

UAS

Machine Learning



Dosen Pengampu:

ESTIYAN DWIPPRIYOKO, S.KOM., MT

Disusun Oleh:

Nur Kholis Aprianto

41155050190009

Teknik Informatika B

UNIVERSITAS LANGLANGBUANA

2022/2023

BAGIAN 1

1. Apa itu *Linear dan Logistic Regresion* dan apa gunanya?

Jawab : Regresi linear adalah teknik analisis data yang memprediksi nilai data yang tidak diketahui dengan menggunakan nilai data lain yang terkait dan diketahui. Secara matematis memodelkan variabel yang tidak diketahui atau tergantung dan variabel yang dikenal atau independen sebagai persamaan linier. Misalnya, anggaplah Anda memiliki data tentang pengeluaran dan pendapatan Anda untuk tahun lalu. Teknik regresi linier menganalisis data ini dan menentukan bahwa pengeluaran Anda adalah setengah dari penghasilan Anda. Mereka kemudian menghitung biaya masa depan yang tidak diketahui dengan mengurangi separuh pendapatan yang diketahui di masa depan.

Regresi logistik adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut dengan istilah *Ordinary Least Squares (OLS) regression*. Perbedaannya adalah pada regresi logistik, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk atau Tinggi dan Rendah.

2. Apa itu *Support Vector Machine* dan apa gunanya?

Jawab : *Support Vector Machine* adalah salah satu algoritma yang termasuk dalam kategori *Supervised Learning*, yang artinya data yang digunakan untuk belajar oleh mesin merupakan data yang memiliki label sebelumnya. Algoritma ini digunakan untuk klasifikasi dan regresi.

3. Apa itu K-Nearest Neighbor dan apa gunanya?

Jawab : Algoritma K-Nearest Neighbor adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut, K-Nearest Neighbor digunakan untuk mengambil keputusan.

4. Apa itu *Naive Bayes* dan apa gunanya?

Jawab : *Naive Bayes* merupakan metode pengklasifikasian berdasarkan probabilitas sederhana dan dirancang agar dapat dipergunakan dengan asumsi antar variabel penjelas saling bebas. Naive Bayes digunakan untuk masalah klasifikasi.

5. Apa itu Decision Tree dan apa gunanya?

Jawab : Decision Tree adalah algoritma machine learning yang menggunakan seperangkat aturan untuk membuat keputusan dengan struktur seperti pohon yang memodelkan kemungkinan hasil, biaya sumber daya, utilitas, dan kemungkinan konsekuensi ataupun resiko. Decision Tree biasanya digunakan untuk mengambil suatu keputusan.

6. Apa itu *Random Forest* dan apa gunanya?

Jawab : *Random Forest* merupakan salah satu metode dalam Decision Tree yang merupakan kombinasi dari masing-masing tree yang baik kemudian dikombinasikan ke

dalam suatu model. Random Forest ini biasanya digunakan untuk pengklasifikasian data set dalam jumlah besar.

7. Apa itu *K-Means* dan apa gunanya?

Jawab : *K-Means* adalah algoritma yang membutuhkan parameter input sebanyak k dan membagi sekumpulan n objek kedalam k cluster sehingga tingkat kemiripan antar anggota dalam satu cluster tinggi, sedangkan tingkat kemiripan dengan anggota pada cluster lain sangat rendah. Algoritma *K-Means* biasanya digunakan untuk clustering dokumen.

8. Apa itu *Agglomerate Clustering* dan apa gunanya?

Jawab : *Agglomerative Clustering* adalah metode pengelompokan data yang dimulai dengan setiap satu pengamatan sebagai clusternya sendiri kemudian terus mengelompokkan pengamatan ke dalam kelompok yang semakin besar. *Agglomerative cluster* digunakan untuk mengelompokkan kumpulan data.

9. Apa itu *Apriori Algorithm* dan apa gunanya?

Jawab : Algoritma *Apriori* adalah proses ekstraksi informasi dari suatu database, dilanjutkan dengan melakukan frequent item/itemset dan candidate generation dalam pembentukan asosiasi rule mining guna mendapatkan hasil nilai minimum support dan hasil nilai minimum confidence. Algoritma ini biasanya digunakan untuk menentukan pola dengan frekuensi tinggi.

10. Apa itu *Self Organizing Map* dan apa gunanya?

Jawab : *Self-organizing maps* merupakan suatu jenis artificial neural network yang dilatih dengan metode unsupervised learning. SOM digunakan untuk membagi pola masukan kedalam beberapa kelompok cluster.

BAGIAN 2

Syntax

1. Library

```
In [1]: #Load Library
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler
import seaborn as sns
import warnings
from scipy import stats
warnings.filterwarnings('ignore')
```

2. Load Dataset

```
In [2]: #Load dataset
df = pd.read_csv('Liga120192021.csv')
print(df.to_string())
```

```
In [4]: df.head()
```

Out[4]:

	Pass1	Pass2	Pass3	Pass4	Pass5	Pass6	Pass7	Pass8	Pass9	Pass10
0	11	24	2	20	10	11	13	11	16	71
1	10	11	13	11	20	12	13	20	77	71
2	16	8	16	17	21	22	3	20	10	13
3	22	16	8	16	2	17	23	8	82	4
4	20	12	16	8	16	17	21	23	22	13

```
In [5]: df.tail()
```

Out[5]:

	Pass1	Pass2	Pass3	Pass4	Pass5	Pass6	Pass7	Pass8	Pass9	Pass10
98	53	77	10	66	10	55	66	55	11	10
99	30	22	23	22	74	23	12	23	13	7
100	25	27	74	93	27	11	93	74	27	25
101	27	7	27	7	25	12	27	13	21	7
102	13	11	23	2	23	12	11	13	21	25

3. Membuat Model KMeans

```
In [3]: #membuat function untuk membuat model kmeans
def createModelBy2Column(index):

    #ambil 2column berdasarkan index
    new_df = df[['Pass{0}'.format(index), 'Pass{0}'.format(index+1)]]
    scaler = StandarScaler()
    scaler.fit(new_df)
    df_scaled = scaler.transform(new_df)
    df_scaled = pd.DataFrame(df_scaled)

    #memprediksi menggunakan K-Means
    km = KMeans(n_clusters=2)
    y_predicted = km.fit_predict(df_scaled)

    #atur ulang column
    new_df.loc[:, "cluster"] = y_predicted
    new_df.loc[:, "Perpindahan"] = 'Pemain {0} - Pemain{1}'.format(index, index+1)
    new_df.loc[:, "Passer"] = new_df['Pass{0}'.format(index)]
    new_df.loc[:, "Receiver"] = new_df['Pass{0}'.format(index+1)]
    new_df.drop(['Pass{0}'.format(index), 'Pass{0}'.format(index+1)], axis=1)
    return new_df
```

4. Looping Prediksi 2 kolom dan Scatter Plot

```
results = None

#menggabungkan hasil prediksi
for key in range(len(df.columns)-1):
    index = key + 1
    result = createModelBy2Column(index)
    if result is None:
        result = result
    else:
        results = pd.concat([results, result])

#menampilkan scatter plot
g = sns.FacetGrid(result, col="Perpindahan", hue = "Cluster", height=5, col_wrap=3,)
g.map(sns.scatterplot, "Passer", "Receiver")
g.add_legend()
```

Output :



