LAPORAN TUGAS KECIL

Implementasi Convex Hull untuk Visualisasi Tes *Linear Separability Dataset* dengan Algoritma *Divide and Conquer*

Ditujukan untuk memenuhi salah satu tugas kecil mata kuliah IF2211 Strategi Algoritma pada Semester II Tahun Akademik 2021/2022

Disusun oleh:

Andhika Arta Aryanto (K3) **13520081**

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

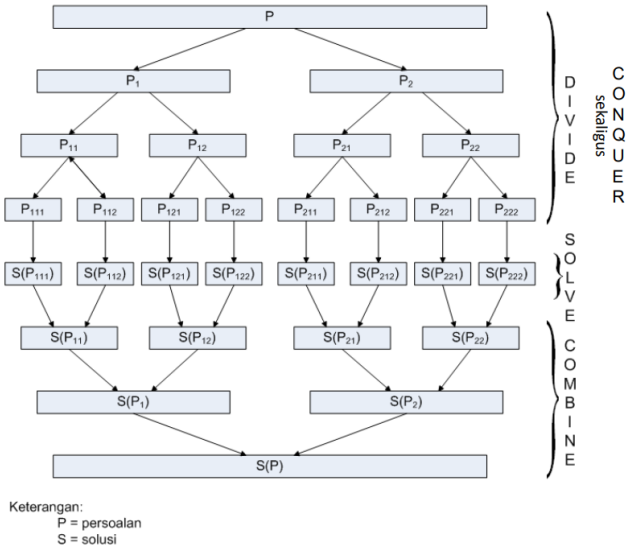
BAB I

PENJELASAN ALGORITMA PROGRAM

1.1 Divide and Conquer

Algoritma Divide and Conquer adalah strategi pemecahan masalah yang besar dengan cara melakukan pembagian masalah yang besar tersebut menjadi beberapa bagian yang lebih kecil secara rekursif hingga masalah tersebut dapat dipecahkan secara langsung, Solusi yang didapat dari setiap bagian kemudian digabungkan untuk membentuk sebuah solusi yang utuh. Untuk defisini Divide dan Conquer itu sendiri :

- 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 : menggabungkan solusi masing – masing upa – persoalan sehingga membentuk solusi persoalan semula

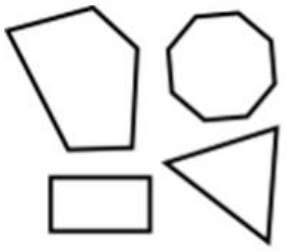


Obyek persoalan yang dibagi : masukan (input) atau *instances* persoalan yang berukuran n seperti : tabel, matriks, eksponen, polinom, dan masih banyak lagi. Tiap upa persoalan setelah dilakukan divide memiliki karakteristik yang sama dengan karakteristik persoalan semula sehingga metode *Divide and Conquer* lebih natural diungkapan dalam skema rekursif sebagai berikut :

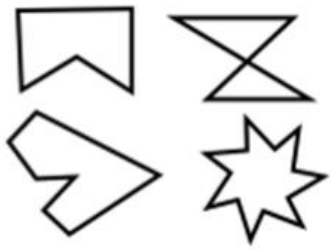
```
procedure DIVIDEandCONQUER(input P : problem, n : integer)
{ Menyelesaikan persoalan P dengan algoritma divide and conquer
  Masukan: masukan persoalan P berukuran n
  Luaran: solusi dari persoalan semula   }
Deklarasi
   r : integer

Algoritma
  if n ≤ n0 then  {ukuran persoalan P sudah cukup kecil }
     SOLVE  persoalan P yang berukuran n ini
  else
     DIVIDE  menjadi r upa-persoalan, P1, P2, …, Pr , yang masing-masing berukuran n1, n2, …, nr
     for  masing-masing P1, P2, …, Pr , do
        DIVIDEandCONQUER(Pi,  ni)
     endfor
     COMBINE  solusi dari P1, P2, …, Pr  menjadi solusi persoalan semula
  endif
```

1.2 Menyelesaikan persoalan Convex Hull dengan Divide and Conquer

Himpunan titik pada bidang planar disebut convex jika untuk sembarang dua titik pada bidang tersebut (misal p dan q) , seluruh segmen garis yang berakhir di p dan q berada pada himpunan tersebut



Gambar 1: convex



Gambar 2:non convex

Pada tucil kali ini, diberikan tugas untuk membuat library yang dapat menyelesaikan persoalan Convex Hull menggunakan algoritma Divide and Conquer. Secara umum, algoritma ini akan mencari dua titik terjauh dan dilakukan divide dengan membagi titik – titik menjadi 2 yaitu bagian atas convex dan bawah, setelah dilakukan pembagian, Conquer dilakukan dengan pencarian titik terjauh untuk dimasukkan ke dalam convex, hal ini akan dilakukan terus menerus sampai tidak ada lagi titik yang masuk menjadi bagian atas/bawah convex. Library berhasil dibuat dengan fungsi utama bernama ConvexHull. Berikut langkah detail yang dilakukan dalam library yang telah dibuat :

1. Fungsi ConvexHull menerima sekumpulan titik (ndarray) yang berasal dari suatu dataset
2. Dibuat array *solution* untuk menampung titik – titik yang akan menjadi convex hull
3. Dilakukan quicksort pada array menggunakan library python berdasarkan absis yang menaik (ordinat akan dibandingkan apabila absis sama)
4. Setelah sort dilakukan , index pertama dan terakhir menjadi p1 dan pn , dimasukkan ke array solusi, dan dihapus dari kumpulan titik awal

