Modul

Pengenalan Pola

Classification With Linear SVM Python

I. TUJUAN

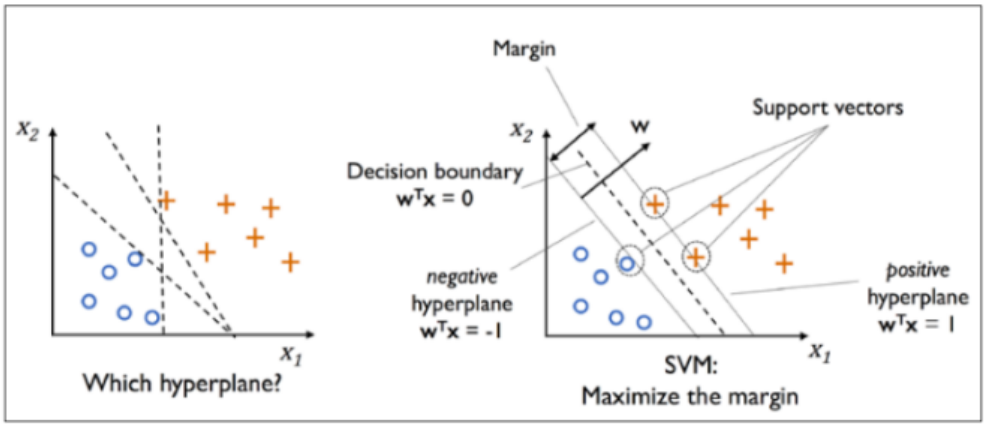
1. Mahasiswa mengetahui penggunaan Linear SVM pada Pengenalan Pola
2. Mahasiswa mengetahui tahapan dalam metode Linear SVM.
3. Mahasiswa mampu melakukan pemrograman dasar Linear SVM

II. ALAT DAN BAHAN

1. Laptop/PC
2. Google Colab

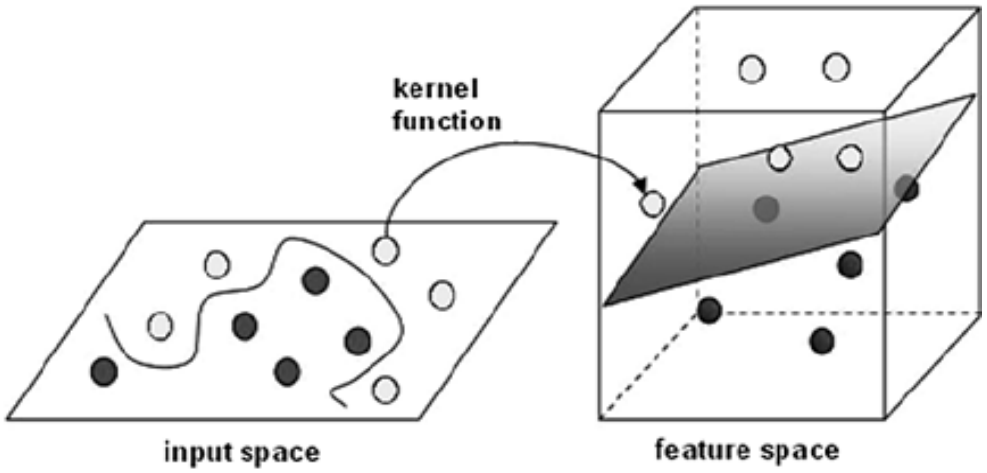
III. TEORI DASAR

Support Vector Machine (SVM) merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan *linear* maupun *non linear*.



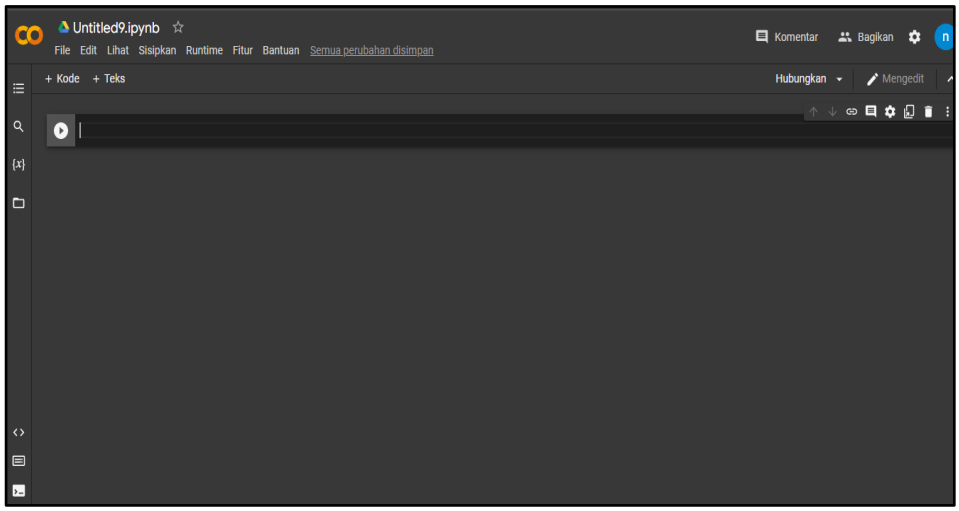
Gambar 1 Hyperplane yang memisahkan dua kelas positif (+1) dan negatif(-1)

SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. H*yperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut *plane similarly*, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut *hyperplane*. *Hyperplane* yang ditemukan SVM diilustrasikan seperti Gambar 1 posisinya berada ditengah-tengah antara dua kelas, artinya jarak antara hyperplane dengan objek-objek data berbeda dengan kelas yang berdekatan (terluar) yang diberi tanda bulat kosong dan positif. Dalam SVM objek data terluar yang paling dekat dengan *hyperplane* disebut *support vector*. Objek yang disebut *support vector* paling sulit diklasifikasikan dikarenakan posisi yang hampir tumpang tindih (*overlap*) dengan kelas lain. Mengingat sifatnya yang kritis, hanya *support vector* inilah yang diperhitungkan untuk menemukan *hyperplane* yang paling optimal oleh SVM.

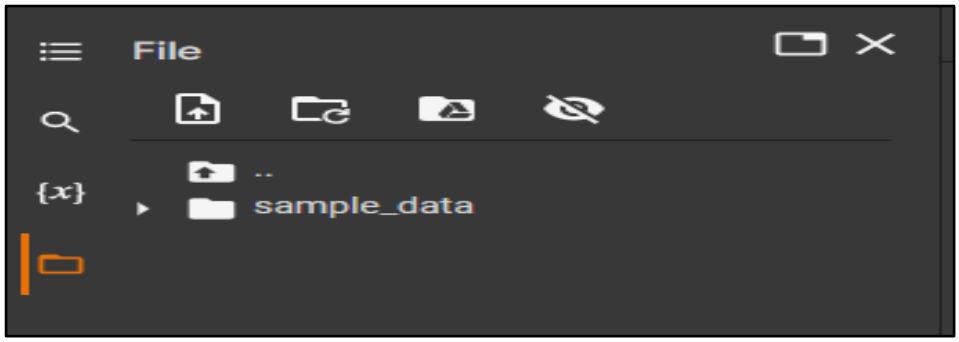


IV. LANGKAH KERJA

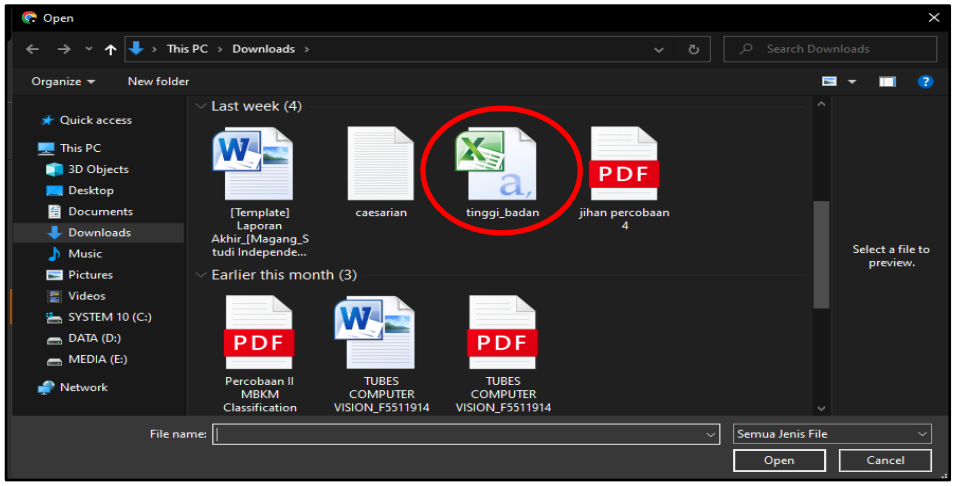
1. Bukalah Google Colab



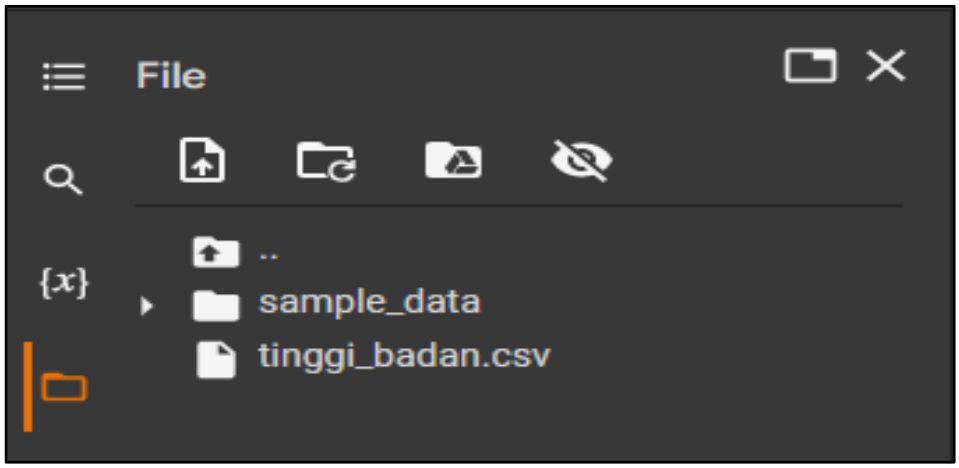
2. Pilihlah icon file pada sebelah kiri kemudian klik upload



3. Pilih lokasi dataset dan upload



4. Setelah terupload dataset akan muncul pada bagian jendela kiri



5. Masukan kode program untuk meginput library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# use seaborn plotting defaults
import seaborn as sns; sns.set()
```

6. Masukan kode program untuk menambahkan dataset ke dalam program dan menampilkan data pertama

