

Laporan Tugas Kecil
IF2211 Strategi Algoritma



Dibuat oleh:

13520026 Muhammad Fajar Ramadhan

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132

DAFTAR ISI

Algoritma <i>Divide and Conquer</i>	3
Source Kode.....	3
Algoritma Convex Hull	3
Visualisasi	5
Referensi	11
Link Drive	11

Algoritma *Divide and Conquer*

Divide and Conquer merupakan salah satu algoritma dalam menyelesaikan berbagai macam permasalahan komputasional. Algoritma ini menggunakan konsep untuk memecah masalah menjadi sub-masalah hingga cukup sederhana untuk diselesaikan. Terdapat tiga tahap dalam penggunaan algoritma *Divide and Conquer*.

- **Divide**: membagi persoalan menjadi beberapa upa-persoalan yang memiliki kemiripan dengan persoalan semula namun berukuran lebih kecil (idealnya berukuran hampir sama)
- **Conquer (solve)**: menyelesaikan masing-masing upa-persoalan (secara langsung jika sudah berukuran kecil atau secara rekursif jika masih berukuran besar).
- **Combine**: mengabungkan solusi masing-masing upa-persoalan sehingga membentuk solusi persoalan semula.

Dalam menyelesaikan permasalahan *Convex Hull* ini strategi algoritma yang diterapkan adalah sebagai berikut.

1. Urutkan daftar point berdasarkan absis nya terurut menaik, jika sama maka urutkan berdasarkan ordinatnya terurut menaik
2. Definisikan dua point extreme misal A dan B dimana A merepresentasikan point paling kiri atau absis paling kecil dan B merepresentasikan point paling kanan atau absis paling besar. Buat Garis dari point A dan B
3. Bagi daftar point menjadi 2 daerah yaitu disebelah kiri (atas) garis AB dan disebelah kanan (bawah) garis AB.
4. Cari point terjauh dari garis AB misal C, masing-masing dilakukan untuk kedua daerah kiri dan kanan
5. Hapus garis AB dan tambahkan ke solusi garis AC dan CB
6. Lakukan langkah 4 dan 5 yaitu mencari point terjauh dari garis AC dan CB (point terjauh sebelah kiri atau atas) dan membentuk garis baru
7. Lakukan sehingga tidak ada lagi point terjauh di sebelah kiri yang dapat membentuk garis baru

Source Kode

Algoritma Convex Hull

```
import numpy as np

def isEmpty(Arr):
    # Cek list kosong
    return len(Arr) == 0

def findSide(p1,p2,p3):
    # Memeriksa point p3 apakah disebelah kiri atau kanan garis yang dibentuk
    # p1 dan p2
    aX = p1[0]
    aY = p1[1]
    bX = p2[0]
    bY = p2[1]
    cX = p3[0]
```

```

cY = p3[1]

val = ((bX - aX)*(cY - aY) - (bY - aY)*(cX - aX))
thresh = 1e-9
if val >= thresh:
    return 1
elif val <= -thresh:
    return -1
else:
    return 0

def DividePoints(ArrPoints,point1,point2):
    # Membagi point menjadi 2 sebelah kiri garis dan sebelah kanan garis
    tempArr = ArrPoints
    right = []
    left = []
    for i in tempArr:
        det = findSide(point1,point2,i)
        if(det>0):
            left.append(i)
        elif(det<0):
            right.append(i)

    return left, right

def Jarak(p1,p2,p3):
    # Mengembalikan jarak dari titik p3 dengan garis yang dibentuk p1 dan p2
    a = np.array(p1)
    b = np.array(p2)
    c = np.array(p3)
    Dist = np.cross(b-a,c-a)/np.linalg.norm(b-a)
    return Dist

def findHull(ArrP,p1,p2,solution):
    # Algoritma dalam mencari ConvexHull setelah point-point sudah dibagi 2
    if(isEmpty(ArrP)):
        return
    else:
        farthestPoint = ArrP[0]
        maxDistance = 0.0

        for i in ArrP:
            tempJarak = Jarak(p1,p2,i)
            if(tempJarak >= maxDistance):
                maxDistance = tempJarak
                farthestPoint = i

        solution.remove([p1,p2])

