

# **UJIAN AKHIR SEMESTER**

## **Machine Learning**

Dosen Pengampu : Estiyan Dwipriyoko, S.Kom., MT.



**Disusun Oleh:**

Naji Reinaldy Amarta

41155050190045

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS LANGLANGBUANA**

**BANDUNG**

**2022**

## **KATA PENGANTAR**

Dengan menyebut nama Allah yang Maha Pengasih lagi Maha Penyayang, puji syukur kita panjatkan kehadirat Allah SWT, yang telah melimpahkan Rahmat dan Hidayah-Nya sehingga dapat menyelesaikan Tugas Ujian Akhir Semester ini.

Tugas Ujian Akhir Semester ini telah kami susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan Tugas ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan Tugas Ujian Akhir Semester ini.

Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari susunan kalimat maupun tata bahasanya. Oleh karena itu, dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami mendapatkan wawasan yang lebih untuk membuat tugas ini dikemudian hari.

Akhir kata, kami ucapkan terima kasih.

Bandung, 25 Januari 2023

Naji Reinaldy Amatha

## **BAGIAN I**

### **LANDASAN TEORI**

#### **1. Regresi Linear**

Regresi Linear adalah teknik analisis data yang memprediksi nilai data yang tidak diketahui dengan menggunakan nilai data lain yang terkait dan diketahui. Dalam Machine Learning, program komputer yang disebut algoritme menganalisis set data besar dan bekerja mundur dari data tersebut untuk menghitung persamaan regresi linier.

Model Regresi Linier relatif sederhana dan memberikan rumus matematika yang mudah ditafsirkan untuk menghasilkan prediksi. Bisnis menggunakannya untuk mengonversi data mentah secara andal dan dapat diprediksi menjadi kecerdasan bisnis serta wawasan yang dapat ditindaklanjuti.

#### **Regresi Logistik**

Regresi Logistik adalah teknik analisis data yang menggunakan matematika untuk menemukan hubungan antara dua faktor data. Kemudian menggunakan hubungan ini untuk memprediksi nilai dari salah satu faktor tersebut berdasarkan faktor yang lain. Prediksi biasanya memiliki jumlah hasil yang terbatas, seperti ya atau tidak.

Model Machine Learning yang dibangun menggunakan regresi logistik membantu organisasi mendapatkan wawasan yang dapat ditindaklanjuti dari data bisnis mereka. Mereka dapat menggunakan wawasan ini untuk analisis prediktif untuk mengurangi biaya operasional, meningkatkan efisiensi, dan menskalakan dengan lebih cepat.

## **2. Support Vector Machine**

Support Vector Machine atau *SVM* adalah model pembelajaran terawasi dengan algoritma pembelajaran terkait yang menganalisis data untuk klasifikasi dan analisis regresi. Adapun definisi lain yang dikembangkan oleh AT&T Bell Laboratories menyebut SVM adalah salah satu metode prediksi yang paling kuat, yang didasarkan pada kerangka belajar statistik.

## **3. K-Nearest Neighbours**

KNN atau K-Nearest Neighbours adalah model machine learning yang dapat digunakan untuk melakukan prediksi berdasarkan kedekatan karakteristik dengan sejumlah tetangga terdekat. Prediksi yang dilakukan dapat diterapkan baik pada classification maupun regression task.

## **4. Naive Bayes**

Naïve Bayes merupakan salah satu metode pembelajaran mesin yang memanfaatkan perhitungan probabilitas dan statistik yang dikemukakan oleh ilmuwan inggris Thomas Bayes, yaitu memprediksi probabilitas pada masa depan berdasarkan pengalaman pada masa sebelumnya. Adapun penjelasan dari Bayes Theorem yang menawarkan suatu formula untuk menghitung nilai probability dari suatu event dengan memanfaatkan pengetahuan sebelumnya dari kondisi terkait, atau sering kali dikenal dengan istilah conditional probability.

## **5. Decision Tree**

Decision Tree atau Pohon Keputusan adalah alat pendukung keputusan yang menggunakan model keputusan seperti pohon dan kemungkinan konsekuensinya. Decision Tree ini adalah salah satu cara untuk menampilkan algoritma yang hanya berisi pernyataan kontrol bersyarat. Decision Tree umumnya digunakan untuk riset operasi, khususnya dalam analisis keputusan, untuk membantu mengidentifikasi strategi yang paling mungkin mencapai tujuan.

## **6. Random Forest**

Random Forest atau Random Decision Forest (Hutan Keputusan Acak) adalah metode pembelajaran ansambel untuk klasifikasi, regresi dan tugas lain yang beroperasi dengan membangun banyak pohon keputusan pada waktu pelatihan. Random Forest umumnya mengungguli Decision Tree (Pohon Keputusan), tetapi akurasinya lebih rendah daripada Decision Tree.