```
[2] data = pd.read_csv('tinggi_badan.csv')

data.head()
```

7. Masukan kode program untuk membagi dataset menjadi 2 variabel yaitu x dan y

```
X = data.drop('BB', axis=1)
y = data['BB']
```

8. Masukan kode program untuk membagi dataset menjadi data training dan data test

```
[5] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

9. Masukan kode program untuk membuat model SVM terhadap training

```
[6] from sklearn.svm import SVC
    svclassifier = SVC(kernel='linear')
    svclassifier.fit(X_train, y_train)
```

10. Masukan kode program untuk memprediksi hasil test set pada variable x

```
[7] y_pred = svclassifier.predict(X_test)
```

11. Masukan kode program untuk membuat Confusion matrix

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

12. Masukan kode program untuk membuat cluster

```
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=50, centers=2,
                  random_state=0, cluster_std=0.60)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn');
```

```
[10] xfit = np.linspace(-1, 3.5)
     plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='autumn')
     plt.plot([0.6], [2.1], 'x', color='red', markeredgewidth=2, markersize=10)

     for m, b in [(1, 0.65), (0.5, 1.6), (-0.2, 2.9)]:
         plt.plot(xfit, m * xfit + b, '-k')

     plt.xlim(-1, 3.5);
```

V. HASIL PERCOBAAN

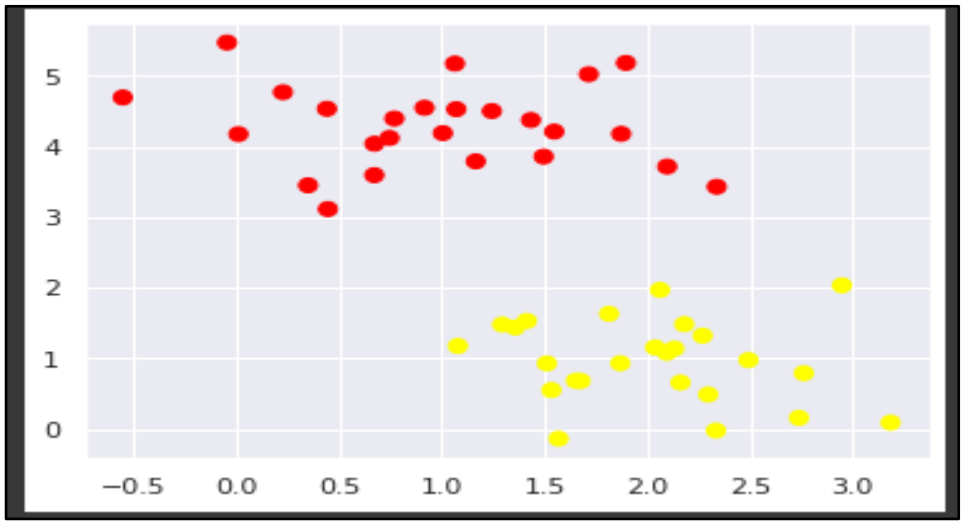
1. Tampilan 5 data awal dari dataset tinggi badan

