Klasifikasi Naive Bayes Pada Data Iris Menggunakan Python

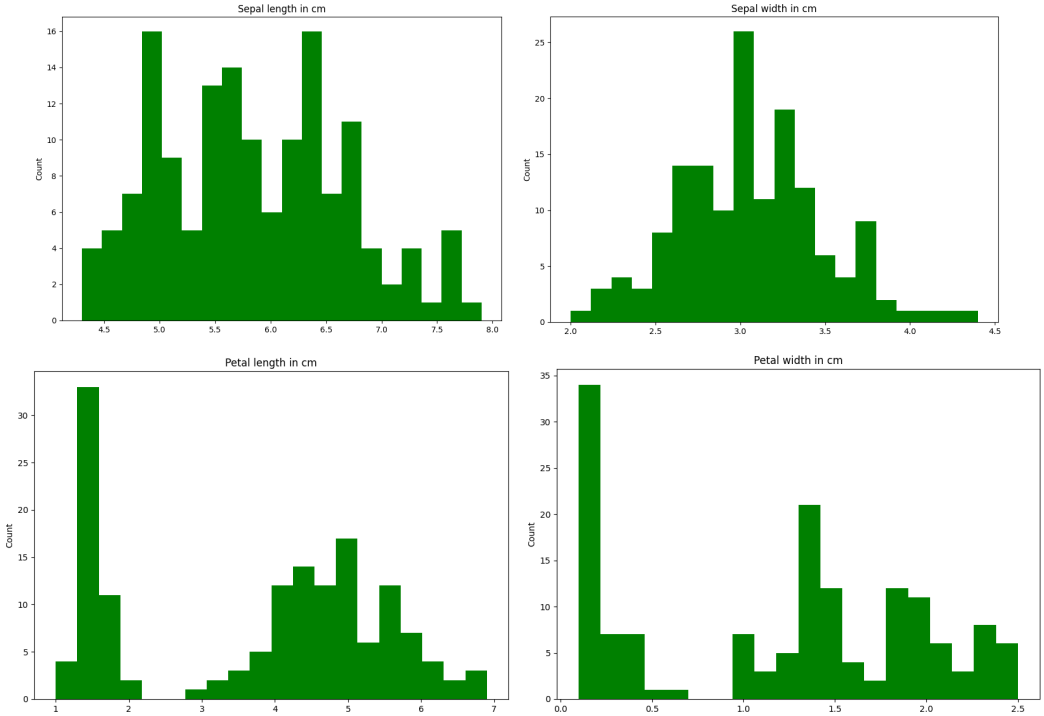
Wildan Fajri Alfarabi G64190060
(github : https://github.com/wildanfajri1alfarabi/pengenalan-pola/tree/main/week3-naive-bayes)

Dataset dan Visualisasi Karakteristik Dataset

Dataset yang digunakan adalah dataset Iris, yang merupakan dataset bunga iris dengan 4 fitur dan 3 kelas bunga iris. Fitur yang terdapat pada dataset adalah fitur panjang dan lebar dari masing-masing penciri bunga iris, yaitu sepal dan petal dari bunga. Kelas pada dataset ada 3, yang merupakan jenis bunga iris {Setosa, Versicolor dan Virginica}.

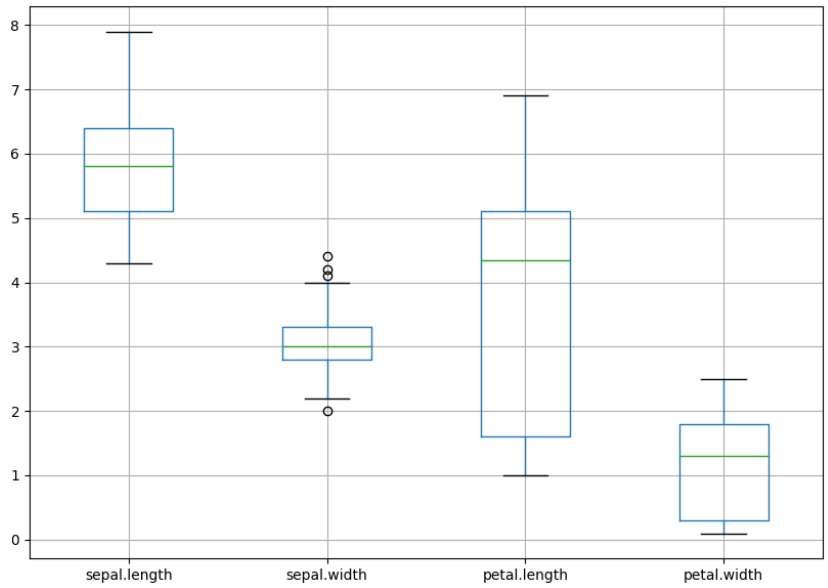
Sebaran data Iris dapat dilihat menggunakan library Pandas menggunakan fungsi describe(). Total baris atau record data iris sebanyak 150. Fitur sepal length memiliki rata-rata 5.83cm dengan maksimum dan minimumnya adalah 7.9cm dan 4.3cm. Fitur sepal width memiliki rata-rata 3.95733cm dengan maksimum dan minimumnya adalah 4.4cm dan 2cm. Statistika ringkasan dan sebaran data untuk fitur lainnya dapat dilihat pada gambar dan histogram dibawah. Sebaran data ini dapat digunakan untuk klasifikasi jenis bunga iris, pada lkp ini akan menggunakan model Naive Bayes menggunakan package scikit-learn.