```

```

        solution.append([p1, farthestPoint])
        solution.append([farthestPoint, p2])
        S1, temp = DividePoints(ArrP, p1, farthestPoint)
        S2, temp = DividePoints(ArrP, farthestPoint, p2)

        findHull(S1, p1, farthestPoint, solution)
        findHull(S2, farthestPoint, p2, solution)

def ConvexHull(points):
    # Algoritma dalam mencari ConvexHull dari sekumpulan point-point
    hull = []
    tempArr = points.copy()
    tempArr.sort()
    p1 = tempArr[0]
    p2 = tempArr[-1]
    hull.append([p1, p2])
    hull.append([p2, p1])
    S1, S2 = DividePoints(tempArr, p1, p2)

    findHull(S1, p1, p2, hull)
    findHull(S2, p2, p1, hull)

    for i in range(len(hull)):
        hull[i] = [points.index(hull[i][0]), points.index(hull[i][1])]

    return hull

```

Visualisasi

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from myConvexHull import myConvexHull as my

print("` ` ` | ~~~~ Program Convex Hull ~~~~ | ` ` `")
print("Pilihan data set: ")
print("1. Iris")
print("2. Wine")
print("3. Breast Cancer")
n = int(input("Masukkan pilihan sesuai nomor diatas: "))
if(n==1):
    data = datasets.load_iris()

```

```

elif(n==2):
    data = datasets.load_wine()
elif(n==3):
    data = datasets.load_breast_cancer()
#create a DataFrame
print("\nPilihan Convex Hull")
a = 0
b = 0
if(n==1):
    print("1. Petal Length vs Petal Width")
    print("2. Sepal Length vs Sepal Width")
    anw = int(input("Masukkan pilihan Convex Hull sesuai nomor: "))
    if(anw==1):
        a=2
        b=3
    else:
        a=0
        b=1
elif(n==2):
    print("1. Alcohol vs Magnesium")
    print("2. Ash vs Alcalinity of Ash")
    anw = int(input("Masukkan pilihan Convex Hull sesuai nomor: "))
    if(anw==1):
        a=0
        b=4
    else:
        a=2
        b=3
elif(n==3):
    print("1. Compactness vs Concavity")
    print("2. Texture vs Smoothness")
    anw = int(input("Masukkan pilihan Convex Hull sesuai nomor: "))
    if(anw==1):
        a=5
        b=6
    else:
        a=1
        b=4

