MACHINE LEARNING

UJIAN AKHIR SEMESTER

MUHAMAD TAOFIK SETIAWAN

41155050190002

TEKNIK INFORMATIKA – B

FAKULTAS TEKNIK

PRODI TEKNIK INFORMATIKA

UNIVERSITAS LANGLANGBUANA

2023

Bagian 1 (40%) Jawab Pertanyaan berikut:

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

Analisis Regresi : digunakan untuk menganalisis hubungan dan pengaruh antara variabel bebas/independen dan variabel dependen/terikat.

a) Linear Regresion

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 kita mempunyai data tentang pengeluaran dan pendapatan kita tahun lalu. Teknik regresi linier menganalisis data ini dan menentukan bahwa pengeluaran kita setengah dari penghasilan kita. Kemudian menghitung biaya masa depan yang tidak diketahui dengan mengurangi separuh pendapatan yang diketahui di masa depan.

b) Logistik Regresion

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

Misalnya, kita ingin menebak apakah pengunjung di suatu situs web akan mengeklik tombol checkout di keranjang belanja mereka atau tidak. Analisis regresi logistik melihat perilaku pengunjung di masa lalu, seperti waktu yang dihabiskan di situs web dan jumlah item di keranjang. Analisis regresi logistik menentukan bahwa, di masa lalu, jika pengunjung menghabiskan lebih dari lima menit di situs web dan menambahkan lebih dari tiga item ke keranjang, pengunjung akan mengeklik tombol checkout. Dengan menggunakan informasi ini, fungsi regresi logistik dapat memprediksi perilaku pengunjung baru di situs web.

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

Support Vector Machine termasuk metode Superviced Machine Learning yang menggunakan algoritma klasifikasi untuk masalah klasifikasi yang memisahkan 2 kelas. Prinsip dasar SVM adalah linear classifier, dan selanjutnya dikembangkan agar bisa bekerja pada non-liniar dengan konsep kernel trick pada ruang berdimensi tinggi.

Dalam SVM, setiap item/data diplot sebagai titik dalam ruang n-dimensi, dengan nilai setiap fitur menjadi nilai kordinat tertentu. Kemudian dilakukan klasifikasi dengan mencari *hyperplane* yang membedakan kedua kelas dengan baik.

Tujuan dari SVM adalah menemukan hyperplane dalam ruang n-dimensi yang secara jelas mengklasifikasikan titik data.

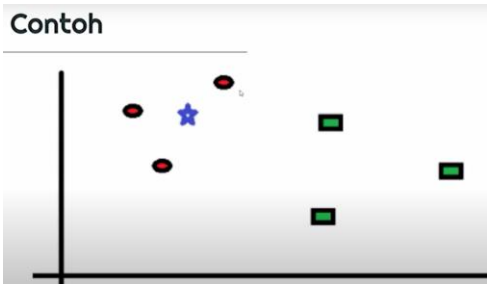
Tujuan hyperplane adalah untuk menentukan bidang yang memiliki margin maksimum yaitu jarak maksimum antara titik data dari kedua kelas. Memaksimalkan jarak margin memberikan beberapa penguatan sehingga titik data uji dapat diklasifikasikan dengan baik.

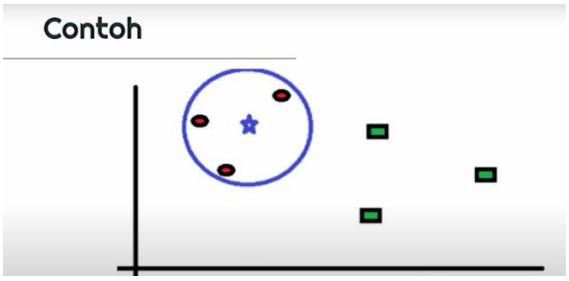
3. Apa itu *K-Nearest Neighbor* dan apa gunanya?

K-Nearest Neighbor menangkap gagasan kesamaan (jarak/kedekatan) dengan menghitung jarak antar titik pada grafik.

Algoritma K-Nearest Neighbor mengasumsikan bahwa hal serupa ada dalam jarak dekat atau dengan kata lain hal-hal yang serupa dekat satu sama lain.

