UJIAN AKHIR SEMESTER

MACHINE LEARNING

Ade Wahyu (41155050190051)

Teknik Informatika B

UNIVERSITAS LANGLANGBUANA

Jl. Karapitan No. 116, Cikawao, Lengkong, Kota Bandung

Tlp. (022) 4218084

2023

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

   Jawaban: 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?
   Jawaban: Support Vector Machine atau SVM adalah algoritme pembelajaran mesin yang diawasi yang dapat digunakan untuk klasifikasi dan regresi. Cara kerja SVM didasarkan pada SRM atau Structural Risk Minimization yang dirancang untuk mengolah data menjadi Hyperplane yang mengklasifikasikan ruang input menjadi dua kelas. Teori SVM diawali dengan pengelompokan kasus-kasus linier yang dapat dipisahkan dengan hyperplane dan dibagi menurut kelasnya.

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

   Jawaban: K-nearest neighbors atau knn adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari k tetangga terdekatnya (*nearest neighbors*). Dengan k merupakan banyaknya tetangga terdekat.

   K-nearest neighbors melakukan klasifikasi dengan proyeksi data pembelajaran pada ruang berdimensi banyak. Ruang ini dibagi menjadi bagian-bagian yang merepresentasikan kriteria data pembelajaran. Setiap data pembelajaran direpresentasikan menjadi titik-titik ***c*** pada ruang dimensi banyak.

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

   Jawaban: Naive Bayes adalah algoritma machine learning untuk masalah klasifikasi. Ini didasarkan pada teorema probabilitas Bayes. Hal ini digunakan untuk klasifikasi teks yang melibatkan set data pelatihan dimensi tinggi. Beberapa contohnya adalah penyaringan spam, analisis sentimental, dan klasifikasi artikel berita.

   Algoritma Naive Bayes disebut "naif" karena membuat asumsi bahwa kemunculan fitur tertentu tidak tergantung pada kemunculan fitur lainnya.

Misalnya, jika kalian mencoba mengidentifikasi buah berdasarkan warna, bentuk, dan rasanya, maka buah berwarna oranye, bulat, dan tajam kemungkinan besar adalah jeruk. Bahkan jika ciri-ciri ini bergantung satu sama lain atau pada keberadaan ciri-ciri lain, semua sifat ini secara individual berkontribusi pada kemungkinan bahwa buah ini adalah jeruk dan itulah sebabnya buah ini dikenal sebagai "naif."

Adapun bagian "Bayes", mengacu pada ahli statistik dan filsuf, Thomas Bayes dan teorema yang dinamai menurut namanya, teorema Bayes, yang merupakan dasar untuk Algoritma Naive Bayes.

5. Apa itu *Decision Tree* dan apa gunanya?

   Jawaban: 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 atau resiko. Konsepnya adalah dengan cara menyajikan algoritma dengan pernyataan bersyarat, yang meliputi cabang untuk mewakili langkah-langkah pengambilan keputusan yang dapat mengarah pada hasil yang menguntungkan.

   Berikut ini adalah alasan menggunakan algoritma decision tree;

   1) Decision tree biasanya meniru kemampuan berpikir manusia saat membuat keputusan, sehingga mudah dimengerti;

   2) Logika dibalik decision tree dapat dengan mudah dipahami karena menunjukkan struktur seperti pohon.

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

   Jawaban: Algoritma Random Forest disebut sebagai salah satu algoritma machine learning terbaik, sama seperti Naïve Bayes dan Neural Network. Random Forest adalah kumpulan dari decision tree atau pohon keputusan. Algoritma ini merupakan kombinasi masing-masing tree dari decision tree yang kemudian digabungkan menjadi satu model. Biasanya, Random Forest dipakai untuk masalah regresi dan klasifikasi dengan kumpulan data yang berukuran besar. Random Forest adalah algoritma dalam machine learning yang digunakan untuk pengklasifikasian data set dalam jumlah besar. Karena fungsinya bisa digunakan untuk banyak dimensi dengan berbagai skala dan performa yang tinggi. Klasifikasi ini dilakukan melalui penggabungan tree dalam decision tree dengan cara training dataset . menggunakan decision tree atau pohon keputusan untuk melangsungkan proses seleksi, di mana tree atau pohon decision tree akan dibagi secara rekursif berdasarkan data pada kelas yang sama. Dalam hal ini, penggunaan tree yang semakin banyak akan memengaruhi akurasi yang didapat menjadi lebih optimal. Penentuan klasifikasi dengan Random Forest dilakukan berdasarkan hasil voting dan tree yang terbentuk. Random Forest adalah algoritma untuk pengklasifikasian. Random Forest bekerja dengan membangun beberapa decision tree dan menggabungkannya demi mendapatkan prediksi yang lebih stabil dan akurat.