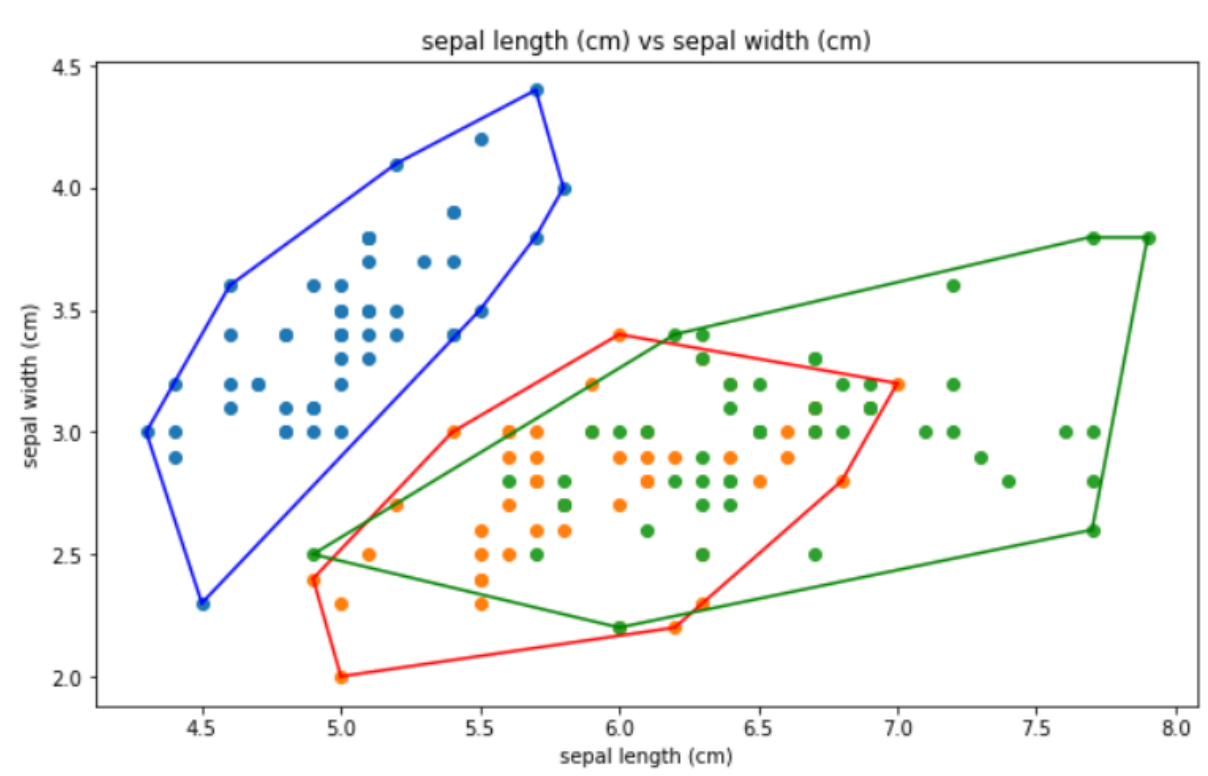
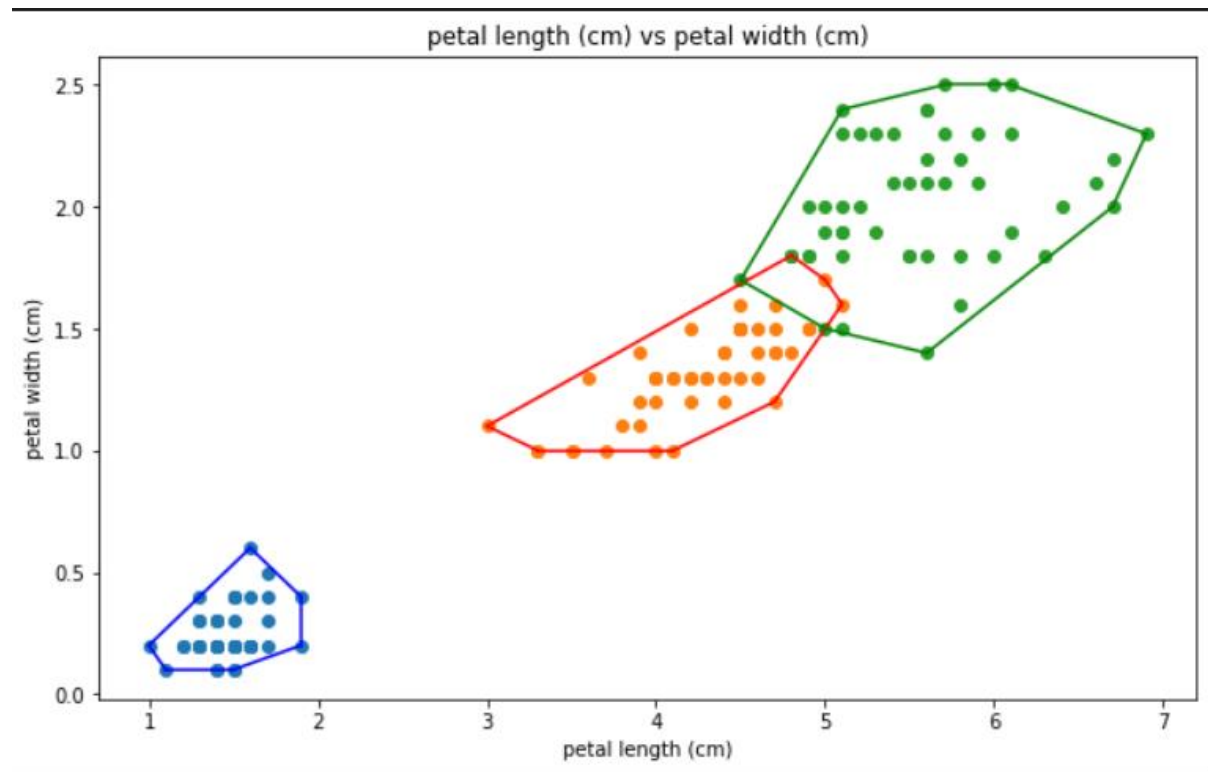
df = pd.DataFrame(data.data, columns=data.feature_names)
df['Target'] = pd.DataFrame(data.target)
plt.figure(figsize = (10, 6))
colors = ['b','r','g','y','c','m','k']
str = data.feature_names[a] + " vs " + data.feature_names[b]
plt.title(str)
plt.xlabel(data.feature_names[a])
plt.ylabel(data.feature_names[b])
