

## **Laporan Hasil Eksperimen Tugas Akhir Data Mining**



**Nama : A11.2019.12168**

**Nim : M Iqbal A A**

- **Judul**

“Penerapan Metode Clustering dengan K Naïve Bayes Untuk Memetakan Kumpulan Tim terkuat dari gelaran AFC Asian Cup”

- **Implementasi dengan koding python**

Pada ekpserimen ini akan di implementasikan menggunakan bahasa pemrograman python dan tools jupyter notebook.berikut ini adalah penjelasan langkah-langkah ,source code serta output yang dihasilkan saat eksperimen data mining dijalankan

1. pertama saya mengimport beberapa package/library yang dibutuhkan seperti numpy,matplotlib.pyplot dan juga pandas

a. import library

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

2. setelah itu import dataset yang telah didapatkam dan dimodifikasi, beri nama dataset menjadi asia\_cup\_team\_dataset dengan format file csv serta dibagi menjadi 2 bagian (x,y)

b. import dataset csv dan membagi menjadi 2 bagian (x,y)

```
In [2]: dataset = pd.read_csv('asia_cup_team_dataset.csv')
x = dataset.iloc[:, [3,5]].values
y = dataset.iloc[:, -1].values
```

3. menampilkan dataset dengan memanggil atribut kunci yang di dalamnya

c. import dataset dan menampilkan atribut kunci

```
In [3]: dataset = pd.read_csv('asia_cup_team_dataset.csv')
dataset.keys()
```

berikut hasil output dan beberapa atribut kunci yang ditampilkan

```
Out[3]: Index(['no ', 'team', 'menang', 'seri', 'kalah', 'point', 'jumlah gol',
              'status'],
              dtype='object')
```

4. menampilkan dan mengecek dataset secara keseluruhan.

d. cek dataset

```
In [4]: dataset = pd.read_csv('asia_cup_team_dataset.csv')
dataset
```

dibawah ini tampilan dari dataset secara keseluruhan.

	no	team	menang	seri	kalah	point	jumlah gol	status
0	1	Suriah	2	1	0	7	15	1
1	2	Australia	0	0	3	0	-8	0
2	3	Iran	1	1	1	4	9	0
3	4	Arab Saudi	2	0	1	6	5	0
4	5	Qatar	3	0	0	9	12	1
5	6	Jepang	3	0	0	9	20	1
6	7	Uni Emirat Arab	0	1	2	1	9	0
7	8	Korea Selatan	1	2	0	5	7	0
8	9	China	3	0	0	9	12	1
9	10	Oman	1	0	2	3	4	0
10	11	Irak	2	1	0	7	4	0
11	12	Vietnam	2	1	0	7	7	0
12	13	Lebanon	3	0	0	9	10	1
13	14	Yordania	0	0	3	0	-7	0
14	15	Palestina	2	1	0	7	8	0
15	16	Uzbekistan	0	2	1	2	6	0
16	17	India	3	0	0	9	9	1
17	18	Bahrain	2	1	0	7	4	0
18	19	Tajikistan	0	0	3	3	-3	0
19	20	Indonesia	1	1	1	4	2	0
20	21	Thailand	1	1	1	4	5	0
21	22	Hong Kong	2	0	1	6	4	0
22	23	Malaysia	1	1	1	4	6	0
23	24	Kyrgystan	2	1	0	7	1	0

- **Percobaan menggunakan metode klasifikasi Naive Bayes**

1. pertama – tama klasifikasikan naïve bayes dengan mengimport library train test split untuk membagi data menjadi 50% data test dan 50% data training ke dalam package sklearn.

a. mensplit dataset menjadi training set dan test set

```
In [5]: from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test, = train_test_split(x, y, test_size = 0.25, random_state = 0)
```

2. melakukan featur scaling pada saat preprocessing di dalam sklearn dengan mengimport library standartscaler, lalu membuat 2 variabel dari hasil split dataset sebelumnya menjadi x\_train dan x\_test.