	Tinggi_Badan	BB
0	183.78	70
1	183.30	80
2	182.79	75
3	182.47	71
4	182.10	85

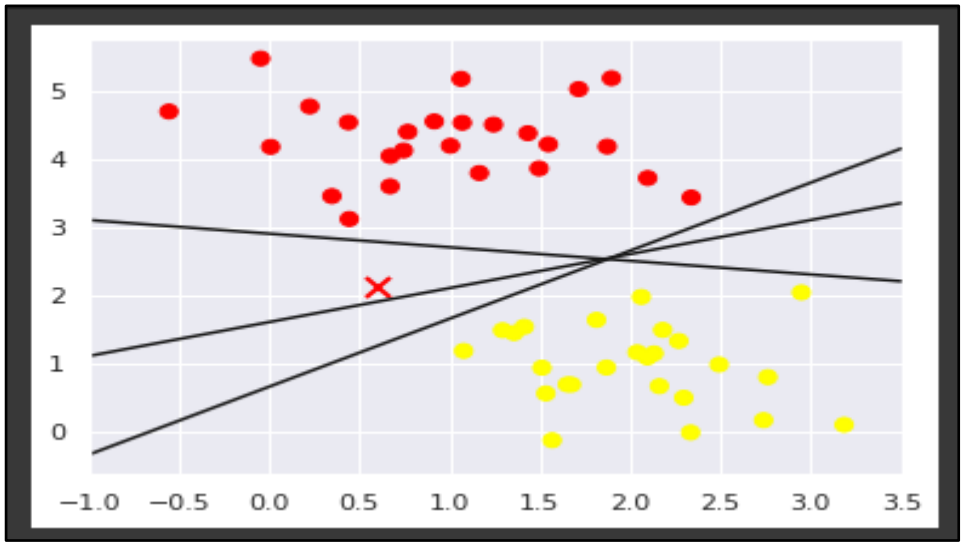
2. Tampilan dari confusion matriks data test dan data training

```
[[0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 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 0 0]
 [1 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 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 0]]
              precision    recall  f1-score   support

          50       0.00      0.00      0.00       0.0
          54       0.00      0.00      0.00       1.0
          61       0.00      0.00      0.00       0.0
          64       0.00      0.00      0.00       1.0
          67       0.00      0.00      0.00       2.0
          69       0.00      0.00      0.00       1.0
          70       0.00      0.00      0.00       1.0
          74       0.00      0.00      0.00       1.0
          77       0.00      0.00      0.00       1.0
          79       0.00      0.00      0.00       0.0
          80       0.00      0.00      0.00       0.0
          81       0.00      0.00      0.00       1.0

    accuracy                           0.00       9.0
   macro avg       0.00      0.00      0.00       9.0
weighted avg       0.00      0.00      0.00       9.0
```

3. Tampilan visualisasi hasil model SVM terhadap Training set



4. Tampilan visualisasi hasil model SVM terhadap Test set



VI. ANALISIS

Pada percobaan ini dilakukan prediksi data Tinggi Badan(cm) dan Data Berat Badan (Kg) disini tinggi badan merupakan variabel independen (x), dan berat badan merupakan variabel dependen (y). Ada beberapa langkah yang harus dilakukan yaitu pertama adalah mengimpor library yang diperlukan seperti import numpy as np & import pandas as pd, import matplotlib.pyplot as plt berfungsi untuk bisa dengan mudah melakukan visualisasi data seperti

membuat gambar, membuat area plot dalam gambar, menambah label di plot dan lainnya. *import seaborn as sns; sns.set()* untuk melakukan pembetukan grafik pada klsifikasi data, *data = pd.read_csv('tinggi_badan.csv')* untuk membaca dataset yang akan dikalsifikasi dalam format csv yaitu data tinggi badan, *data.head()* berfungsi untuk menampilkan lima data awal dalam dataset. *train_test_split(X, y, test_size = 0.20)* untuk membagi data menjadi training dan test, *svclassifier = SVC(kernel='linear')* untuk melakukan pemodelan klasifikasi sederhana menggunakan svc, *y_pred = svclassifier.predict(X_test)* untuk melakukan prediksi dari data test variablex,*from sklearn.metrics import classification_report, confusion_matrix* untuk membuat confusion matrik data data test dan training variable y. *from sklearn.datasets import make_blobs* untuk membandingkan hasil cluster dari data training dan data test.

VII. ANALISIS

Kernel SVM biasanya digunakan ketika dataset berupa data yang tidak linier, atau data memuat kelas-kelas yang overlap atau tercampur. Ini merupakan salah satu kelebihan SVM, di mana algoritma ini dapat digunakan untuk proses klasifikasi maupun regresi, pada data linier dan non-linier.

.