## **7. K-Means**

K-means Merupakan salah satu algoritma yang bersifat unsupervised learning. K-means memiliki fungsi untuk mengelompokan data kedalam data cluster. Algoritma ini dapat menerima data tanpa ada label kategori.

## **8. Agglomerate Clustering**

Agglomerate Clustering adalah strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu cluster yang terpisah kemudian membentuk cluster yang semakin membesar.

## **9. Apriori Algorithm**

Algoritma Apriori adalah salah satu algoritma pada data mining untuk mencari frequent item/itemset pada transaksional database. Algoritma Apriori diperkenalkan untuk mencari frequent tertinggi dari suatu database. Algoritma ini banyak digunakan pada data transaksi atau biasa disebut market basket, seperti sebuah swalayan memiliki market basket yang membantu pemilik swalayan dapat mengetahui pola pembelian seorang konsumen.

## **10. Self Organizing Map**

SOM atau singkatan dari Self Organizing Map merupakan suatu jenis artificial neural network yang dilatih dengan metode unsupervised learning yang mampu menghasilkan sebuah representasi terpisah atas ruang input sampel pelatihan dengan dimensi rendah (dua dimensi). Representasi

tersebut kemudian disebut sebagai “Map”. SOM juga merupakan metode untuk melakukan pengurangan dimensi pada sampel yang dilatih.

## BAGIAN II

### STUDI KASUS

#### 1. Pemodelan data Liga120192021.csv

	A	B	C	D	E	F	G	H	I	J	K
1	Tanding	Pass1	Pass2	Pass3	Pass4	Pass5	Pass6	Pass7	Pass8	Pass9	Pass10
2	0	11	24	2	20	10	11	13	11	16	71
3	1	10	11	13	11	20	12	13	20	77	71
4	2	16	8	16	17	21	22	3	20	10	13
5	3	22	16	8	16	2	17	23	8	82	4
6	4	20	12	16	8	16	17	21	23	22	13
7	5	20	11	22	17	21	8	77	13	91	10
8	6	16	17	21	22	21	93	91	11	3	10
9	7	20	11	22	78	2	77	8	17	23	71
10	8	24	11	3	20	12	93	91	10	8	13
11	9	11	22	78	2	77	20	12	93	4	13
12	10	22	16	2	17	23	8	78	2	12	10
13	11	22	21	93	91	11	12	93	91	13	9
14	12	21	8	77	13	91	10	20	93	77	16
15	13	8	16	17	21	22	24	11	3	10	77
16	14	16	11	12	93	91	93	77	16	13	9
17	15	1	9	20	24	9	10	93	77	22	10
18	16	8	22	16	2	17	23	11	22	10	77
19	17	2	77	20	12	3	20	12	10	12	93
20	18	17	23	8	93	20	13	11	23	77	2
21	19	12	3	20	12	22	78	2	10	9	77
22	20	1	2	1	9	20	24	9	10	20	10
23	21	24	11	3	20	12	13	20	13	11	10
24	22	24	11	20	11	22	78	2	77	8	13
25	23	22	16	8	16	17	21	22	21	93	23
26	24	22	8	22	16	2	17	23	8	93	91
27	25	21	22	24	11	3	8	10	23	91	21
28	26	93	91	93	77	16	2	77	20	12	13
29	27	24	9	10	93	77	17	23	8	93	13
30	28	2	17	23	11	22	12	3	20	12	16
31	29	12	3	20	12	10	16	8	16	17	20
32	30	11	22	12	3	20	20	12	10	16	13
33	31	77	16	2	77	20	12	12	10	16	21
34	32	93	77	17	23	8	93	12	16	3	10
35	33	2	77	20	12	12	10	16	10	3	20
36	34	20	11	22	78	11	22	12	8	13	10
37	35	9	10	93	77	17	20	13	11	91	10
38	36	17	23	8	93	10	23	91	10	93	13
39	37	20	12	13	20	13	11	22	21	93	23

2. Metode yang digunakan yaitu metode Logistik Regression yang untuk mengetahui variabel kunci yang memiliki pengaruh terhadap suatu variabel bergantung, pemodelan serta estimasi atau peramalan.

Import Dataset *Liga120192021.csv* yang telah dibuat dengan syntax :

```
import pandas as pd
import numpy as np
from scipy import stats
from sklearn.linear_model import
LogisticRegression
from sklearn.model_selection import
train_test_split
from sklearn.cluster import KMeans
from sklearn.metrics import confusion_matrix
```

```
In [2]: import pandas as pd
import numpy as np
from scipy import stats
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.metrics import confusion_matrix
```

Load Dataset Liga120192021.csv

### Load Dataset Liga120192021.csv

```
In [8]: df = pd.read_csv(r'F:\Kuliah\SEMESTER 7\Machine Learning\UAS\Liga120192021.csv')
```

```
In [12]: df
```

Out[12]:

