

Nama : Aldewajana Astinaputra Hasyim

Nim : 3332190038

Matakuliah : Kecerdasan Buatan

Ujian Tengah Semester

1. Analisa algoritma untuk *logistic_regression.py*. Dan analisa algoritmanya dan jalankan di komputer anda. (Untuk Chapter 2)
2. Analisa algoritma untuk *decision_trees.py*. Dan analisa algoritmanya dan jalankan di komputer anda. (Untuk Chapter 3)
3. Analisa algoritma untuk *mean_shift.py*. Dan analisa algoritmanya dan jalankan di komputer anda. (untuk Chapter 4)
4. Analisa algoritma untuk *nearest_neighbors_classifier.py*. Dan analisa algoritmanya dan jalankan di komputer anda (untuk Chapter 5)
5. Analisa algoritma untuk *states.py*. Dan analisa algoritmanya dan jalankan di komputer anda (untuk Chapter 6)

Jawab :

1. Logistic Regression

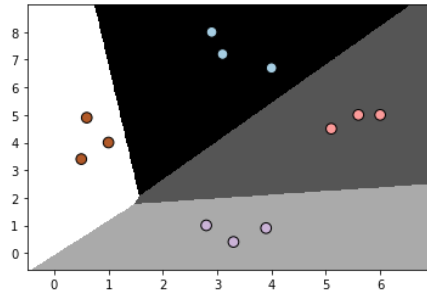
Dibawah ini adalah data sampel yang di input dengan code seperti berikut

```
# Define sample input data
x = np.array([[3.1, 7.2], [4, 6.7], [2.9, 8], [5.1, 4.5], [6, 5], [5.6, 5], [3.3, 0.4], [3.9, 0.9], [2.8, 1], [0.5, 3.4], [1, 4], [0.6, 4.9]])
y = np.array([0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3])
```

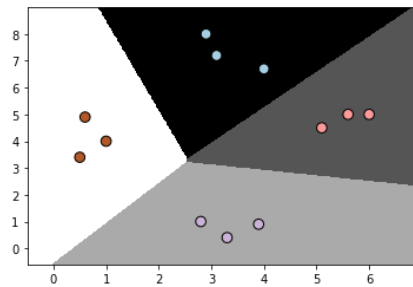
Lalu membuat sebuah *logistic regression classifier* dengan *script* berikut :

```
classifier = linear_model.LogisticRegression(solver='liblinear', C=1)
```

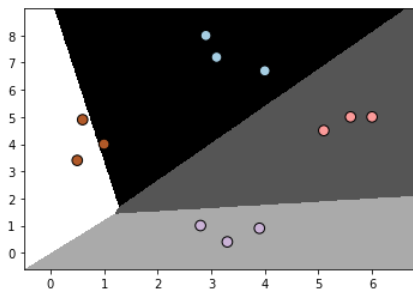
Dari hasil menggunakan C=1 didapatkan gambar



Gambar diatas merupakan hasil apabila menggunakan kurva yang bernilai 1 atau pada *classifier* nya dituliskan $C=1$. Untuk melihat perbandingannya diuji 1 kali lagi dengan nilai kurva yang berbeda yaitu $C = 100$



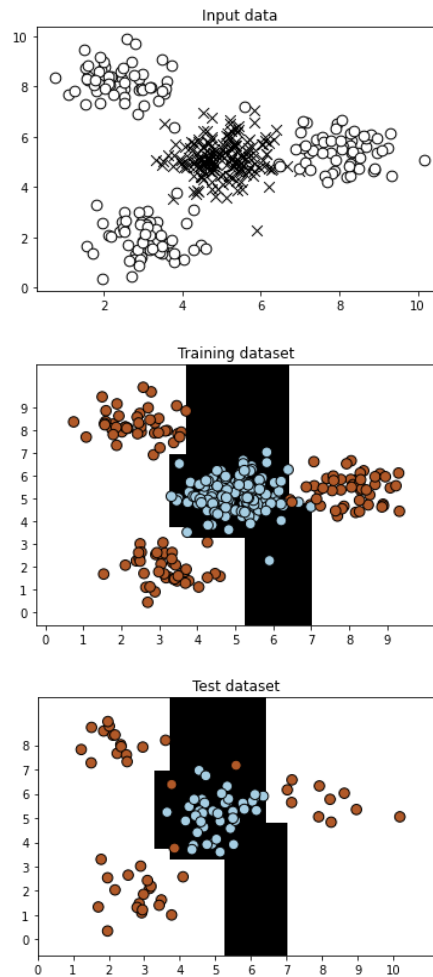
Dari perbandingan 2 grafik antara kurva bernilai 1 dan kurva bernilai 100 terdapat perbedaan yaitu pada bentuk output klasifikasinya atau disebut dengan area segmentasinya. Area tersebut dapat dilihat pada kurva 100 terlihat lebih rapih dan smooth dibandingkan dengan kurva yang bernilai 1. *Logistic regression* ini menampilkan 4 buah data yang kemudian algoritma akan mengklasifikasikan data tersebut dengan 4 area pada masing masing data.