5. Digunakan fungsi aboveOrBelow untuk membuat 2 array baru , satu untuk yang berada diatas garis p1-pn dan satu lagi untuk yang dibawah
6. Pengelompokkan ini dilakukan dengan membuat garis p1-pn menjadi fungsi garis y = mx + c dan melakukan substitusi titik yang dikategorikan. Apabila hasil yang didapat adalah y0 > mx0 + c berarti titik tersebut berada diatas garis dan dimasukkan ke array s1 , dan sebaliknya akan dimasukkan ke array s2
7. Kedua array tersebut akan dimasukkan kedalam fungsi *recursiveconvex*, fungsi ini adalah fungsi rekursif yang akan terus menerus melakukan Divide and Conquer sampai list yang diterima sudah kosong (menandakan semua titik sudah diproses dan rekursif berhenti)
8. Pada awalnya akan dicari titik terjauh dari garis p1-pn, dan titik tersebut diberi nama *farthest* dan dimasukkan ke array solusi serta dihapus dari array awal
9. Setelah ditemukan , fungsi aboveOrBelow digunakan lagi, namun untuk garis yang berbeda yaitu p1-*farthest* dan *farthest*-p2. Lalu fungsi *recursiveconvex* akan dipanggil lagi untuk garis dan array titik yang baru
10. Fungsi *recursiveconvex* ini akan terus dipanggil , dengan setiap iterasi akan menghasilkan *farthest* baru untuk dimasukkan ke dalam kumpulan solusi dan membuat *farthest* menjadi p1 , p2 berikutnya.
11. Fungsi rekursif ini akan berakhir saat menerima array kosong, menandakan sudah tidak ada titik yang berada diatas ataupun bawah garis yang baru dibuat, menandakan garis sudah merupakan bagian paling luar dan merupakan convex untuk kumpulan titik tersebut
12. Kumpulan titik disort berdasarkan sudutnya dengan sebuah titik tengah agar membentuk array yang clockwise. Hal ini dilakukan untuk keperluan plotting. Setelah itu akan dikembalikan array solusi yang berisi semua titik yang merupakan bagian dari convex

Setelah array dari titik yang menjadi convex dibuat, terdapat file main.py yang memiliki tugas untuk menerima dataset, mengubah dataset menjadi kumpulan titik, memasukkan kumpulan titik ke fungsi yang telah dibuat, dan menampilkan hasil ConvexHull yang telah dicari.

Dalam library, terdapat fungsi – fungsi lain yang membantu pencarian convex, berikut fungsi – fungsi tersebut :

FUNGSI	**KEGUNAAN**
ConvexHull(numpyarray)	Fungsi utama yang akan mengembalikan kumpulan titik yang merupakan bagian dari convex hull
recursiveconvex(p1,p2,points,int)	Fungsi rekursif yang melakukan Divide and Conquer sampai semua titik berhasil diproses

aboveOrBelow(p1,p2,array)	Mencari kumpulan titik yang berada diatas atau dibawah garis p1-p2
distance(p1,p2,p3)	Menghitung jarak antara titik p3 dengan garis p1p2
farthestpoint(p1,p2,c)	Mencari titik terjauh pada array c dari garis p1p2
findAngle(p1,pmax,p2)	Mencari sudut yang dibentuk dari 3 titik p1,pmax,dan p2

BAB II

IMPLEMENTASI PROGRAM

2.1 Source Program dalam Python

2.1.1 Library ConvexHull.py

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn import datasets

def ConvexHull(numpyarray) :
    solution = []
    points = numpyarray.tolist()

    #Sort array untuk mencari 2 titik terjauh awal
    sortedArray = sorted(points , key=lambda k: [k[0], k[1]])
    p1 = sortedArray[0]
    pn = sortedArray[-1]
    solution += [p1,pn]

    #Setelah dimasukkan ke array solusi, hilangkan kedua titik
    sortedArray.pop(0)
    sortedArray.pop(-1)

    #Membagi kumpulan titik menjadi convex hull atas dan bawah
    s1,s2 = aboveOrBelow(p1,pn,sortedArray)

    #Dilakukan rekursi(divide terus menerus)
    solution += recursiveconvex(p1,pn,s1,1)
    solution += recursiveconvex(p1,pn,s2,2)

    solution = np.asarray(solution)


#Agar garis yang dibentuk membentuk poligon mengelilingi sempurna, titik - titik di sort berdasarkan sudut yang dibuat dengan titik tengah
    x = [p[0] for p in solution]
    y = [p[1] for p in solution]
    center = (sum(x) / len(solution), sum(y) / len(solution))
    solution = sorted(solution,key = lambda k: (math.atan2(k[1]-center[1],k[0]-center[0])))
    solution.append(solution[0])
    return solution
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```python
def recursiveconvex(p1,p2,points,int) :

#Fungsi rekursif untuk mencari titik2 convex hull lainnya, int disini menandakan arah , 1 berarti atas(kasus convex hull atas) , 2 bawah (convex hull bawah)
    #Exit case untuk rekursif
    if points == [] or p1 is None or p2 is None :
        return[]

    solution = []
    #Mencari titik terjauh dari garis
    farthest = farthestpoint(p1,p2,points)
    solution += [farthest]
    points.remove(farthest)


#Membagi kumpulan titik menjadi atas dan bawah dengan garis yang baru dibuat
    above1, below1 = aboveOrBelow(p1,farthest,points)
    above2, below2 = aboveOrBelow(p2,farthest,points)

