

**UAS
MACHINE LEARNING**



MOCHAMAD AKBAR RAHMAN

41155050190023

A1

SEMESTER 7

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS LANGLANGBUANA

BANDUNG

2023

BAGIAN 1

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

Linear Regression adalah metode statistik yang digunakan untuk memprediksi hubungan antara satu atau lebih variabel independen (x) dan satu variabel dependen (y). Model linear dibentuk dengan menemukan garis yang paling baik yang mewakili hubungan antara variabel independen dan dependen.

Logistic Regression, juga disebut sebagai regresi logistik, adalah metode statistik yang digunakan untuk memprediksi probabilitas terjadinya suatu kejadian berdasarkan satu atau lebih variabel independen. Model logistik dibentuk dengan menemukan garis yang paling baik yang mewakili hubungan antara variabel independen dan probabilitas kejadian.

Kedua metode digunakan dalam berbagai bidang, seperti analisis pasar, ilmu kedokteran, ilmu sosial, dll. Linear Regression digunakan untuk memprediksi hubungan antara variabel numerik, sementara Logistic Regression digunakan untuk memprediksi hubungan antara variabel kategori.

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

Support Vector Machine (SVM) adalah algoritma klasifikasi yang digunakan untuk memisahkan dua kelas dari data dengan menemukan garis atau hyperplane yang paling baik yang memisahkan dua kelas tersebut. SVM mencari hyperplane yang memiliki jarak terdekat ke setiap titik data dari kedua kelas, yang disebut sebagai "vektor dukungan".

SVM juga dapat digunakan untuk klasifikasi multi-kelas dengan membuat beberapa model SVM untuk setiap pasangan kelas dan memilih kelas dengan skor probabilitas tertinggi.

SVM juga dapat digunakan dalam konteks regresi dengan mencari hyperplane yang meminimalkan rata-rata kuadrat dari jarak antara titik data dan hyperplane.

SVM banyak digunakan pada aplikasi yang melibatkan pengenalan pola, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks. SVM juga digunakan dalam bidang seperti bioinformatika, komputasi visi, dan sistem rekomendasi.

Keunggulan dari SVM adalah mampu menangani data dengan jumlah fitur yang besar dan dapat menangani data yang tidak linear dengan menambahkan fitur melalui kernel trick.

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

K-Nearest Neighbor (K-NN) adalah algoritma klasifikasi yang digunakan untuk menentukan kelas dari suatu data baru dengan mencari k data terdekat dalam dataset

yang sudah diketahui kelasnya dan memilih kelas yang paling banyak muncul di k data terdekat tersebut. K adalah jumlah tetangga terdekat yang digunakan untuk menentukan kelas data baru.

K-NN digunakan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks. K-NN juga digunakan dalam bidang seperti bioinformatika, komputasi visi, dan sistem rekomendasi. Keunggulan dari K-NN adalah mudah digunakan dan interpretasinya, tidak memerlukan proses pelatihan seperti algoritma supervised lainnya, dan mampu menangani data yang tidak linear. Namun, K-NN memerlukan penyimpanan data latih yang besar dan memerlukan waktu untuk mencari tetangga terdekat ketika digunakan untuk mengklasifikasikan data baru.

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

Naive Bayes adalah algoritma klasifikasi yang digunakan untuk menentukan kelas dari suatu data baru dengan mengasumsikan bahwa semua fitur independen dari data adalah independen satu sama lain. Algoritma ini berdasarkan teorema Bayes dan digunakan untuk menentukan probabilitas kelas dari data baru berdasarkan probabilitas kelas dari data latih dan probabilitas fitur dari data latih.

Naive Bayes digunakan dalam berbagai bidang, seperti pengenalan teks, pengenalan suara, dan klasifikasi email spam. Naive Bayes juga digunakan dalam bidang seperti bioinformatika, ilmu kedokteran, dan analisis pasar. Keunggulan dari Naive Bayes adalah mudah digunakan dan interpretasinya, tidak memerlukan banyak data latih, dan mampu menangani data yang tidak linear. Namun, asumsi independensi fitur dari Naive Bayes seringkali tidak sesuai dengan kenyataan sehingga dapat menurunkan performa dari algoritma.

