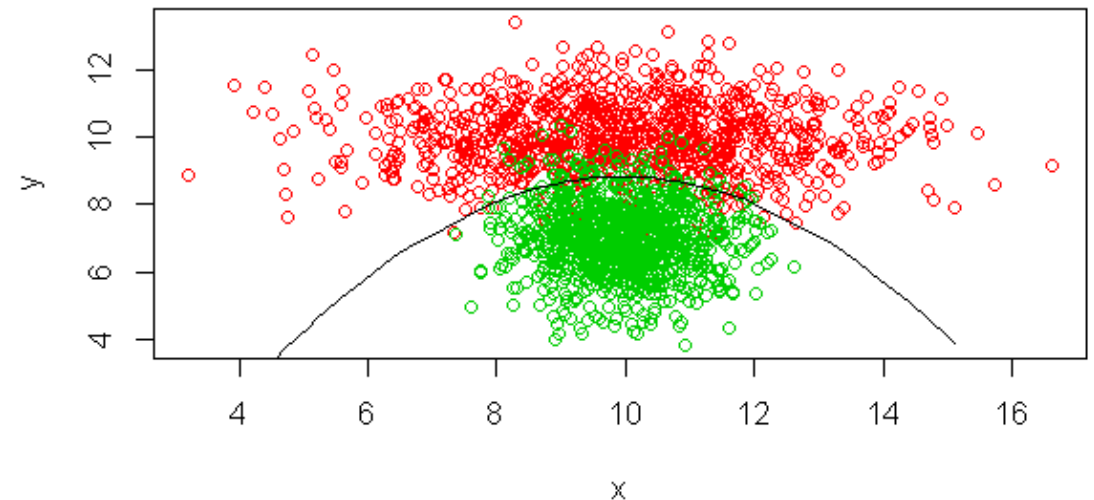


KLASIFIKASI

- Klasifikasi adalah algoritma yang menggunakan data dengan **target/class/label berupa nilai kategorikal (nominal)**
- Contoh, apabila **target/class/label** adalah pendapatan, maka bisa digunakan nilai nominal (kategorikal) sbb: pendapatan besar, menengah, kecil
- Contoh lain adalah rekomendasi contact lens, apakah menggunakan yang jenis **soft**, **hard** atau **none**
- Algoritma klasifikasi yang biasa digunakan adalah: Naive Bayes, K-Nearest Neighbor, C4.5, ID3, CART, Linear Discriminant Analysis, etc

BAYESIAN CLASSIFICATION

- Bayesian Classification adalah pengklasifikasian statistik yang dapat digunakan untuk memprediksi *probabilitas* keanggotaan suatu class.
- Bayesian Classification didasarkan pada *teorema Bayes* yang memiliki kemampuan klasifikasi serupa *decision tree* dan *neural network*.
- Bayesian Classification terbukti memiliki *akurasi dan kecepatan* yang tinggi saat diaplikasikan ke dalam database dengan data yang besar.