    #Rekursif
    if int == 1 :
        solution += recursiveconvex(p1,farthest,above1,1)
        solution += recursiveconvex(farthest,p2,above2,1)
    else :
        solution += recursiveconvex(p1,farthest,below1,2)
        solution += recursiveconvex(farthest,p2,below2,2)

    return solution

```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```python
def distance(p1,p2,p3) :
    #fungsi menghitung jarak dari garis p1-p2 ke titik p3
    p1 = np.asarray(p1)
    p2 = np.asarray(p2)
    p3 = np.asarray(p3)
    return np.abs(np.cross(p2-p1, p1-p3)) / np.linalg.norm(p2-p1)

def farthestpoint(p1,p2,c) :
    #Mencari titik terjauh ke garis p1-p2 dari kumpulan titik c
    max = -1
    farthest = None
    for point in c :
        temp = distance(p1,p2,point)
        if temp > max :
            max = temp
            farthest = point
        elif temp == max :
            if(findAngle(p1,point,p2) > findAngle(p1,farthest,p2)) :
                farthest = point
                max = temp
    return farthest


def findAngle(p1,pmax,p2) :

    #mencari sudut antara ketiga titik, untuk kasus ini berarti sudut antara garis p1-pmax dan pmax-p2
    ang = math.degrees(math.atan2(p2[1]-pmax[1], p2[0]-pmax[0]) - math.atan2(p1[1]-pmax[1], p1[0]-pmax[0]))
    return ang + 360 if ang < 0 else ang

```

```python
def aboveOrBelow(p1,p2,array) :

    #Fungsi untuk mencari kumpulan titik yang menjadi bagian atas atau bawah suatu garis dalam convex hull
    s1 = []
    s2 = []

    #X sama, garis vertikal tidak ada atas bawah
    if p2[0] - p1[0] == 0 :
        return s1,s2


    #Membuat garis menjadi rumus umum y = mx + c, setiap poin akan disubsitusi, apabila y0 > mx0 + c berarti dia diatas dan sebaliknya
    grad = (p2[1] - p1[1]) / (p2[0] - p1[0])
    c = p1[1] - (grad * p1[0])
    for points in array :
        if points[1] < grad * (points[0]) + c :
            s2.append(points)
        if points[1] > grad * (points[0]) + c :
            s1.append(points)

    return s1,s2
```

2.2.1 main.py

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
import matplotlib.pyplot as plt
from ConvexHull import ConvexHull
from os import path

print("================================================")
print("Pilihan dataset :")
print("DATASET SCIKIT LEARN")
print("1 Iris")
print("2 Wine")
print("3 Breast Cancer")
print(
"Program ini juga bisa menerima file CSV, untuk melakukan ini mas
kkan CSV ke folder test, pastikan semua value dalam csv merupakan
 nomor"
)
print("4. CSV sendiri (ada target)")
print("5. CSV sendiri (tidak ada target)")
print("================================================")
cc = int(input())

if (cc == 1) :
    data = datasets.load_iris()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    hastarget = True
elif (cc == 2) :
    data = datasets.load_wine()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    hastarget = True
elif (cc == 3) :
    data = datasets.load_breast_cancer()
    df = pd.DataFrame(data.data, columns=data.feature_names)
    df['Target'] = pd.DataFrame(data.target)
    hastarget = True
elif (cc == 4) :
    data = input("Nama file csv = ")
    file = ".\\test\\" + data
    while not path.exists(file) :
        print("File tidak dapat ditemukan")
        data = input("Nama file csv  = ")
        file = ".\\test\\" + data
    df = pd.read_csv(file)
    if ('target') in df or ('Target') in df or ('TARGET') in df :
        hastarget = True
    if (hastarget == False) :
        print(
"CSV TIDAK ADA TARGET ! pilih 5 untuk csv tanpa target")
elif (cc == 5) :
    hastarget = False
    data = input("Nama file csv = ")
    file = ".\\test\\" + data
    while not path.exists(file) :
        print("File tidak dapat ditemukan")
        data = input("Nama file csv  = ")
        file = ".\\test\\" + data
    df = pd.read_csv(file)
df.head()
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```
if(hastarget) :
    if(cc == 4) :
        print(
"============================================================")
        print("Kolom - kolom pada dataset :")
        col = df.columns.values.tolist()
        i = 1
        for col_names in col :
            if (str.lower(col_names) != "target") :
                print(f"{i}.{col_names}")
                i += 1