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

Decision Tree adalah algoritma klasifikasi yang digunakan untuk menentukan kelas dari suatu data baru dengan membuat pohon keputusan yang mengeksplorasi fitur-fitur dari data. Setiap node dari pohon keputusan mewakili suatu fitur, dan setiap cabang dari node mewakili nilai dari fitur tersebut. Kelas dari data baru ditentukan dengan mengikuti cabang dari pohon keputusan yang sesuai dengan nilai-nilai dari fitur data baru.

Decision Tree digunakan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks. Decision Tree juga digunakan dalam bidang seperti bioinformatika, komputasi visi, dan sistem rekomendasi. Keunggulan dari Decision Tree adalah mudah digunakan dan interpretasinya, tidak memerlukan banyak data latih, dan mampu menangani data yang tidak linear. Namun, pohon keputusan dapat menjadi sangat kompleks dan tidak stabil, sehingga dapat menurunkan performa dari algoritma.

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

Random Forest adalah algoritma klasifikasi yang digunakan untuk menentukan kelas dari suatu data baru dengan membuat sejumlah pohon keputusan yang dibangun secara acak dari data latih. Kelas dari data baru ditentukan dengan mengambil mayoritas dari kelas yang ditentukan oleh pohon-pohon keputusan tersebut.

Random Forest digunakan dalam berbagai bidang, seperti pengenalan wajah, pengenalan suara, dan pengenalan teks. Random Forest juga digunakan dalam bidang seperti bioinformatika, komputasi visi, dan sistem rekomendasi. Keunggulan dari Random Forest adalah mampu menangani overfitting yang sering terjadi pada pohon keputusan tunggal, mampu menangani data yang tidak linear, dan meningkatkan akurasi dari algoritma.

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

K-Means adalah algoritma pengelompokan (clustering) yang digunakan untuk mengelompokkan data menjadi k kelompok yang berbeda (k disebut sebagai jumlah kelompok) berdasarkan kesamaan fitur. Algoritma ini mengejar kondisi di mana titik data dalam satu kelompok sama dekatnya dengan centroid dari kelompok tersebut dan jauh dari centroid dari kelompok lain.

K-Means digunakan dalam berbagai bidang, seperti pembelajaran mesin, data mining, dan analisis data. contohnya dalam bidang marketing, K-means dapat digunakan untuk mengelompokkan pelanggan berdasarkan pengeluaran, sehingga perusahaan dapat mengambil keputusan marketing yang tepat untuk setiap kelompok. Keunggulan dari K-Means adalah mudah digunakan dan interpretasinya, dan mampu menangani data yang tidak linear. Namun, algoritma ini sensitif terhadap pemilihan nilai k dan posisi awal centroid sehingga bisa saja menghasilkan hasil yang berbeda pada setiap eksekusi.

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

Agglomerative Clustering (atau Hierarchical Clustering) adalah algoritma pengelompokan (clustering) yang digunakan untuk mengelompokkan data menjadi hierarki dari kelompok yang berbeda. Algoritma ini memulai dengan menganggap setiap titik data sebagai satu kelompok, kemudian menggabungkan dua kelompok yang paling mirip menjadi satu kelompok yang lebih besar, dan seterusnya hingga semua titik data tergabung dalam satu kelompok besar atau sejumlah kelompok yang ditentukan.

Agglomerative Clustering digunakan dalam berbagai bidang, seperti pembelajaran mesin, data mining, dan analisis data. Contohnya dalam bidang genetika, Agglomerative Clustering dapat digunakan untuk mengelompokkan gen yang memiliki aktivitas yang sama. Keunggulan dari Agglomerative Clustering adalah dapat menghasilkan hierarki dari kelompok yang menunjukkan relasi antar kelompok, serta dapat menangani data yang tidak linear. Namun, algoritma ini memerlukan waktu yang cukup lama untuk mengeksekusi dan memerlukan pemilihan metrik yang tepat untuk menentukan kesamaan antar kelompok.