	sepal.length	sepal.width	petal.length	petal.width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333
std	0.828066	0.435866	1.765298	0.762238
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

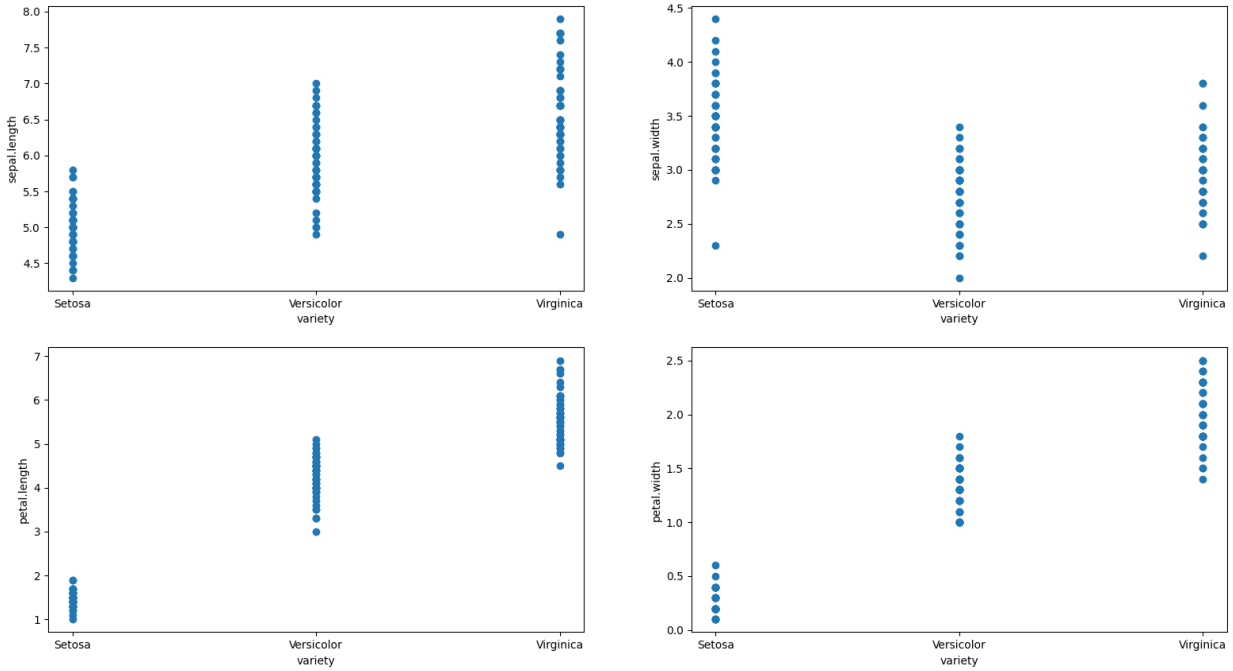


Berikut adalah visualisasi sebaran dataset iris yang lainnya, yaitu adalah boxplot dan juga scatterplot. Boxplot pada dataset Iris dapat digunakan untuk menentukan data *outlier* dimana terdapat pada fitur sepal

width yang ditandai dengan lingkaran diluar garis boxplot. Boxplot tersebut juga dapat menunjukan media (garis hijau) dan juga Kuartil 1 dan 3 (Kotak biru). Fitur yang memiliki nilai sebaran yang sempit adalah sepal width dan fitur dengan sebaran data paling lebar adalah petal length.



Visualisasi selanjutnya adalah boxplot, dapat dilihat bahwa masing-masing fitur yang menjadi penciri dari kelas jenis iris tertentu memiliki perbedaan rentang nilai. Pada fitur petal, semakin besar lebar dan panjangnya, maka akan semakin berbeda kelas jenis irisnya.