        print(
"============================================================")
        print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
        X = 0
        Y = 0
        while (X <= 0 or X >= len(col)) :
            X = int(input("Kolom Yang menjadi X : "))
        while (Y <= 0 or Y >= len(col)) :
            Y = int(input("Kolom Yang menjadi Y : "))
        ## 4. INISIALISASI PYPLOT
        plt.figure(figsize = (10, 6))
        plt.title(label='ConveX Hull', fontsize=20)
        xname = col[X-1]
        yname = col[Y-1]
        plt.title(f'{xname} vs {yname}')
        plt.xlabel(col[X-1])
        plt.ylabel(col[Y-1])
        target = df.target.unique()
        print(X)
        print(Y)
        ## 5. MEMBUAT CONVEXHULL
        for i in range(len(target)) :
            bucket = df[df['target'] == target[i]]
            bucket = bucket.iloc[:,[X-1,Y-1]].values
            hull = ConvexHull(bucket)
            plt.scatter(bucket[:, 0], bucket[:, 1], label=
target[i])
            Xarray, Yarray = zip(*hull)
            plt.plot(Xarray, Yarray, '-o')
    elif(cc == 1 or cc ==2 or cc == 3) :
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

```
        print("============================================================")
        print("Kolom - kolom pada dataset :")
        col = df.columns.values.tolist()
        i = 1
        for col_names in col :
            if (str.lower(col_names) != "target") :
                print(f"{i}.{col_names}")
                i += 1

        print("============================================================")
        print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
        X = 0
        Y = 0
        while (X <= 0 or X >= len(col)) :
            X = int(input("Kolom Yang menjadi X : "))
        while (Y <= 0 or Y >= len(col)) :
            Y = int(input("Kolom Yang menjadi Y : "))
        plt.figure(figsize = (10, 6))
        xname = data.feature_names[X-1]
        yname = data.feature_names[Y-1]
        plt.title(f'{xname} vs {yname}')
        plt.xlabel(xname)
        plt.ylabel(yname)
        for i in range(len(data.target_names)):
            bucket = df[df['Target'] == i]
            bucket = bucket.iloc[:,[X-1,Y-1]].values
            hull = ConvexHull(bucket)
#bagian ini diganti dengan hasil implementasi
        #ConvexHull Divide & Conquer
            plt.scatter(bucket[:, 0], bucket[:, 1], label=
data.target_names[i])
            print(hull)
            x, y = zip(*hull)

            plt.plot(x, y, '-o')
    plt.legend()
    plt.show()
```

```
elif(hastarget == False and cc == 5) :
    print("============================================================")
    print("Kolom - kolom pada dataset :")
    col = df.columns.values.tolist()
    i = 1
    for col_names in col :
        if (str.lower(col_names) != "target") :
            print(f"{i}.{col_names}")
            i += 1

    print("============================================================")
    print("Pilih 2 kolom Yang ingin di plot (dalam angka):")
    X = 0
    Y = 0
    while (X <= 0 or X >= len(col)) :
        X = int(input("Kolom Yang menjadi X : "))
    while (Y <= 0 or Y >= len(col)) :
        Y = int(input("Kolom Yang menjadi Y : "))
    ## 4. INISIALISASI PYPLOT
    plt.figure(figsize = (10, 6))
    plt.title(label='ConveX Hull', fontsize=20)
    plt.xlabel(col[X-1])
    plt.ylabel(col[Y-1])
    print(X)
    print(Y)
    ## 5. MEMBUAT CONVEXHULL


    bucket = df.iloc[:,[X-1,Y-1]].values
    hull = ConvexHull(bucket)
    plt.scatter(bucket[:, 0], bucket[:, 1])
    Xarray, Yarray = zip(*hull)
    plt.plot(Xarray, Yarray, '-o')
    plt.show()

```

BAB III

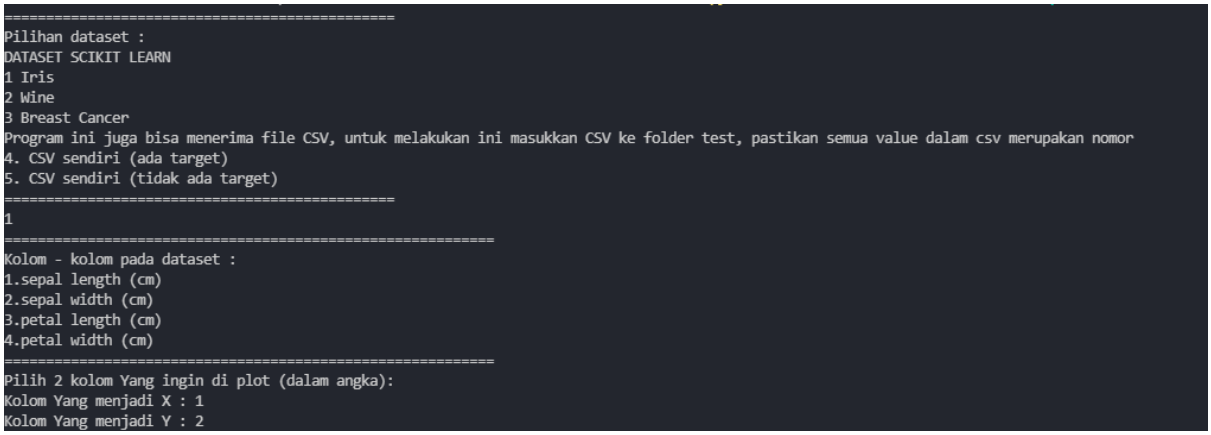
PENGUJIAN PROGRAM

3.1 Input dan Output Pengujian Program

Dilakukan pengujian library convex hull untuk 6 dataset berbeda, 3 dataset berasal dari modul python scikit learn dan 3 dataset lagi merupakan 3 CSV. Akan diambil 2 pasangan atribut random yang akan diplot dan dicari convex hullnya. Untuk dataset iris diambil atribut yang diminta oleh spek tucil ini , yaitu petal length vs petal width dan sepal length vs sepal width. Berikut hasil pengujian :

3.1.1 Interaksi awal Program

Berikut interaksi yang akan dilakukan pengguna saat awal memakai program



```
==========================================
Pilihan dataset :
DATASET SCIKIT LEARN
1 Iris
2 Wine
3 Breast Cancer
Program ini juga bisa menerima file CSV, untuk melakukan ini masukkan CSV ke folder test, pastikan semua value dalam csv merupakan nomor
4. CSV sendiri (ada target)
5. CSV sendiri (tidak ada target)
==========================================
4
Nama file csv = heart.csv
======================================================
Kolom - kolom pada dataset :
1.age
2.sex
3.cp
4.trestbps
5.chol
6.fbs
7.restecg
8.thalach
9.exang
10.oldpeak
11.slope
12.ca
13.thal
======================================================
Pilih 2 kolom Yang ingin di plot (dalam angka):
Kolom Yang menjadi X : 3
Kolom Yang menjadi Y : 4
3
4
```