Ketika mencoba dengan kurva bernilai 0.5 terdapat data yang keluar dari areanya yang seharusnya data tersebut masuk ke area berwarna putih, hal ini dinamakan sebagai *underfit*. *Underfit* sendiri terjadi ketika algoritma tidak dapat membedakan data sehingga data tersebut terklasifikasi di dua tempat.

2. Decision Trees

Untuk *decision trees script* yang digunakan berada di chapter 3. Setelah menginputkan *script* tersebut dan dijalankan didapat output seperti dibawah ini



```
#####
Classifier performance on training dataset

      precision    recall  f1-score   support

 Class-0       0.99      1.00      1.00       137
 Class-1       1.00      0.99      1.00       133

 accuracy              1.00       270
 macro avg           1.00      1.00      1.00       270
 weighted avg        1.00      1.00      1.00       270

#####
#####

Classifier performance on test dataset

      precision    recall  f1-score   support

 Class-0       0.93      1.00      0.97        43
 Class-1       1.00      0.94      0.97        47

 accuracy              0.97        90
 macro avg           0.97      0.97      0.97        90
 weighted avg        0.97      0.97      0.97        90

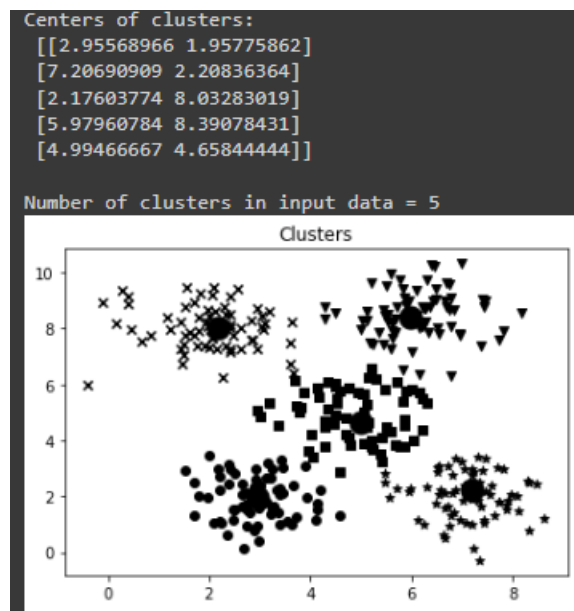
#####
```

Didapat data-data dari sebuah data acak yang digunakan untuk pengujian. Dari data terdapat istilah-istilah seperti *precision*, *recall*, *f1-score*, dan *support*. *Precision* merupakan ketepatan data menempati plot yang seharusnya, *recall* merupakan banyaknya data yang terpanggil, *f1-score* merupakan nilai harmonis rata-rata, dan *support* merupakan banyaknya data yang dimasukkan.

3. Data Clustering

Untuk *data clustering* AKA *meanshift script* yang digunakan terdapat pada chapter 4.