for i in range(len(data.target_names)):

```

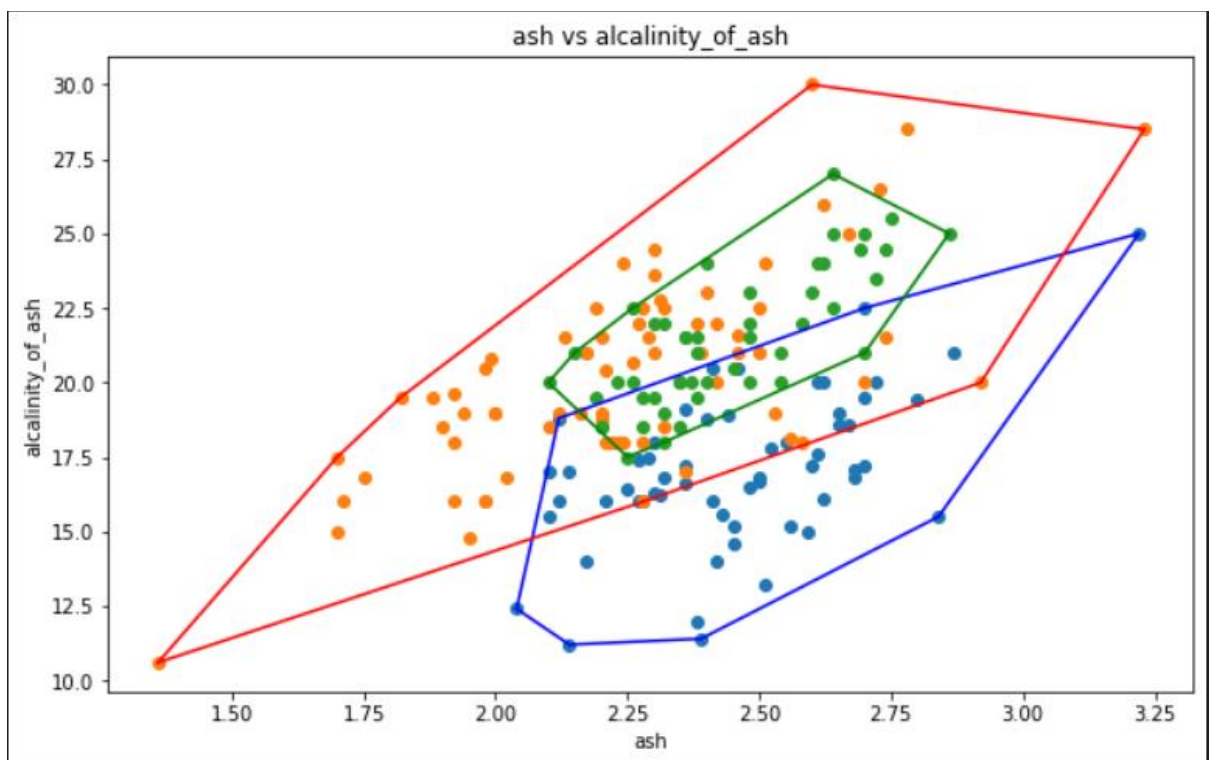
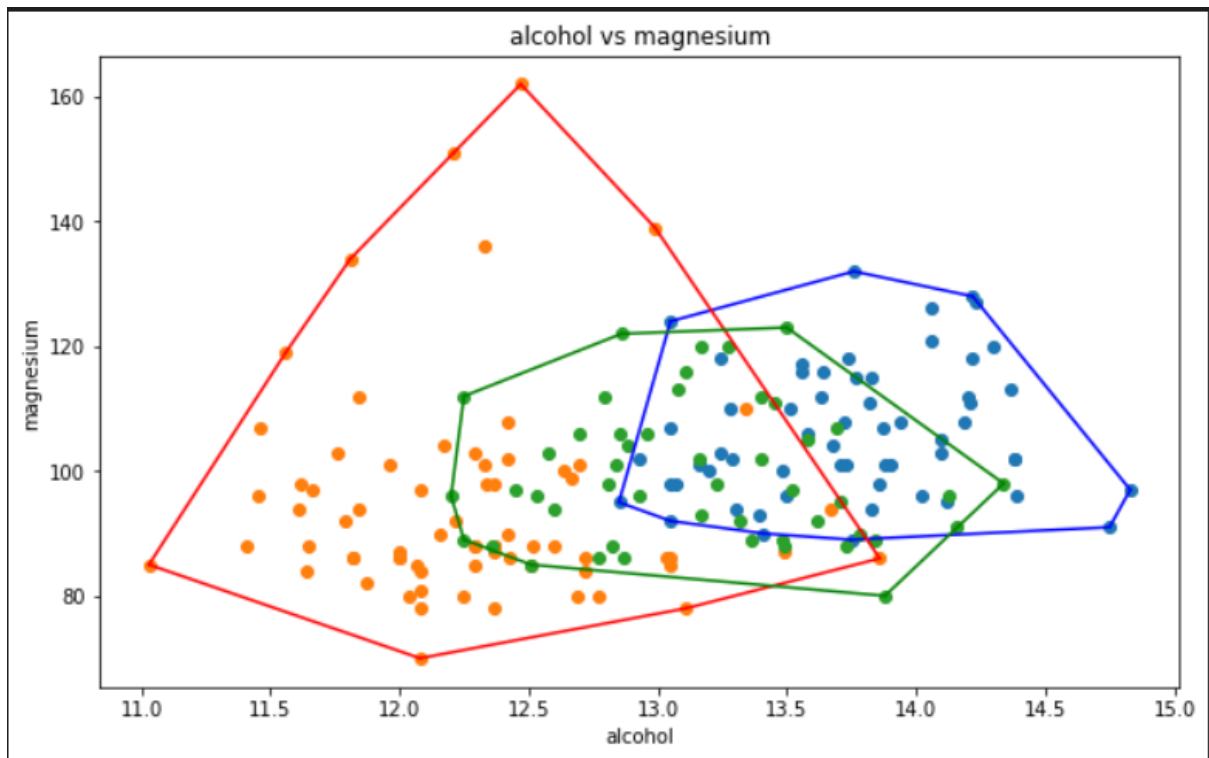
```
bucket = df[df['Target'] == i]
bucket = bucket.iloc[:,[a,b]].values
bucket2 = bucket.tolist()
hull = my.ConvexHull(bucket2)
plt.scatter(bucket[:, 0], bucket[:, 1], label=data.target_names[i])
for simplex in hull:
    plt.plot(bucket[simplex, 0], bucket[simplex, 1], colors[i%7])