3.1.2 Pengujian Dataset

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

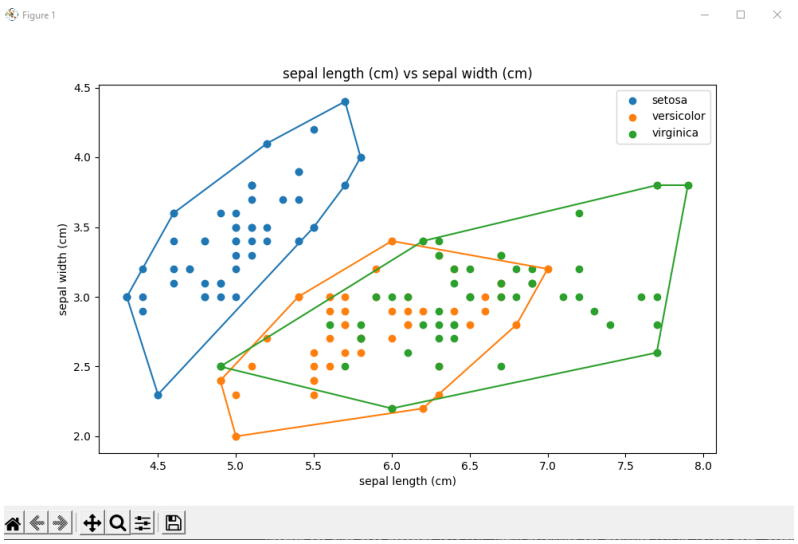
DATASET 1 : Iris

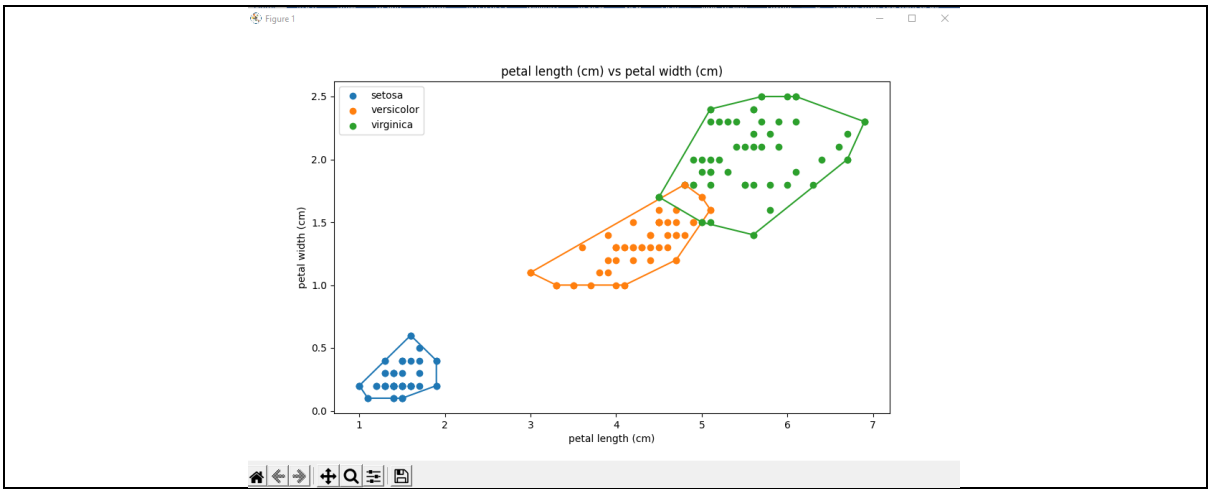
Nama Dataset :	***Pasangan Atribut :***
Iris	sepal-length, sepal-width petal-length, petal-width

Input

```
     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

     Target
0         0
1         0
2         0
3         0
4         0
..      ...
145       2
146       2
147       2
148       2
149       2

[150 rows x 5 columns]
```

Output





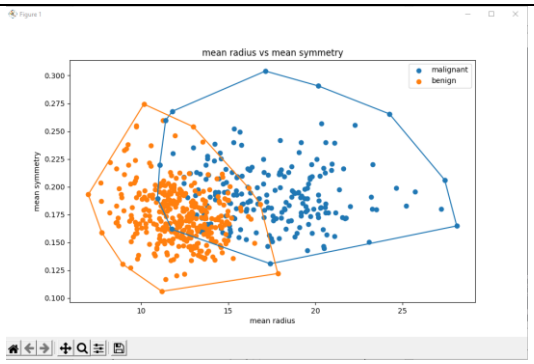
DATASET 2 : Breast_Cancer

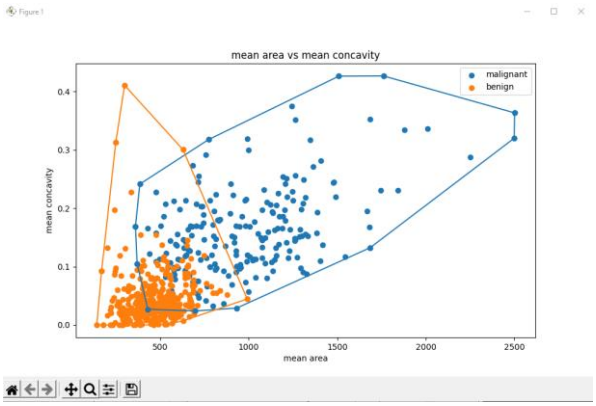
Nama Dataset :	***Pasangan Atribut :***
Breast Cancer	mean-radius, mean-symmetry mean-area, mean concavity