Klasifikasi Naive Bayes Untuk Dataset Iris

Tahap pertama adalah pemisahan kelas dataset dan juga atribut/fitur dari dataset. Setelah data dipisahkan, selanjutnya adalah pemisahan dataset latih dan dataset uji model prediksi. Pemisahan dataset latih dan uji yang saya terapkan adalah 75% data latih dan 25% data uji. Berikut adalah codenya :

```
x_train, x_test, y_train, y_test = train_test_split(
        iris_data_noclass, iris_data_class,test_size=0.25, random_state=5)
```

Setelah menentukan proporsi dan membagi data latih dan uji, selanjutnya adalah memanggil fungsi naive bayes gaussian dari package scikit-learn. Gaussian Naive Bayes adalah *classifier* Naive Bayes sederhana yang memiliki asumsi data dari masing-masing label diambil dari distribusi Gaussian sederhana. Input data latih (penciri dan kelas) digunakan untuk melatih model prediksi dengan cara input data latih ke dalam model dengan fungsi fit().

```
# Pakai fungsi naive bayes gaussian dari pkg scikit
    iris_naivebayes_model = GaussianNB()

# Input data latih (fitur dan kelas) untuk latih model dengan fungsi fit
    naivebayes_train = iris_naivebayes_model.fit(x_train, y_train)
```

Setelah dilakukan pelatihan model, selanjutnya model dapat digunakan untuk memprediksi data uji.

```
iris_data_class_pred = naivebayes_train.predict(x_test)
```

Ide utama dari model prediksi naive bayes adalah berdasarkan nilai probabilitas dari dataset. Probabilitas kemunculan kelas, probabilitas *neighbourhood* (Kemunculan rentang nilai fitur penciri terhadap kelas jenis) dan probabilitas lainnya yang akan digunakan untuk memprediksi kelas data yang baru pertama kali model lihat. Berikut ini adalah hasil prediksi dari model yang telah dilatih diatas.

	Prediksi	KelasIris
82	Versicolor	Versicolor
134	Versicolor	Virginica
114	Virginica	Virginica
42	Setosa	Setosa
109	Virginica	Virginica
57	Versicolor	Versicolor
1	Setosa	Setosa
70	Virginica	Versicolor
25	Setosa	Setosa
84	Versicolor	Versicolor
66	Versicolor	Versicolor
133	Versicolor	Virginica
102	Virginica	Virginica
107	Virginica	Virginica
26	Setosa	Setosa
23	Setosa	Setosa
123	Virginica	Virginica
130	Virginica	Virginica
21	Setosa	Setosa
12	Setosa	Setosa

71	Versicolor	Versicolor
128	Virginica	Virginica
48	Setosa	Setosa
72	Versicolor	Versicolor
88	Versicolor	Versicolor
148	Virginica	Virginica
74	Versicolor	Versicolor
96	Versicolor	Versicolor
63	Versicolor	Versicolor
132	Virginica	Virginica
39	Setosa	Setosa
53	Versicolor	Versicolor
79	Versicolor	Versicolor
10	Setosa	Setosa
50	Versicolor	Versicolor
49	Setosa	Setosa
43	Setosa	Setosa
135	Virginica	Virginica

Pada hasil prediksi model, mayoritas data uji tepat diprediksi oleh model. Tetapi ada beberapa record data yang salah diklasifikasi seperti pada nomor 134 jenis Virginica salah diprediksi menjadi Versicolor. Hal ini dikarenakan model memprediksi kelas berdasarkan probabilitas data kelas yang diprediksi. Berikut adalah sebaran probabilitas prediksi dari data uji.

```
probabilitas prediksi
 [[4.98983163e-068 9.99976886e-001 2.31140535e-005]
 [2.38605399e-151 6.05788555e-001 3.94211445e-001]
 [1.52930285e-229 6.40965644e-007 9.99999359e-001]
 [1.00000000e+000 3.59116140e-019 4.35301434e-027]
 [2.95380506e-300 1.34947916e-012 1.00000000e+000]
 [2.16647395e-038 9.99999791e-001 2.08857492e-007]
 [1.00000000e+000 1.05055556e-017 6.59080633e-026]
 [9.55629438e-147 1.08807974e-001 8.91192026e-001]
 [1.00000000e+000 2.52482409e-016 1.46364748e-024]
 [6.02158895e-108 9.90558853e-001 9.44114652e-003]
 [2.10951969e-108 9.88848490e-001 1.11515105e-002]
 [1.45542701e-135 7.56061808e-001 2.43938192e-001]
 [8.96514127e-239 1.55965373e-007 9.99999844e-001]
 [5.55336438e-231 1.60582831e-006 9.99998394e-001]
 [1.00000000e+000 3.60824506e-014 6.21633393e-022]
 [1.00000000e+000 1.29808311e-010 2.24314344e-018]
 [3.09565382e-153 1.07603601e-001 8.92396399e-001]
 [3.81317125e-230 1.03976329e-006 9.99998960e-001]
 [1.00000000e+000 1.37507763e-015 5.26100959e-023]
```

Pada data array index kesatu, dua dan tiga, menunjukan probabilitas prediksi untuk kelas Setosa, Versicolor dan Virginica. Model memprediksi untuk data pertama memiliki probabilitas sebesar 0.9997 sebagai kelas Versicolor. Untuk data kedua (nomor id 134) probabilitas prediksi yang besar adalah kelas Versicolor (0.6057) dibandingkan dengan kelas Virginica (kelas asli dari record data) yang hanya memiliki probabilitas sebesar (0.3942). Oleh karena itu, model memilih untuk mengklasifikasikan data record 134 sebagai versicolor bukannya virginica.