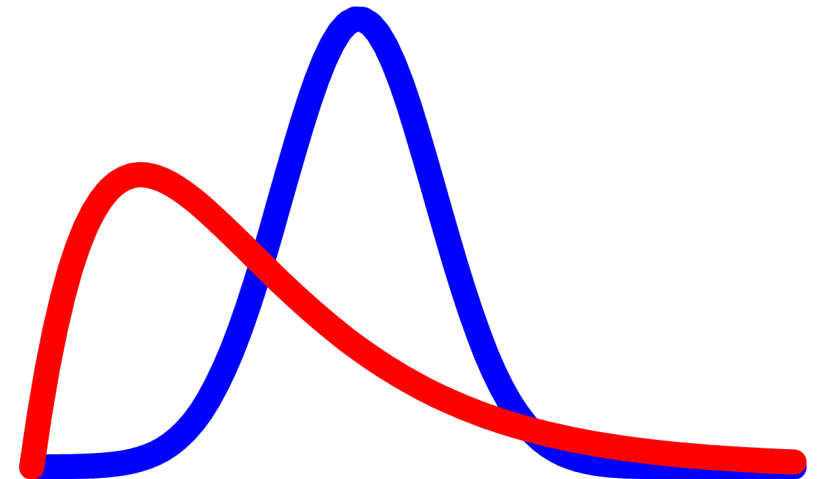
RUMUS TEOREMA BAYES

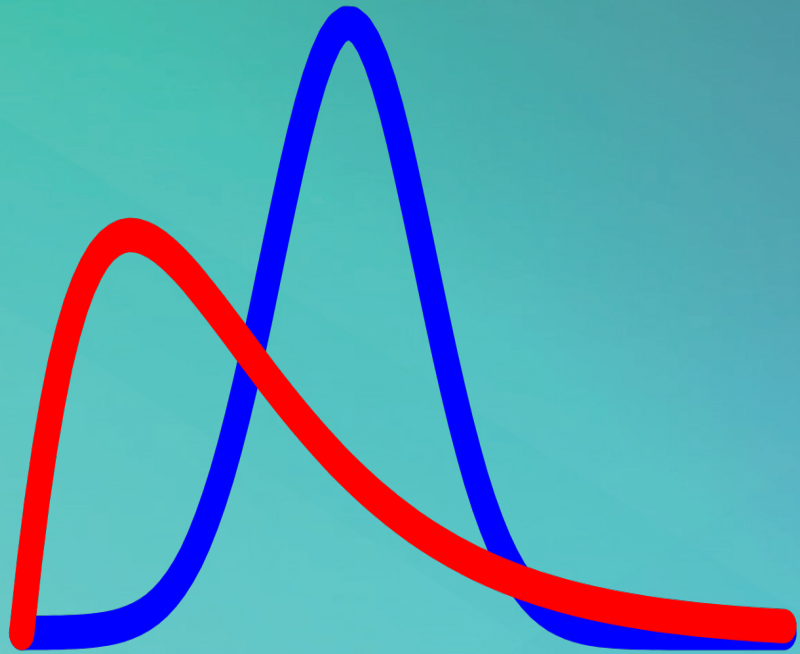
Teorema Bayes memiliki bentuk umum sbb :

$$P(H|X) = \frac{P(X|H) P(H)}{P(X)}$$

Keterangan :

- X : data dengan *class* yang belum diketahui
- H : hipotesis data X merupakan suatu *class* spesifik
- P(H|X) : probabilitas hipotesis H berdasar kondisi X
(*posteriori probability*)
- P(H) : probabilitas hipotesis H (*prior probability*)
- P(X|H) : probabilitas X berdasar kondisi hipotesis H
- P(X) : probabilitas dari X





Contoh Perhitungan

Klasifikasi menggunakan Naive Bayes

DATA TRAINING

Terdapat dua class dari
klasifikasi yang dibentuk
yaitu :

C1 => buys_computer = yes

C2 => buys_computer = no

Nilai X yang belum
diketahui Label / Kelas :

X =

(

age="<=30",
income="Medium",
student="Yes",
credit_rating="Fair";

)

$$P(C_i | \mathbf{X}) = P(C_i) \prod_{k=1}^N P(X_k = x_k | C_i)$$

Id	Age	Income	Student	Credit_rating	Class: buys_computer
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31..40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31..40	Low	Yes	Excellent	Yes
8	<=30	Medium	No	Fair	No
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31..40	Medium	No	Excellent	Yes
13	31..40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

PENYELESAIAN (1)

Dibutuhkan untuk memaksimalkan $P(X|C_i) P(C_i)$ untuk $i=1, 2$

- $P(C_i)$ merupakan prior probability untuk setiap class berdasar data contoh

$$P(\text{buys_computer}=\text{"yes"}) = 9/14 = 0.643$$

$$P(\text{buys_computer}=\text{"no"}) = 5/14 = 0.357$$

- Hitung $P(X|C_i)$ untuk $i=1, 2$

$$P(\text{age}=\text{"<=30"} | \text{buys_computer}=\text{"yes"}) = 2/9 = 0.222$$

$$P(\text{age}=\text{"<=30"} | \text{buys_computer}=\text{"no"}) = 3/5 = 0.600$$

$$P(\text{income}=\text{"medium"} | \text{buys_computer}=\text{"yes"}) = 4/9 = 0.444$$

$$P(\text{income}=\text{"medium"} | \text{buys_computer}=\text{"no"}) = 2/5 = 0.400$$

$$P(\text{student}=\text{"yes"} | \text{buys_computer}=\text{"yes"}) = 6/9 = 0.667$$

$$P(\text{student}=\text{"yes"} | \text{buys_computer}=\text{"no"}) = 1/5 = 0.200$$

$$P(\text{credit_rating}=\text{"fair"} | \text{buys_computer}=\text{"yes"}) = 6/9 = 0.667$$

$$P(\text{credit_rating}=\text{"fair"} | \text{buys_computer}=\text{"no"}) = 2/5 = 0.400$$