Algoritma K-Nearest Neighbor adalah algoritma *supervised learning* dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori terdekat. Tujuan dari algoritma ini adalah untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel-sampel dari training data. Algoritma K-Nearest Neighbor menggunakan *Neighborhood Classification* sebagai nilai prediksi dari nilai instance yang baru.





Jadi, semisal kita punya instance baru berbentuk bintang, lalu kita ingin mengklasifikasi bintang tersebut masuk ke kelas lingkaran merah atau kotak hijau. Untuk mengetahuinya kita cukup menentukan berapa tetangga yang akan kita jadikan titik patokan, semilai kita akan menentukan 3 tetangga terdekat, jadi bisa kita tarik lurus dari bintang tersebut ke titik titik tetangga yang terdekatnya, maka nanti hasilnya bintang biru terklasifikasi di kelas lingkaran merah.

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

Naïve Bayes merupakan sebuah metoda klasifikasi menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan inggris Thomas Bayes. Algoritma naïve Bayes memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. Ciri utama dari Naïve Bayes Classifier ini adalah asumsi yang sangat kuat akan independensi dari masing-masing kondisi/kejadian.

Naïve Bayes Classifier bekerja sangat baik dibanding dengan model classifier lainnya. Hal ini dibuktikan pada jurnal Xhemali, Daniela, Chris J. Hinde, and Roger G. Stone. "Naïve Bayes vs Decision Trees vs Neural network in the classification of traininng page."(2009). Bahwa "Naïve Bayes Classifier memiliki tingga akurasi yang lebih baik dibanding model classifier lainnya.

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

Decision Tree adalah teknik pembelajaran Suprevised yang dapat didunakan untuk masalah klasifikasi dan regresi, tetapi lebih umum digunakan untuk menyelesaikan masalah klasifikasi. Disebut Decision Tree karena mirip dengan pohon, dimulai dari simpul akar, yang diperluas dengan cabang dan daun sampai membangun struktur seperti pohon.

Classifier berbentuk tree dimana :

- Simpul internal mewakili fitur dari kumpulan data
- Cabang mewakili aturan keputusan

- Setiap simpul daun mewakili hasilnya.

Dalam sebuah pohon keputusan terdapat dua node yaitu :

- **Node Keputusan / Decision Node** digunakan untuk membuat keputusan apa dan memiliki banyak cabang.
- **Simpul Daun / Leaf Node** merupakan output dari keputusan tersebut dan tidak mengandung cabng lebih lanjut.

Decision Tree mudah dipahami karena mengikuti proses yang sama dengan cara manusia saat membuat keputusan dalam kehidupan nyata, dapat memecahkan masalah terkait keputusan dengan baik karena membantu untuk memikirkan semua kemungkinan hasil dari suatu masalah dan juga lebih sedikit persyaratan untuk data cleaning dibandingkan dengan algoritma lain. Selain itu decisin tree pun memiliki beberapa hambatan seperti jika mengandung banyak lapisan atau layer yang membuatnya rumit, memungkinkan overfitting serta jika memililiki label kelas yang banyak maka kompleksitas komputasi dapat meningkat.

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

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?**

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?**

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?**

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. Biasanya banyak digunakan di toko-toko untuk mengatur penempatan barang atau

mengontrol penjualan, misal orang beli rokok maka juga beli koreknya, sehingga rokok dan korek memiliki nilai asosiasi.

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

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 (60%) Studi Kasus: Membuat model dengan *machine learning* pada data Liga120192021.csv

1. Buatlah model dengan machine learning pada data *Liga120192021.csv* !
2. Untuk metode silakan PILIH SALAH SATU: *Logistic Regression, Support Vector Machine, K-Nearest Neigbour, K Means*. Sebut metode yang dipilih!
3. Buat model perkolom, klasifikasi = 2, *cluster* = 2.
4. Buat juga hasil pemetaan warna dengan scattered plot !

5. Hasilnya selain dikumpulkan berupa ***pdf***, harus disimpan di *Github* masing-masing. **Berikan URL** *Github* masing-masing!

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>
```

Link GitHub : taostwn36/UAS_ML_MuhamadTaofikSetiawan (github.com)