	Tanding	Pass1	Pass2	Pass3	Pass4	Pass5	Pass6	Pass7	Pass8	Pass9	Pass10
0	0	11	24	2	20	10	11	13	11	16	71
1	1	10	11	13	11	20	12	13	20	77	71
2	2	16	8	16	17	21	22	3	20	10	13
3	3	22	16	8	16	2	17	23	8	82	4
4	4	20	12	16	8	16	17	21	23	22	13
...	...	...	...	...	...	...	...	...	...	...	...
98	98	53	77	10	66	10	55	66	55	11	10
99	99	30	22	23	22	74	23	12	23	13	7
100	100	25	27	74	93	27	11	93	74	27	25
101	101	27	7	27	7	25	12	27	13	21	7
102	102	13	11	23	2	23	12	11	13	21	25

103 rows x 11 columns

Mencari Modus dari dataset *Liga120192021.csv* yang telah diimport dengan syntax :

```
from scipy import stats
df = pd.read_csv('Liga120192021.csv')
x = stats.mode(df)
print(x)
```

```
In [8]: from scipy import stats
df = pd.read_csv('Liga120192021.csv')
x = stats.mode(df, keepdims=True)
print(x)

ModeResult(mode=array([[11, 11, 2, 2, 20, 12, 12, 10, 21, 10]], dtype=int64), count=array([[ 8, 13, 10, 12, 8, 12, 12, 10, 10, 16]]))
```



### 3. Membuat Model Perkolom

Pada klasifikasi logistic regression ini saya menggunakan data kolom pass4 terhadap kolom pass5, dan kolom pass8 terhadap kolom pass9

### Klasifikasi Data Pada Kolom Pass4 terhadap Pass5

```
In [16]: models = {}
for col in df.columns[1:]:
    x = df['Pass4'].values.reshape(-1, 1)
    y = df['Pass5'].values
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.40, random_state=48)
    model = LogisticRegression(max_iter=5000).fit(x_train, y_train)
    models[col] = model
    print("Accuracy for", col, ":", model.score(x_test, y_test))
```

```
Accuracy for Pass1 : 0.023809523809523808
Accuracy for Pass2 : 0.023809523809523808
Accuracy for Pass3 : 0.023809523809523808
Accuracy for Pass4 : 0.023809523809523808
Accuracy for Pass5 : 0.023809523809523808
Accuracy for Pass6 : 0.023809523809523808
Accuracy for Pass7 : 0.023809523809523808
Accuracy for Pass8 : 0.023809523809523808
Accuracy for Pass9 : 0.023809523809523808
Accuracy for Pass10 : 0.023809523809523808
```

### Klasifikasi Data Pada Kolom Pass8 terhadap Pass9

```
In [18]: models = {}
         for col in df.columns[1:]:
             x = df['Pass8'].values.reshape(-1, 1)
             y = df['Pass9'].values
             x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.40, random_state=48)
             model = LogisticRegression(max_iter=5000).fit(x_train, y_train)
             models[col] = model
             print("Accuracy for", col, ":", model.score(x_test, y_test))
```

```
Accuracy for Pass1 : 0.023809523809523808
Accuracy for Pass2 : 0.023809523809523808
Accuracy for Pass3 : 0.023809523809523808
Accuracy for Pass4 : 0.023809523809523808
Accuracy for Pass5 : 0.023809523809523808
Accuracy for Pass6 : 0.023809523809523808
Accuracy for Pass7 : 0.023809523809523808
Accuracy for Pass8 : 0.023809523809523808
Accuracy for Pass9 : 0.023809523809523808
Accuracy for Pass10 : 0.023809523809523808
```

### Confusion Matrix Pada Pass4 dan Pass8

### Confusion Matrix Pada Pass4

```
In [19]: x = df.iloc[:, 1:].values
x = df['Pass4'].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=40, random_state=48)
model = LogisticRegression(max_iter=10000).fit(x_train, y_train)
y_pred = model.predict(x_test)
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

[illegible]

### Confusion Matrix Pada Pass8

```
In [20]: x = df.iloc[:, 1:].values
y = df['Passo'].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=40, random_state=48)
model = LogisticRegression(max_iter=10000).fit(x_train, y_train)
y_pred = model.predict(x_test)
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
```

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
[0 0 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]
[0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0]
[0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0]
[0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 1 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]

## Clustering 2 Model

## Clustering 2 Model Kolom

```
In [21]: x = df.iloc[:, 1:].values
kmeans = KMeans(n_clusters=2, random_state=0).fit(x)
print("\nCluster Center:")
print(kmeans.cluster_centers_)
```

```
Cluster Center:
[[21.62820513 23.97435897 23.23076923 22.71794872 24.15384615 14.94871795
  25.33333333 19.62820513 34.12820513 27.35897436]
 [28.72      24.08      22.16      38.16      19.92      80.44
  35.16      43.68      16.2       25.56      ]]
```