

A. Decision Tree

1. Langkah awal untuk membuat programnya adalah dengan cara mengimport library

```
desicion_tree.py > ...  
1 import matplotlib.pyplot as plt  
2 from subprocess import call  
3 from sklearn.tree import export_graphviz  
4 from sklearn.tree import DecisionTreeClassifier  
5 from sklearn.model_selection import train_test_split  
6 from sklearn import metrics, tree  
7 import pandas as pd
```

Gambar 1 Import

2. Kemudian import dataset yang di gunakan

```
dat = pd.read_csv('tic_tac_toe.csv')
```

Gambar 2 dataset

3. Pisahkan atribut dan label, x adalah atribut dan y adalah class labelnya

```
cols = ['top-left-square', 'top-middle-square', 'top-right-square', 'middle-left-square', 'mid  
x = dat[cols]  
y = dat['Class']
```

Gambar 3 pisahkan atribut dan label

4. Kemudian split data train dan data testing, 20% untuk data testing dan sisanya atau 80% data training

```
x_train, x_test, y_train, y_test = train_test_split(  
    x, y, test_size=0.2, random_state=1)  
dtree = DecisionTreeClassifier()  
dtree = dtree.fit(x_train, y_train)
```

Gambar 4 Split

5. Kemudian memprediksi apakah data testing bernilai benar atau salah

```
y_predict = dtree.predict(x_test)  
dtree.predict_proba(x_test)
```

Gambar 5 prediksi

6. Kemudian tampilkan akurasi klasifikasi

```
print("Akurasi klasifikasi Machine Learning : " + str(metrics.accuracy_score(y_test, y_predict))+"\n")
```

Gambar 6 akurasi

7. Kemudian tampilkan confusion matriksnya

```
print(metrics.confusion_matrix(y_test, y_predict))
```

Gambar 7 matriks

8. Kemudian tampilkan precision, recall dan fmeasure

```
print(metrics.classification_report(y_test, y_predict))
```

Gambar 8 precision, recall dan fmeasure

9. Buat plot treenya

```
export_graphviz(dtree, out_file='tree.dot',
                feature_names=['top-left-square', 'top-middle-square', 'top-right-square', 'middle-left-square',
                              'middle-right-square', 'bottom-left-square', 'bottom-middle-square', 'bottom-right'],
                class_names='Class',
                rounded=True, proportion=False,
                precision=2, filled=True)
```

Gambar 9 plot tree

10. Berikut hasilnya

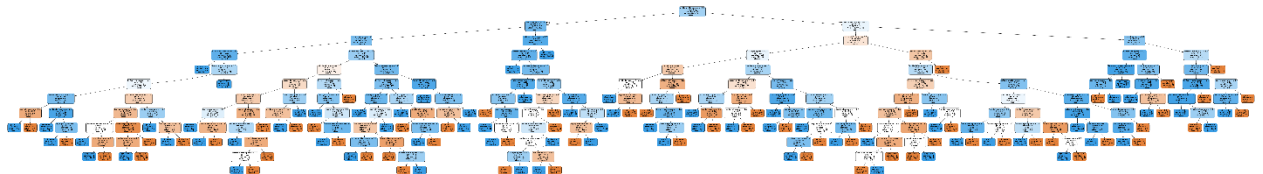
```
PS E:\BAHAN KULIAH\semester 5\DM\Uas Lab> python .\desicion_tree.py
Akurasi klasifikasi Machine Learning : 0.859375

[[ 50 17]
 [ 10 115]]
```

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| negative | 0.83 | 0.75 | 0.79 | 67 |
| positive | 0.87 | 0.92 | 0.89 | 125 |
| accuracy | | | 0.86 | 192 |
| macro avg | 0.85 | 0.83 | 0.84 | 192 |
| weighted avg | 0.86 | 0.86 | 0.86 | 192 |

Gambar 10 hasil

11. Berikut tree nya



Gambar 11 tree

B. Naïve Bayes

1. Langkah awal untuk membuat programnya adalah dengan cara mengimport library

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn import metrics
import pandas as pd
```

Gambar 1 import

2. Kemudian import dataset yang di gunakan

```
dat = pd.read_csv('tic_tac_toe.csv')
```

Gambar 2 dataset

3. Pisahkan atribut dan label, x adalah atribut dan y adalah class labelnya

```
cols = ['top-left-square', 'top-middle-square', 'top-right-square', 'middle-left-square', 'middle-middle-square',
        'middle-right-square', 'bottom-left-square', 'bottom-middle-square', 'bottom-right-square']
x = dat[cols]
y = dat['Class']
```

Gambar 3 pisahkan atribut dan class label

4. Kemudian split data train dan data testing, 20% untuk data testing dan sisanya atau 80% data training

```
x_train, x_test, y_train, y_test = train_test_split(
    x, y, test_size=0.2, random_state=1)
```

Gambar 4 split

5. Buat classifier

```
naive = GaussianNB()
```

Gambar 5 classifier

6. Kemudian latih dengan data training

```
naive = naive.fit(x_train, y_train)
```

Gambar 6 latih data

7. Kemudian memprediksi apakah data testing bernilai benar atau salah

```
y_predict = naive.predict(x_test)
naive.predict_proba(x_test)
```

Gambar 7 prediksi

8. Tampilkan akurasi

```
print("Akurasi klasifikasi Machine Learning : " +
      str(metrics.accuracy_score(y_test, y_predict))+"\n")
```

Gambar 8 akurasi

9. Tampilkan confusion matriksnya

```
print(metrics.confusion_matrix(y_test, y_predict))
```

Gambar 9 confusion

10. Kemudian tampilkan precision, recall dan fmeasure

```
print(metrics.classification_report(y_test, y_predict))
```

Gambar 10 precision, recall dan fmeasure

11. Hasil

```
PS E:\BAHAN KULIAH\semester 5\DM\Uas Lab> python .\naive_bayes.py
Akurasi klasifikasi Machine Learning : 0.7395833333333334

[[ 17  50]
 [  0 125]]
      precision    recall  f1-score   support

   negative       1.00      0.25      0.40         67
   positive       0.71      1.00      0.83        125

   accuracy              0.74         192
  macro avg       0.86      0.63      0.62         192
 weighted avg       0.81      0.74      0.68         192

PS E:\BAHAN KULIAH\semester 5\DM\Uas Lab> 
```

Gambar 11 hasil

Kesimpulan :

Pada dataset tic_tac_toe dapat kita lihat bahwa metode decision tree memiliki akurasi 85,9% dan naive bayes dengan akurasi 73,9%, jadi dari hasil tersebut dapat disimpulkan bahwa metode decision tree memiliki akurasi yang lebih tinggi daripada metode naïve bayes.

Decision Tree

Decision Tree adalah alat pendukung keputusan yang menggunakan model keputusan seperti pohon dan konsekuensinya yang mungkin, termasuk hasil acara kebetulan, biaya sumber daya, dan utilitas. Ini adalah salah satu cara untuk menampilkan algoritma yang hanya berisi pernyataan kontrol bersyarat.

Naive Bayes

Naive Bayes merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris Thomas Bayes. Algoritma Naive Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dr Naïve Bayes Classifier ini adalah asumsi yg sangat kuat (naïf) akan independensi dari masing-masing kondisi / kejadian. Naive Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan pada jurnal Xhemali, Daniela, Chris J. Hinde, and Roger G. Stone. "Naive Bayes vs. decision trees vs. neural networks in the classification of training web pages." (2009), mengatakan bahwa "Naïve Bayes Classifier memiliki tingkat akurasi yg lebih baik dibanding model classifier lainnya".