Input

```
     mean radius  mean texture  mean perimeter  mean area  mean smoothness  \
0          17.99         10.38          122.80     1001.0          0.11840
1          20.57         17.77          132.90     1326.0          0.08474
2          19.69         21.25          130.00     1203.0          0.10960
3          11.42         20.38           77.58      386.1          0.14250
4          20.29         14.34          135.10     1297.0          0.10030
..           ...           ...             ...        ...              ...
564        21.56         22.39          142.00     1479.0          0.11100
565        20.13         28.25          131.20     1261.0          0.09780
566        16.60         28.08          108.30      858.1          0.08455
567        20.60         29.33          140.10     1265.0          0.11780
568         7.76         24.54           47.92      181.0          0.05263

     mean compactness  mean concavity  mean concave points  mean symmetry  \
0             0.27760         0.30010              0.14710         0.2419
1             0.07864         0.08690              0.07017         0.1812
2             0.15990         0.19740              0.12790         0.2069
3             0.28390         0.24140              0.10520         0.2597
4             0.13280         0.19800              0.10430         0.1809
..                ...             ...                  ...            ...
564           0.11590         0.24390              0.13890         0.1726
565           0.10340         0.14400              0.09791         0.1752
566           0.10230         0.09251              0.05302         0.1590
567           0.27700         0.35140              0.15200         0.2397
568           0.04362         0.00000              0.00000         0.1587
...
566               0.1418          0.2218               0.07820       0
567               0.2650          0.4087               0.12400       0
568               0.0000          0.2871               0.07039       1

[569 rows x 31 columns]
```

Output





DATASET 3 : Wine

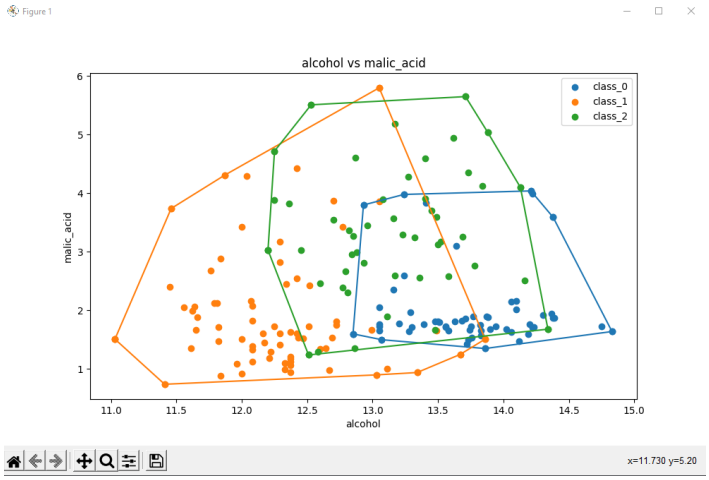
Nama Dataset :	***Pasangan Atribut :***
Wine	alcohol, malic-acid ash,magnesium

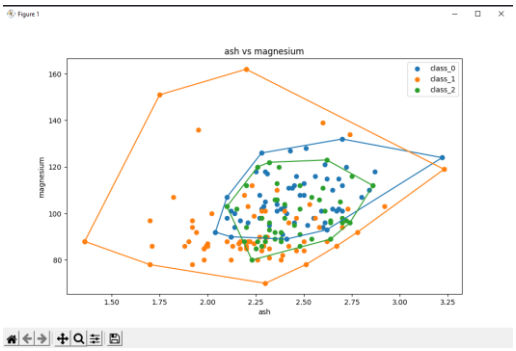
Input

```
     alcohol  malic_acid   ash  alcalinity_of_ash  magnesium  total_phenols  \
0      14.23        1.71  2.43               15.6      127.0           2.80
1      13.20        1.78  2.14               11.2      100.0           2.65
2      13.16        2.36  2.67               18.6      101.0           2.80
3      14.37        1.95  2.50               16.8      113.0           3.85
4      13.24        2.59  2.87               21.0      118.0           2.80
..       ...         ...   ...                ...        ...            ...
173    13.71        5.65  2.45               20.5       95.0           1.68
174    13.40        3.91  2.48               23.0      102.0           1.80
175    13.27        4.28  2.26               20.0      120.0           1.59
176    13.17        2.59  2.37               20.0      120.0           1.65
177    14.13        4.10  2.74               24.5       96.0           2.05

     flavanoids  nonflavanoid_phenols  proanthocyanins  color_intensity   hue  \
0          3.06                  0.28             2.29             5.64  1.04
1          2.76                  0.26             1.28             4.38  1.05
2          3.24                  0.30             2.81             5.68  1.03
3          3.49                  0.24             2.18             7.80  0.86
4          2.69                  0.39             1.82             4.32  1.04
..          ...                   ...              ...              ...   ...
173        0.61                  0.52             1.06             7.70  0.64
174        0.75                  0.43             1.41             7.30  0.70
175        0.69                  0.43             1.35            10.20  0.59
176        0.68                  0.53             1.46             9.30  0.60
177        0.76                  0.56             1.35             9.20  0.61
...
175                        1.56    835.0       2
176                        1.62    840.0       2
177                        1.60    560.0       2

[178 rows x 14 columns]
```

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

Output





Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

DATASET 4 : Banknote

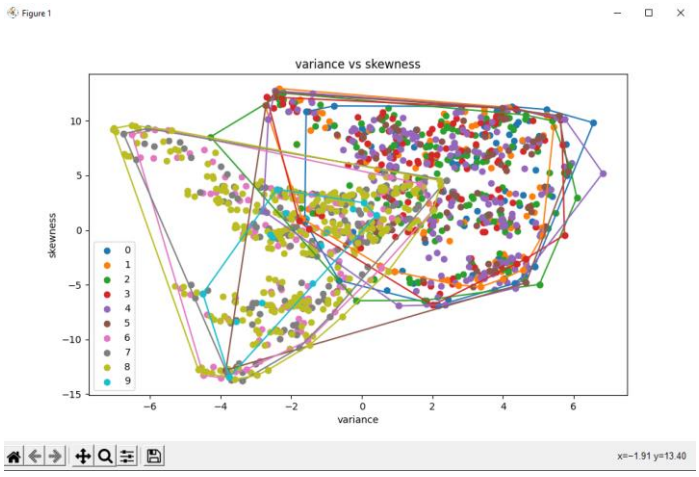
Nama Dataset :	***Pasangan Atribut :***
Banknote	variance, skewness kurtosis, entropy