Berikut adalah metrik seperti confusion matrix dan juga report terkait model prediksi yang dihasilkan. Pada confusion matrix, kolom matriks adalah kelas prediksi dan baris matriks adalah kelas asli dari data Iris. Jumlah NxN matriks sesuai dengan jumlah kelas pada dataset, yaitu {setosa, versicolor dan virginica}. Pada hasil model, kelas setosa 100% tepat diprediksi, sedangkan kelas versicolor salah diprediksi sebagai virginica sebanyak 1 kali, sebaliknya kelas virginica salah diprediksi sebanyak 2 buah sebagai kelas versicolor oleh model. Akurasi rata-rata dari semua kelas yang tepat diprediksi oleh model adalah 92%.

```
Confussion Matrix
 [[12  0  0]
 [ 0 13  1]
 [ 0  2 10]]

Report Klasifikasi
              precision    recall  f1-score   support

      Setosa       1.00      1.00      1.00        12
  Versicolor       0.87      0.93      0.90        14
   Virginica       0.91      0.83      0.87        12

    accuracy                           0.92        38
   macro avg       0.93      0.92      0.92        38
weighted avg       0.92      0.92      0.92        38
```

Code Program

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, classification_report

def show_histogram(data, title="Histogram", bins=20, hist_color="green"):
    plt.figure(figsize=(10, 7))
    plt.hist(data,bins,color=hist_color)
    plt.title(title)
    plt.ylabel("Count")

    plt.show()

def show_boxplot(iris_data_noclass):
    plt.figure(figsize=(10, 7))
    iris_data_noclass.boxplot()
    plt.show()

def show_scatter_plot(data,data_class):
    var_cnt = len(data.axes[1])-1
    fig, ax = plt.subplots(2, var_cnt // 2)
    for idx, var_title in enumerate(data.axes[1]):
        if idx <= var_cnt-1:
            ax[idx//2, idx%2].scatter(data[data_class], data[var_title])
            ax[idx//2, idx%2].set(xlabel=data_class, ylabel=var_title)
    plt.show()

def naive_bayes_gaussian(iris_data_noclass, iris_data_class):
    # Split data uji dan data latih dengan proporsi 1:3
    x_train, x_test, y_train, y_test = train_test_split(
        iris_data_noclass, iris_data_class,test_size=0.25, random_state=5)

    # Pakai fungsi naive bayes gausian dari pkg scikit
    iris_naivebayes_model = GaussianNB()

    # Input data latih (fitur dan kelas) untuk latih model dengan fungsi fit
    naivebayes_train = iris_naivebayes_model.fit(x_train, y_train)

    # Gunakan model naive bayes gausian yg telah dilatih utk prediksi kelas data uji
    iris_data_class_pred = naivebayes_train.predict(x_test)

    # Measurement
```

```
    df_hasil_prediksi = pd.DataFrame({"Prediksi": iris_data_class_pred, "KelasIris":
y_test})
    print(df_hasil_prediksi, "\n")
    print("probabilitas prediksi", "\n", naivebayes_train.predict_proba(x_test),
"\n")
    print("Confussion Matrix", "\n", confusion_matrix(y_test, iris_data_class_pred),
"\n")
    print("Report Klasifikasi", "\n", classification_report(y_test,
iris_data_class_pred), "\n")

if __name__ == "__main__":
    iris_data = pd.read_csv("iris.csv")
    print(iris_data.head(15),"\n")
    print(iris_data.describe())

    #Persiapan dataset iris
    iris_data_class = iris_data["variety"]
    print(iris_data_class.head(), "\n")
    iris_data_noclass = iris_data.drop(columns=["variety"])
    print(iris_data_noclass.head(), "\n")

    #Tampilkan visualisasi sebaran data
    show_histogram(data=iris_data["sepal.length"],title="Sepal length in cm")
    show_histogram(data=iris_data["sepal.width"], title="Sepal width in cm")
    show_histogram(data=iris_data["petal.length"], title="Petal length in cm")
    show_histogram(data=iris_data["petal.width"], title="Petal width in cm")
    show_boxplot(iris_data_noclass)
    show_scatter_plot(data=iris_data,data_class="variety")

    #Naive Bayes
    naive_bayes_gaussian(iris_data_noclass, iris_data_class)
```