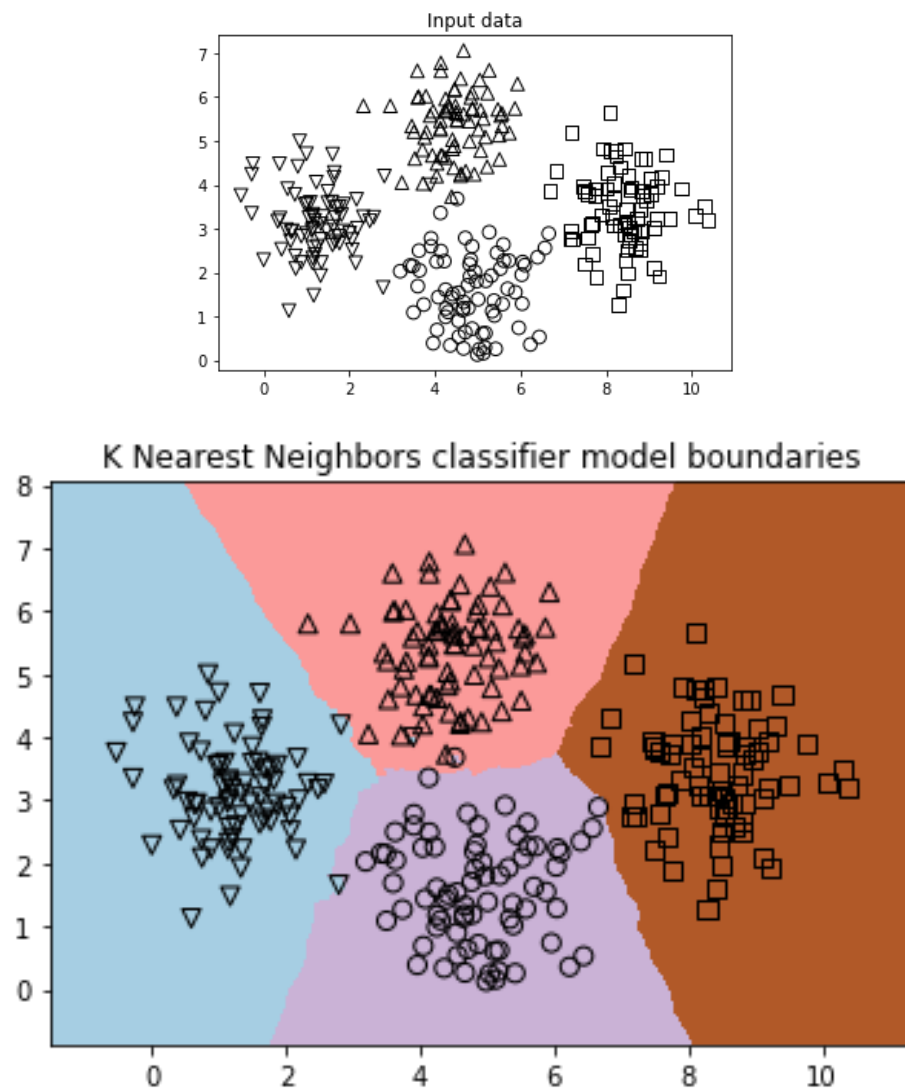
Berikut ini merupakan hasil yang didapat