```

Output

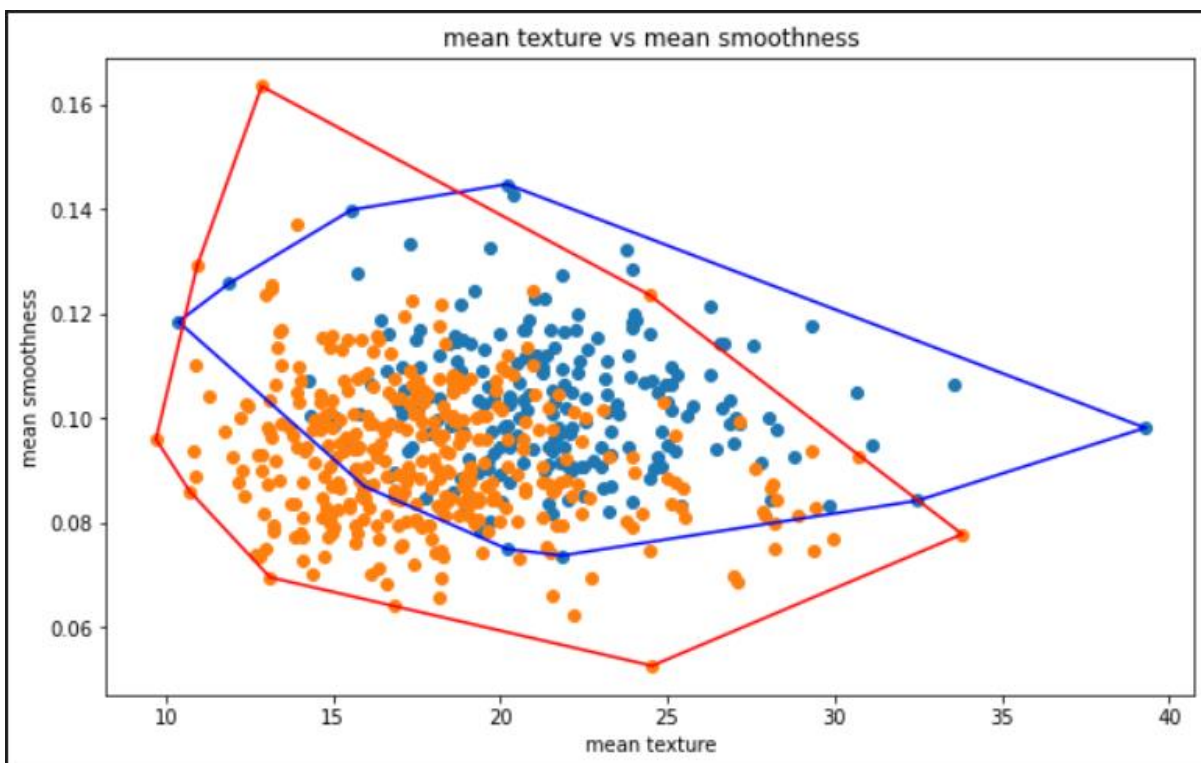
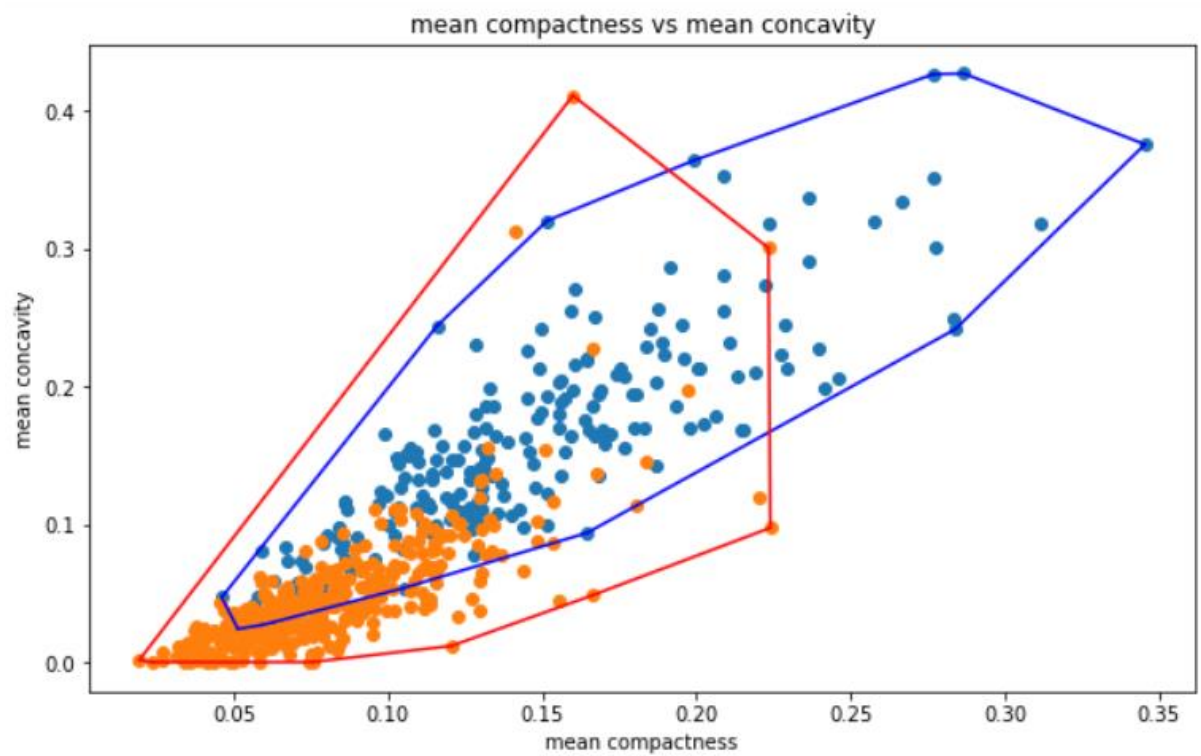
Dataset Iris



Dataset Wine



Dataset Breast Cancer



Referensi

- <https://codecrucks.com/convex-hull-using-divide-and-conquer/>
- [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-\(2022\)-Bagian4.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-(2022)-Bagian4.pdf)

Link Drive

- https://github.com/griendy15/Tucil2_Stima

Poin	Ya	Tidak
1. Pustaka myConvexHull berhasil dibuat dan tidak ada kesalahan	v	
2. Convex hull yang dihasilkan sudah benar	v	
3. Pustaka myConvexHull dapat digunakan untuk menampilkan convex hull setiap label dengan warna yang berbeda.	v	
4. Bonus: program dapat menerima input dan menuliskan output untuk dataset lainnya.	v	