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

Apriori Algorithm adalah algoritma yang digunakan untuk menemukan aturan asosiasi pada data transaksi. Algoritma ini menemukan itemset yang sering muncul bersama-sama (frequent itemset) dan kemudian digunakan untuk menemukan aturan asosiasi yang memiliki tingkat kepercayaan yang cukup tinggi. Apriori Algorithm mengimplementasikan teknik "Apriori Property" yang mengeliminasi itemset yang tidak memenuhi syarat dari data transaksi.

Apriori Algorithm digunakan dalam berbagai bidang, seperti data mining, analisis data, dan business intelligence. Contohnya dalam bidang e-commerce, Apriori Algorithm dapat digunakan untuk menemukan aturan asosiasi antara produk yang sering dibeli bersama sehingga dapat digunakan untuk meningkatkan penjualan. Keunggulan dari Apriori Algorithm adalah mudah digunakan dan interpretasinya, serta mampu menemukan aturan asosiasi yang cukup berguna. Namun, algoritma ini memerlukan waktu yang cukup lama untuk mengeksekusi dan memerlukan pemilihan nilai minimum support yang tepat.

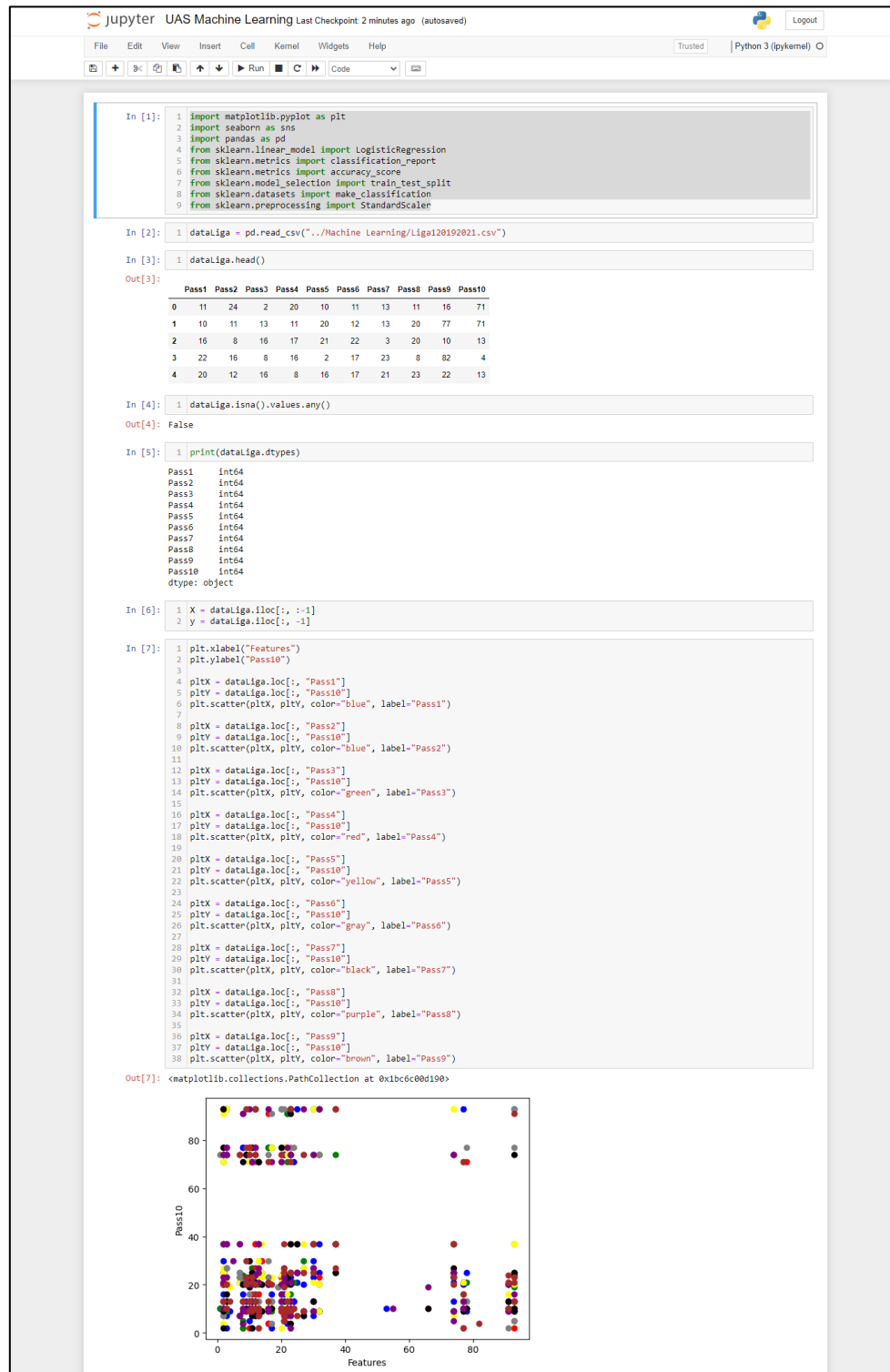
10. Apa itu *Self Organizing Map* dan apa gunanya?