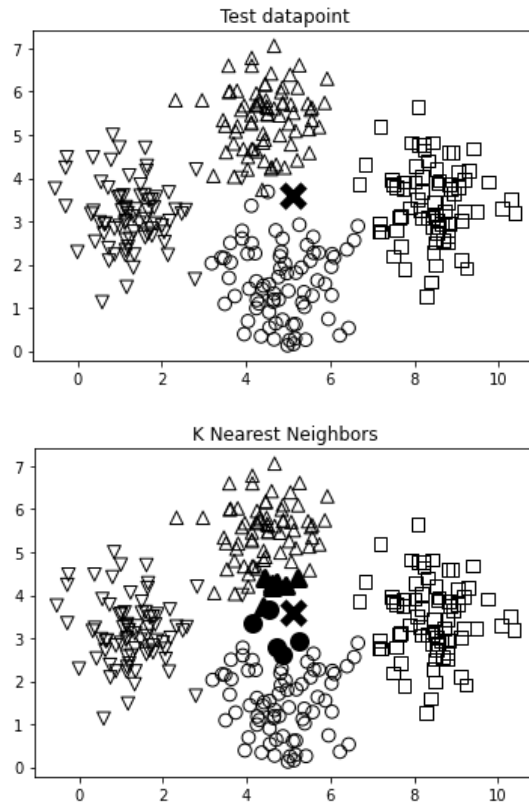


Data *clustering* dapat dinamakan juga sebagai pengelompokan data. Data yang didapat diketahui terdapat 5 buah pengelompokan untuk bentuk silang, segitiga, kotak, bintang, dan lingkaran. Dari ke 5 buah data algoritma mengelompokkan menjadi 5 buah *cluster* yang berdasarkan bentuk atau informasi yang dimilikinya. Di setiap *cluster* terdapat sebuah lingkaran besar seperti data yang menumpuk itu berarti merupakan titik tengah yang dimana sebuah data tersebut sangat mirip atau sangat mendekati dari informasinya.

4. K-Nearest Neighbor

Didapat hasil untuk K-Nearest Neighbor dengan menggunakan *script* dari chapter 5.





Prinsip dari *K-nearest neighbor* hampir sama dengan *clustering (meanshift)*. *K-nearest neighbor* atau kedekatan nilai tetangga berkerja seperti contoh sebuah data *dummy* atau data acak diinputkan yang kemudian di clusterisasi sehingga area dari segmentasinya ditentukan oleh kedekatan nilai tetangganya. Dilihat contoh pada hasil gambar yang berwarna pada area pink yang seharusnya ditempati bentuk segitiga ada sebuah lingkaran didalamnya. Artinya secara kedekatan masuk ke dalam segitiga tetapi secara bentuk masuk ke dalam lingkaran. Dalam kasus ini algoritma tersebut memprediksikan data tersebut kedalam segmen lingkaran dan terkadang kedalam segitiga (dapat dikatakan 50:50)

5. States

Didapat hasil untuk *states* ketika menggunakan *script* dari chapter 6.

```
Is Nevada adjacent to Louisiana?:  
No  
  
List of states adjacent to Oregon:  
Idaho  
Washington  
Nevada  
California  
  
List of coastal states adjacent to Mississippi:  
Alabama  
Louisiana  
  
List of 7 states that border a coastal state:  
Georgia  
Alabama  
New Jersey  
Tennessee  
Vermont  
Idaho  
Mississippi  
  
List of states that are adjacent to Arkansas and Kentucky:  
Missouri  
Tennessee
```

Didapatkan hasil bahwa algoritma dari *state* termasuk ke dalam aturan *rule base*. Di buktikan dengan sebuah pertanyaan yang pada kasus ini sebuah pertanyaan apakah Nevada bersebelahan dengan Louisiana ? pada data *adjacent_state.txt* tidak ada Nevada yang bersebelahan dengan Louisiana sehingga algoritma tersebut dapat menyatakan pertanyaan tersebut dengan pernyataan No. Selanjutnya pada daftar negara bagian dan kota yang bersebelahan dengan Oregon algoritma mengecek kembali *adjacent_state.txt* maka algoritma menyatakan dari pertanyaan tersebut dengan pernyataan Idaho, Nevada, Washington, dan California.

Selanjutnya pada daftar coastal state atau kota pesisir yang bersebelahan dengan Mississippi, algoritma mengecek data dari *coastal_state.txt* maka terdeteksi apa saja yang termasuk yang kemudian pada data *adjacent_state.txt* adakah data dari *coastal_state.txt* yang bersebelahan dengan Mississippi, menurut data yang bersebelahan yaitu Louisiana dan Alabama. Untuk dua pertanyaan terakhir menggunakan algoritma yang sama seperti pertanyaan sebelumnya, algoritma disini melihat data yang dimasukan dan menyesuaikan dengan data yang sudah dibuat. Jika diberikan sebuah pertanyaan maka algoritma akan mengeluarkan hasil sesuai rule base atau aturan sesuai dengan data yang dibuat.

Note tambahan :

- Untuk logistic regression, decision trees, dan Nearest neighbor termasuk kedalam supervised learning
- Untuk Data Clustering (Meanshift) termasuk kedalam supervised learning
- Untuk States termasuk kedalam Expert System