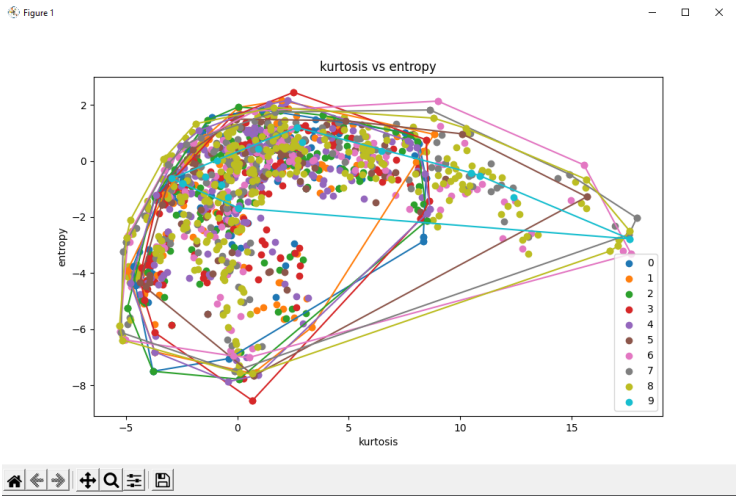
Input

	variance	skewness	kurtosis	entropy	target
0	3.62160	8.66610	-2.8073	-0.44699	0
1	4.54590	8.16740	-2.4586	-1.46210	0
2	3.86600	-2.63830	1.9242	0.10645	0
3	3.45660	9.52280	-4.0112	-3.59440	0
4	0.32924	-4.45520	4.5718	-0.98880	0
...	...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949	9
1368	-1.38870	-4.87730	6.4774	0.34179	9
1369	-3.75030	-13.45860	17.5932	-2.77710	9
1370	-3.56370	-8.38270	12.3930	-1.28230	9
1371	-2.54190	-0.65804	2.6842	1.19520	9

[1372 rows x 5 columns]

Output





Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

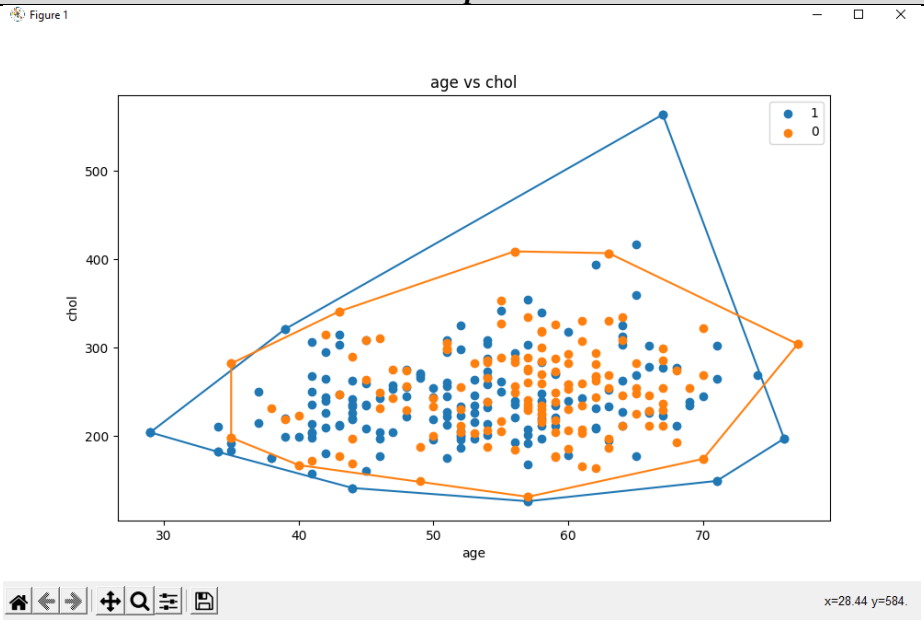
DATASET 5 : Heart

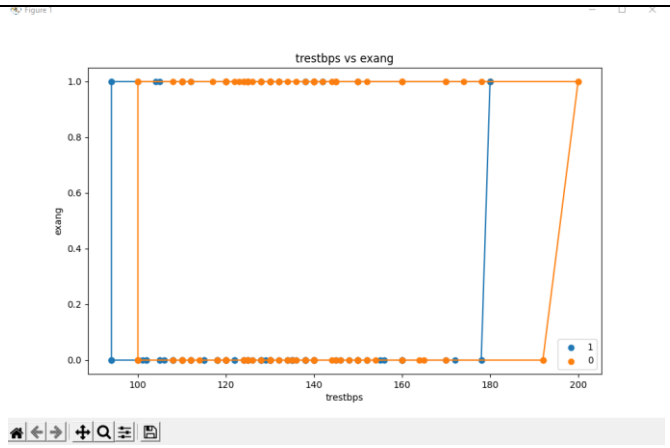
Nama Dataset :	***Pasangan Atribut :***
Heart	age, chol trestbps, exang

Input

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     ...  ..   ...     ...
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0