Self Organizing Map (SOM) adalah algoritma pengelompokan (clustering) yang digunakan untuk mengelompokkan data menjadi sebuah peta yang diatur secara otomatis. Peta ini terdiri dari sejumlah neuron yang dikoneksikan satu sama lain dan mewakili data yang dikelompokkan. Setiap neuron di peta mewakili satu kelompok data dan berinteraksi dengan neuron lainnya untuk membentuk pola yang mencerminkan kesamaan fitur dari data.

SOM digunakan dalam berbagai bidang, seperti pembelajaran mesin, data mining, dan analisis data. Contohnya dalam bidang komputasi visi, SOM dapat digunakan untuk mengelompokkan citra berdasarkan kesamaan fitur, sehingga dapat digunakan untuk pengenalan objek. Keunggulan dari SOM adalah dapat menangani data yang tidak linear dan mampu menghasilkan visualisasi yang mudah dipahami. Namun, algoritma ini memerlukan waktu yang cukup lama untuk mengeksekusi dan memerlukan pemilihan jumlah neuron yang tepat.

BAGIAN 2

Metode Logistic Regression




```

In [8]: 1 X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)

In [9]: 1 model = LogisticRegression()
2 model.fit(X_train, y_train)

C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_1 = _check_optimize_result(
Out[9]: LogisticRegression
LogisticRegression()

In [10]: 1 predictions = model.predict(X_test)
2 print(predictions)

[93 74  9 30 30 71 21 20 24  9 71 30 74 10 13  9 37  9 7 37  9 5 20  9
13  9]

In [11]: 1 print(classification_report(y_test, predictions))
2 print("accuracy: ", accuracy_score(y_test, predictions))

              precision    recall  f1-score   support

     2       0.00      0.00      0.00        1.0
     5       0.00      0.00      0.00        0.0
     7       0.00      0.00      0.00        1.0
     9       0.00      0.00      0.00        2.0
    10       0.00      0.00      0.00        7.0
    13       0.00      0.00      0.00        4.0
    16       0.00      0.00      0.00        1.0
    19       0.00      0.00      0.00        1.0
    20       0.00      0.00      0.00        2.0
    21       0.00      0.00      0.00        2.0
    24       0.00      0.00      0.00        0.0
    25       0.00      0.00      0.00        1.0
    30       0.00      0.00      0.00        0.0
    37       0.00      0.00      0.00        0.0
    71       0.00      0.00      0.00        1.0
    74       0.00      0.00      0.00        2.0
    93       0.00      0.00      0.00        1.0

 accuracy          0.00        26.0
macro avg          0.00      0.00      0.00        26.0
weighted avg       0.00      0.00      0.00        26.0

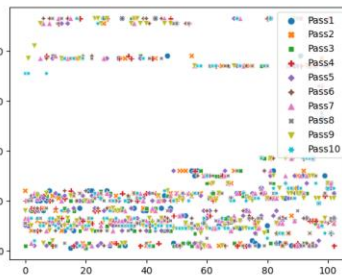
accuracy:  0.0

C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))
C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use 'zero_division' parameter to control this behavior.
warn_prf(average, modifier, msg_start, len(result))

In [12]: 1 # Scatter Plot

In [13]: 1 sns.scatterplot(data = data10a)
2 plt.show()

```



Link GitHub : <https://github.com/akbarrahman112/UASMachineLearning>