b. melakukan scaling

```
In [6]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

3. pada tahap analisis naïve bayes ini dengan gaussian ,maka aktifkan terlebih dahulu packagenya dengan mengimport GaussianNB dan juga membuat fungsi klasifikasi naïve bayes

c. mengimport model naïve bayes di training set

```
In [7]: from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)
```

4. melakukan prediksi dari data testing bagian (x\_test)

d. memprediksi dari hasil test set

```
[8]: y_pred = classifier.predict(x_test)
```

5. setelah melakukan prediksi dari hasil prediksi x\_test ,maka tahap selanjutnya menentukan probabilitas hasil prediksi dengan confusion matrix untuk menentukan ketepatan prediksi.

e. membuat prediksi klasifikasi dengan confusion matrix

```
In [9]: from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(y_test, y_pred)  
print(cm)
```

dari hasil di bawah ini diperoleh prediksi dengan didapaknya probabilitas 6 tim terkuat sesuai dengan dataset.

[[6]]

6. membuat tabel confusion untuk mencoba melakukan klasifikasi prediksi dari data baru dengan data asli(dataset) yang telah di dapat dan dimodifikasi.

f. membuat tabel confusion

```
In [11]: y_actual = pd.Series([1,0,0,0,1,1,1,0,0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0], name = 'actual')  
y_pred = pd.Series([0,0,1,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0], name = 'prediction')  
df_confusion = pd.crosstab(y_actual, y_pred)  
df_confusion
```

berikut hasil yang dapat diperoleh dengan menggabungkan antara data asli pada dataset dengan prediksi terbaru :

1. simbol angka 0 di definisikan sebagai tim lemah/ tidak diunggulkan/terkuat di dapat prediksi 16 tim dari 24 benar
2. diprediksi hanya terdapat 2 tim terlemah dan prediksi tersebut adalah salah karena harusnya didapat lebih dari 2 tim terlemah.
3. terdapat 2tim terkuat /diunggulkan yang menjadi hasil prediksi dan ternyata salah karena harusnya terdapat 5 tim yang akan diklasifikasikan menjadi tim terkuat.
4. simbol angka 1 di definisikan sebagai kelompok tim terkuat dan didapat hasil prediksinya adalah 5 benar.

Out [11]:

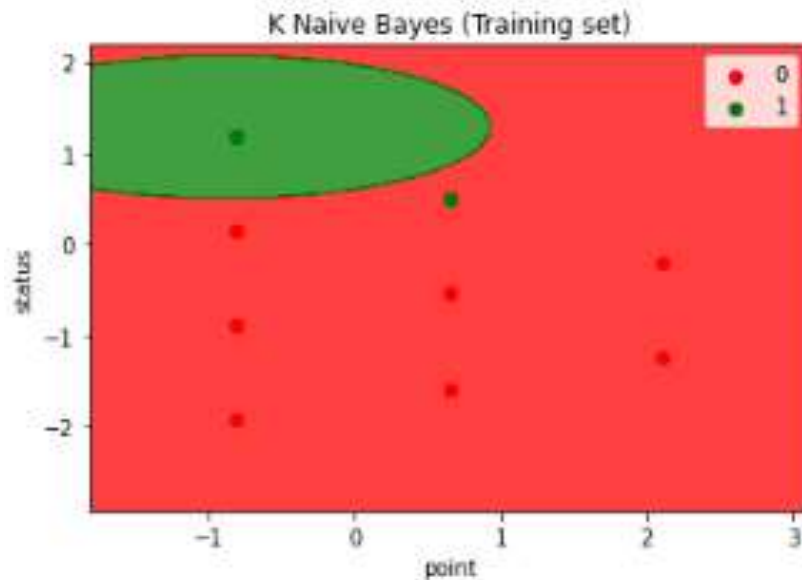
prediction		0	1
actual			
	0	16	2
	1	2	5

7. buat visualisasi dari hasil training set pada (x,y) dengan mengimport library ListedColormap dan mendefinisikan lingkaran warna merah = simbol 0 ( klasifikasi tim terlemah) serta mendefinisikan lingkaran warna hijau = simbol 1 (klasifikasi tim terkuat).

g. visualisasi training set