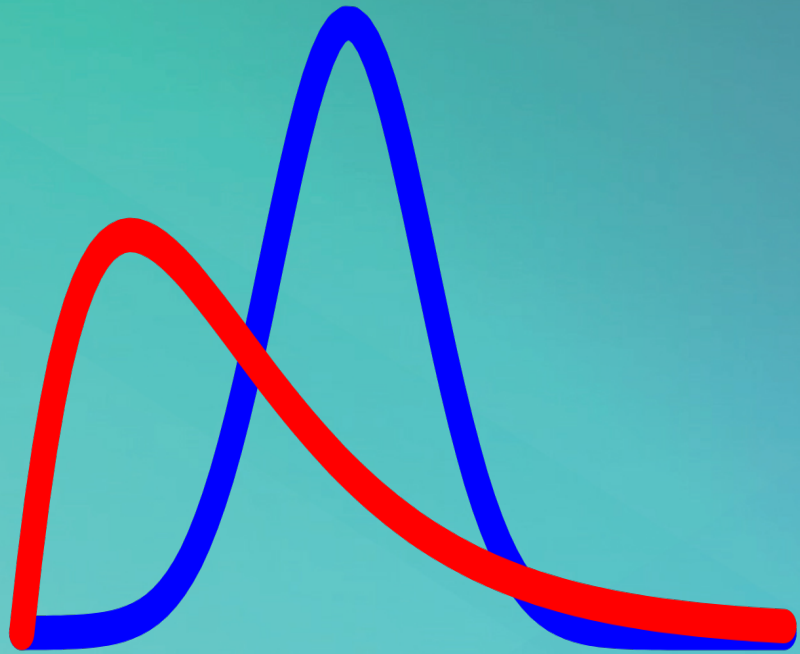
PENYELESAIAN (2)

$$P(X | \text{buys_computer} = \text{"yes"}) = 0.222 * 0.444 * 0.677 * 0.677 \\ = 0.044$$

$$P(X | \text{buys_computer} = \text{"no"}) = 0.600 * 0.400 * 0.200 * 0.400 \\ = 0.019$$

- $P(X | \text{buys_computer} = \text{"yes"}) P(\text{buys_computer} = \text{"yes"})$
 $= 0.044 * 0.643 = \mathbf{0.028}$
- $P(X | \text{buys_computer} = \text{"no"}) P(\text{buys_computer} = \text{"no"})$
 $= 0.019 * 0.357 = 0.007$

Kesimpulan: ***buys_computer = "yes"***



Implementasi Klasifikasi Naive Bayes dengan python

Dataset

dataset - DataFrame						
Index	User ID	Gender	Age	EstimatedSalary	Purchased	
0	15624510	Male	19	19000	0	
1	15810944	Male	35	20000	0	
2	15668575	Female	26	43000	0	
3	15603246	Female	27	57000	0	
4	15804002	Male	19	76000	0	
5	15728773	Male	27	58000	0	
6	15598044	Female	27	84000	0	
7	15694829	Female	32	150000	1	
8	15600575	Male	25	33000	0	
9	15727311	Female	35	65000	0	
10	15570769	Female	26	80000	0	
11	15606274	Female	26	52000	0	
12	15746139	Male	20	86000	0	



Klik icon untuk download

Import library yang digunakan

```
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd
```

Library yang digunakan untuk contoh diatas adalah library **numpy** dan **pandas**.

