy')

grid_search.fit(X_train, y_train)
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

Fitting 5 folds for each of 60 candidates, totalling 300 fits
c:\users\asep dimyati\appdata\local\programs\python\python38\lib\site-packages\sklearn\mo
ass in y has only 4 members, which is less than n_splits=5.
  warnings.warn(
```

[30]:

▸ GridSearchCV

▸ estimator: SVC

▸ SVC

[31]:
```python
print(f'Best Score: {grid_search.best_score_}')

print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")
best_params = grid_search.best_estimator_.get_params()
print(f'Best Parameters:')

for param in parameters:
    print(f'\t{param}: {best_params[param]}')
```

```
Best Score: 0.9
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

Best Parameters:
        kernel: rbf
        C: 1
        gamma: 0.01
```

2.7. Evaluasi Model

```
gamma: 0.01
```

```
[32]: print("Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034\n")

y_pred = grid_search.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
Nama: Triansyah Amarullah Ahmad Prayoga, NPM: 41155058210034

              precision    recall  f1-score   support

           0       1.00      0.50      0.67         2
           1       0.67      1.00      0.80         2

    accuracy                           0.75         4
   macro avg       0.83      0.75      0.73         4
weighted avg       0.83      0.75      0.73         4
```