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

   Jawaban: K-means merupakan salah satu algoritma yang bersifat unsupervised learning. K-Means memiliki fungsi untuk mengelompokkan data kedalam data cluster. Algoritma ini dapat menerima data tanpa ada label kategori. K-Means Clustering Algoritma juga merupakan metode non-hierarchy. Metode Clustering Algoritma adalah mengelompokkan beberapa data ke dalam kelompok yang menjelaskan data dalam satu kelompok memiliki karakteristik yang sama dan memiliki karakteristik yang berbeda dengan data yang ada di kelompok lain. Clustering Algoritma (K-Means) memiliki tujuan untuk meminimalisasikan fungsi objective yang telah di set dalam proses clustering. Tujuan tersebut dilakukan dengan cara meminimalikan variasi data yang ada didalam cluster dan memaksimalikan variasi data yang ada di cluster lainnya.

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

   Jawaban: Agglomerate Clustering atau biasa disebut Algoritma AHC (Agglomerate Hierarchical Clustering) merupakan metode analisis kelompok data, dalam strategi pengelompokan umumnya ada dua jenis Agglomerate (Bottom-Up) dan Devisive (Top-Down). Untuk penggunaannya kita bisa menggunakan matrik jarak antar data (bisa menggunakan Euclidean atau Manhattan Disatance), lalu menggabungkan dua kelompok terdekat menjadi satu kelompok data (bisa menggunakan Single Linkage, Complete Linkage, Average Lingkage), lalu memperbaharui matrik antar data untuk merepresentasikan antara kelompok baru dengan kelompok yang masih tersisa, lalu mengulang kembali memilih jarak dan memperbaruinya sampai hanya tersisa satu kelompok.

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

   Jawaban: Algoritma apriori merupakan algoritma yang banyak digunakan pada asosiasi. Asosiasi sendiri dikenal sebagai Market Basket Anlysis atau Association Rule yang merupakan hubungan (asosiasi) antara kombinasi beberapa item (barang, orang, produk, atau apapun yang di awali dengan kata benda) yang sering muncul secara bersamaan.

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

    Jawaban: *Self-organizing maps* (SOM) adalah salah satu jenis *artificial neural network* atau ANN. Jaringan ini dilatih dengan metode *unsupervised learning* atau tanpa arahan dari data input-target. Jika dibandingkan dengan ANN lainnya, SOM cukup berbeda. *Self-organizing maps* (SOM) merupakan suatu jenis *artificial neural network* yang dilatih dengan metode *unsupervised learning*. Jaringan ini mampu menghasilkan sebuah representasi terpisah atas ruang *input* sampel pelatihan dengan dimensi rendah (biasanya dua dimensi). Representasi tersebut kemudian disebut sebagai "*map*". SOM juga merupakan metode untuk melakukan pengurangan dimensi pada sampel yang dilatih. Gagasan mengenai SOM pertama kali dicetuskan oleh Teuvo Kohonen, seorang peneliti di bidang Ilmu Komputer. Kohonen menciptakan SOM berbeda dari ANN jenis lainnya. Sebab, SOM menerapkan metode pembelajaran kompetitif alih-alih pembelajaran koreksi kesalahan. Jaringan ini juga menerapkan fungsi *neighbourhood* untuk melestarikan sifat topologi dari ruang *input*.

Jaringan SOM terdiri dari dua lapisan penting: *input* dan *output* (*map feature*). Tahap awal SOM dimulai dengan inisialisasi bobot ke vektor. Selanjutnya, beberapa vektor dipilih sebagai sampel secara acak. Vektor yang telah dipetakan kemudian dicari, tujuannya untuk mengetahui mana bobot yang paling mewakili vektor *input*. *Self-organizing maps* (SOM) mampu mempertahankan informasi struktural dari data pelatihan. Selain itu, SOM juga menghasilkan data yang tidak linier secara inheren

Bagian 2

Coding :

1. Load Library

```
# 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

```
[2]:
# Load Dataset
df = pd.read_csv("/kaggle/input/persib-20202021/Liga120192021.csv")
df
```

[2]:

	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
...	...	...	...	...	...	...	...	...	...	...
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

103 rows × 10 columns

3. Membuat function untuk membuat model KMeans

```
# Membuat function untuk membuat model KMeans
def createModelBy2Column(index):
    #Mengambil 2 column berdasarkan index
    new_df = df[['Pass{0}'.format(index), 'Pass{0}'.format(index+1)]]
    scaler = StandardScaler()
    scaler.fit(new_df)
    df_scaled = scaler.transform(new_df)
    df_scaled = pd.DataFrame(df_scaled)

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

    # Mengatur ulang Columns
    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 per-2 kolom dan menampilkan *Scatter Plot*

[4]:

```
results = None

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

# Menampilkan Scatter Plot
g = sns.FacetGrid(results, col="Perpindahan", hue = "Cluster", height=5, col_wrap=3,)
g.map(sns.scatterplot, "Passer", "Receiver")
g.add_legend()
```

Hasil :

```
[4]: <seaborn.axisgrid.FacetGrid at 0x7f721b1d4190>
```