Import Dataset

```
dataset = pd.read_csv('Social_Network_Ads.csv')  
X = dataset.iloc[:, [2, 3]].values  
y = dataset.iloc[:, -1].values
```

Splitting the dataset into the Training set and Test set

```
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)
```

Feature Scaling

```
from sklearn.preprocessing import StandardScaler  
sc = StandardScaler()  
X_train = sc.fit_transform(X_train)  
X_test = sc.transform(X_test)
```

Training the Naive Bayes model on the Training set

```
from sklearn.naive_bayes import GaussianNB  
classifier = GaussianNB()  
classifier.fit(X_train, y_train)
```

Predicting the Test set results

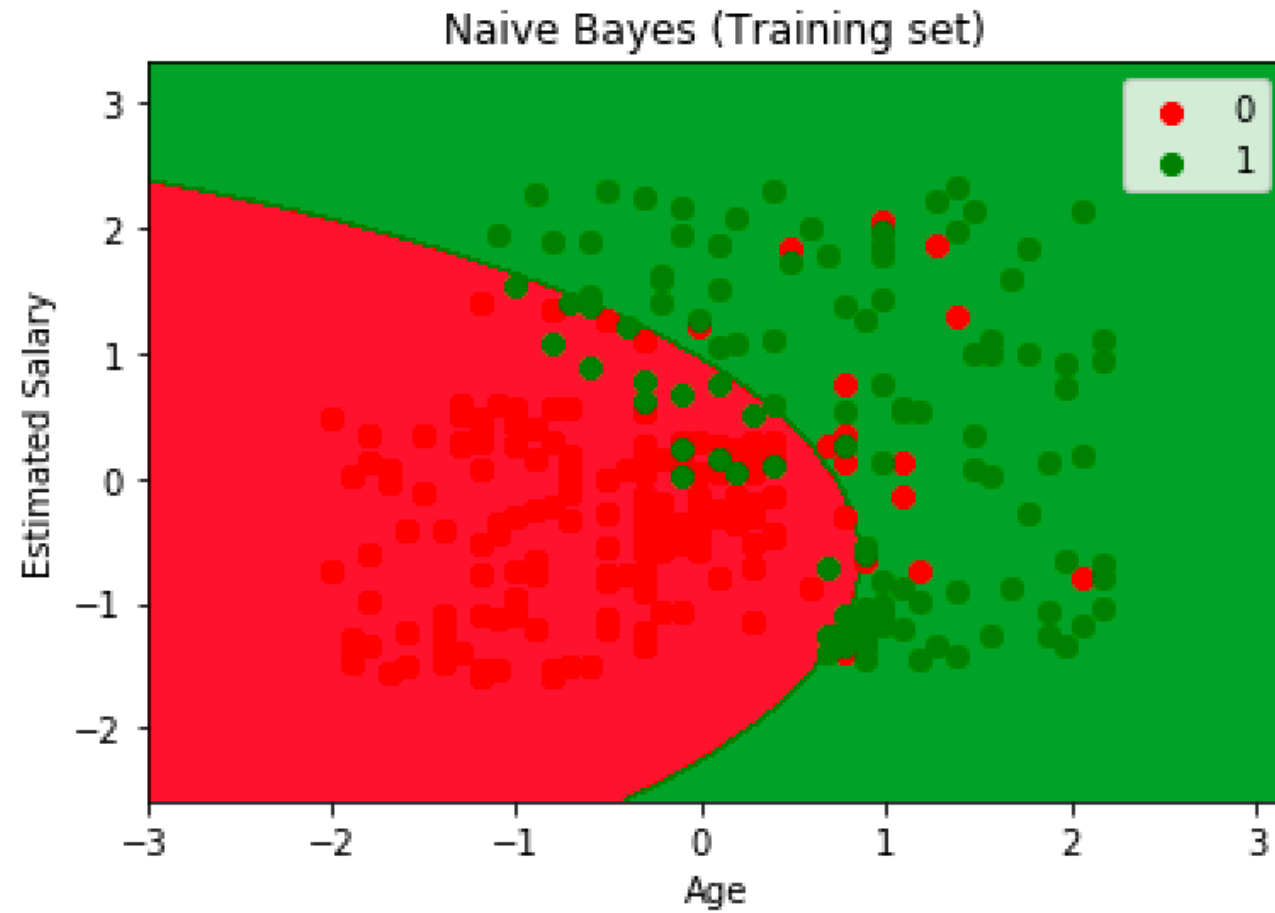
```
y_pred = classifier.predict(X_test)
```

Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(y_test, y_pred)  
print(cm)
```

Visualising the Training set results

```
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('Naive Bayes (Training set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```



Visualising the Test set results

```
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('Naive Bayes (Test set)')
plt.xlabel('Age')
plt.ylabel('Estimated Salary')
plt.legend()
plt.show()
```