[303 rows x 14 columns]
```

Output





DATASET 6 : Indiansdiabetes

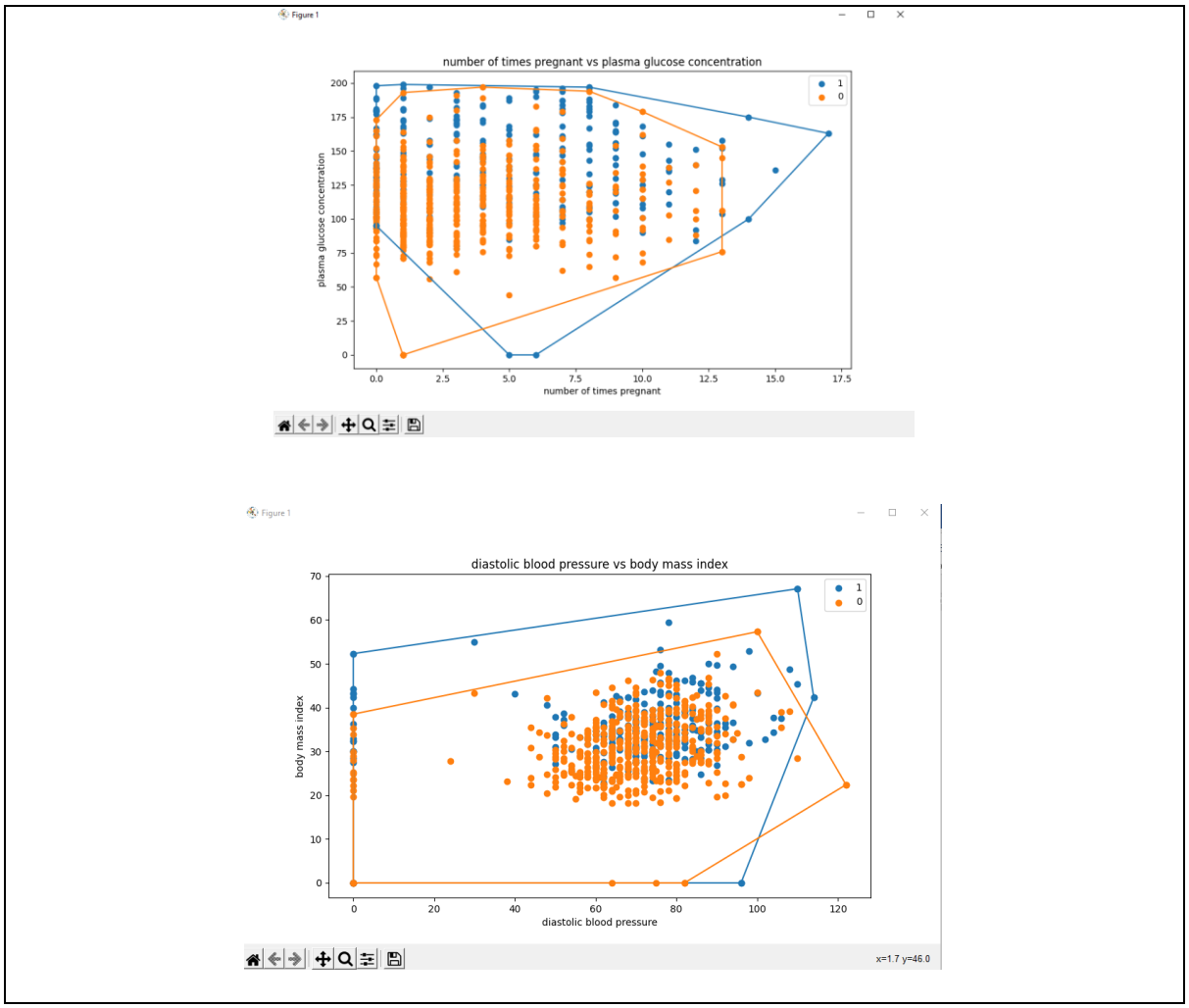
Nama Dataset :	***Pasangan Atribut :***
Indiansdiabetes	number-of-times-pregnant, plasma-glucose-concentration diastolic-blood-pressure, body-mass-index

Input

```
     number of times pregnant  plasma glucose concentration  \
0                           6                            148
1                           1                             85
2                           8                            183
3                           1                             89
4                           0                            137
..                        ...                            ...
763                        10                            101
764                         2                            122
765                         5                            121
766                         1                            126
767                         1                             93

     diastolic blood pressure  triceps skinfold thickness  \
0                          72                           35
1                          66                           29
2                          64                            0
3                          66                           23
4                          40                           35
..                        ...                          ...
763                        76                           48
764                        70                           27
765                        72                           23
766                        60                            0
767                        70                           31
...
765      0
766      1
767      0

[768 rows x 9 columns]
```

Output

Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma



Andhika Arta Aryanto
LAPORAN TUGAS KECIL
IF2211 Strategi Algoritma

LAMPIRAN

1. **REPOSITORY GITHUB :**
   https://github.com/dhikaarta/Convex-Hull-Library-and-Visualizer
2. **CHECKLIST:**

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

DAFTAR PUSTAKA

- Informatika.stei.itb.ac.id/~rinaldi.munir. (2022). Algoritma Divide and Conquer Bagian 1. Diakses pada 26 Februari 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Divide-and-Conquer-(2021)-Bagian1.pdf
- Informatika.stei.itb.ac.id/~rinaldi.munir. (2022). Algoritma Divide and Conquer Bagian 4. Diakses pada 26 Februari 2022, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Divide-and-Conquer-(2022)-Bagian4.pdf\
- https://www.geeksforgeeks.org/get-column-names-from-csv-using-python/
- https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-working-with-csv-files-in-python-with-pandas/