```
In [12]: from matplotlib.colors import ListedColormap
x_set, y_set = x_train, y_train
x1, x2 = np.meshgrid(np.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
plt.contourf(x1, x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
               c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K Naive Bayes (Training set)')
plt.xlabel('point')
plt.ylabel('status')
plt.legend()
plt.show()
```

Berikut ini adalah hasil visualisasi

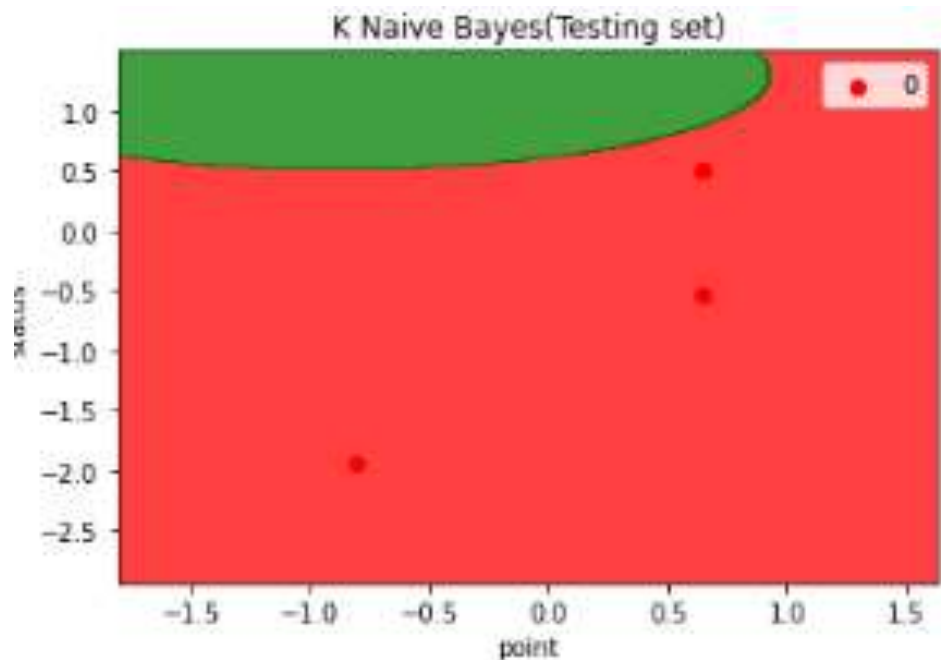


8. buat visualisasi dari hasil Testing set pada (x,y) dengan mengimport library ListedColormap dan mendefinisikan lingkaran warna merah = simbol 0 ( klasifikasi tim terlemah) serta mendefinisikan lingkaran warna hijau = simbol 1 (klasifikasi tim terkuat).

h. visualisasi testing set

```
In [13]: from matplotlib.colors import ListedColormap
x_set, y_set = x_test, y_test
x1, x2 = np.meshgrid(np.arange(start = x_set[:, 0].min() - 1, stop = x_set[:, 0].max() + 1, step = 0.01),
                     np.arange(start = x_set[:, 1].min() - 1, stop = x_set[:, 1].max() + 1, step = 0.01))
plt.contourf(x1, x2, classifier.predict(np.array([x1.ravel(), x2.ravel()]).T).reshape(x1.shape),
             alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(x1.min(), x1.max())
plt.ylim(x2.min(), x2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(x_set[y_set == j, 0], x_set[y_set == j, 1],
               c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('K Naive Bayes (Testing set)')
plt.xlabel('point')
plt.ylabel('status')
plt.legend()
plt.show()
```

Berikut hasil visualisasi hasil testing set



- **Kesimpulan**

Jadi dari percobaan diatas terdapat beberapa klasifikasi yang bermacam-macam hasilnya dan dapat dibandingkan jika ada cukup banyak tim yang diklasifikasikan adalah tim terlemah ataupun tidak diunggulkan dan hanya ada terdapat beberapa tim yang diklasifikasikan menjadi tim terkuat untuk tournament AFC Asia ini.

Dari permasalahan diatas mengenai klasifikasi prediksi tim terkuat dapat diselesaikan dengan teknik data mining metode Klasifikasi Naïve Bayes.