



JAWABAN UJIAN AKHIR SEMESTER (UAS)
SEMESTER GANJIL TAHUN AKADEMIK 2022/2023

Mata Kuliah : Machine Learning
Nama : Rifky Dzalbarry
Npm : 41155050190034
Kelas : INF-A
Semester : VII

Bagian I

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

Linear regression adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel dependen dan satu atau lebih variabel independen dengan menyesuaikan persamaan linear dengan data yang diamati. Persamaan tersebut memiliki bentuk $Y = a + bX$, di mana Y adalah variabel dependen, X adalah variabel independen, a adalah titik potong, dan b adalah kemiringan garis. Tujuan dari linear regression adalah untuk menemukan garis yang paling cocok melalui titik data. Linear regression dapat digunakan untuk analisis regresi sederhana maupun regresi berganda.

Logistic Regression adalah metode statistik yang digunakan untuk menganalisis dataset di mana ada satu atau lebih variabel independen yang menentukan hasil. Hasil diukur dengan variabel dikotom (di mana hanya ada dua kemungkinan hasil). Logistic Regression digunakan

untuk memprediksi hasil biner (1/0, Ya/Tidak, Benar/Salah) diberikan sekumpulan variabel independen. Logistic Regression adalah variasi dari Linear Regression, di mana hasil diprediksi menggunakan probabilitas bukan nilai Y. Logistic Regression menggunakan persamaan sebagai representasi, sangat mirip dengan Linear Regression, namun matematika inti berbeda.

2. Apa itu Support Vector Machine dan apa gunanya?

Support Vector Machine (SVM) adalah sebuah model pembelajaran mesin yang digunakan untuk klasifikasi, regresi, dan deteksi anomali. SVM mencari garis atau hyperplane pemisah terbaik untuk memisahkan data ke dalam kelas yang berbeda. Gunanya adalah membuat prediksi dengan akurasi yang baik pada data yang tidak linier dan memiliki banyak fitur.

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

K-Nearest Neighbor (K-NN) adalah sebuah algoritma klasifikasi yang berdasarkan pada asumsi bahwa data yang serupa berdekatan di ruang feature. Dalam K-NN, setiap titik data diklasifikasikan ke kelas yang sama dengan mayoritas dari K titik terdekatnya. Gunanya adalah membuat prediksi dengan cepat pada data kategori dan untuk mengatasi masalah tidak linear.

4. Apa itu Naive Bayes dan apa gunanya?

Naive Bayes adalah sebuah algoritma klasifikasi probabilistik yang berdasarkan teorema Bayes. Algoritma ini mengasumsikan bahwa setiap fitur dalam data adalah independen satu sama lain dan membuat prediksi berdasarkan probabilitas dari data training. Ada beberapa varian dari Naive Bayes, seperti Gaussian Naive Bayes, Multinomial Naive Bayes, dan Bernoulli Naive Bayes. Gunanya adalah membuat prediksi dengan cepat dan akurat pada data kategori dan text classification.

5. Apa itu Decision Tree dan apa gunanya?

Decision Tree adalah sebuah model pembelajaran mesin yang berdasarkan pada pembuatan pohon keputusan. Pada setiap simpul, model membuat keputusan berdasarkan fitur tertentu dan membagi data ke dalam simpul-simpul berikutnya hingga data terklasifikasi ke dalam kelas akhir. Gunanya adalah membuat model yang mudah dipahami dan visualisasi, serta membuat prediksi pada data dengan banyak fitur dan kompleksitas tidak linier.

6. Apa itu Random Forest dan apa gunanya?

Random Forest adalah sebuah algoritma pembelajaran mesin yang menggabungkan beberapa pohon keputusan (Decision Tree) menjadi satu model. Setiap pohon keputusan dibangun secara acak dari sample data dan fitur, dan hasil akhir dari Random Forest adalah hasil dari voting dari setiap pohon. Gunanya adalah meminimalisir overfitting yang sering terjadi pada model pohon keputusan tunggal dan

membuat prediksi yang akurat pada data dengan banyak fitur dan kompleksitas tidak linier.

7. Apa itu K-Means dan apa gunanya?

K-Means adalah sebuah algoritma clustering yang membagi data menjadi K cluster berdasarkan jarak antara data dan centroid dari setiap cluster. Algoritma ini mengiterasi pemindahan centroid dan pembagian data hingga konvergensi. Gunanya adalah untuk melakukan eksplorasi dan visualisasi data, serta untuk melakukan kompresi data dengan mengurangi jumlah fitur menjadi representasi centroid dari setiap cluster.

8. Apa itu Agglomerate Clustering dan apa gunanya?

Agglomerative Clustering adalah sebuah algoritma clustering yang menggabungkan data ke dalam cluster yang lebih besar, mulai dari setiap data sebagai cluster sendiri dan akhirnya menjadi satu cluster besar. Algoritma ini menentukan jarak antar data atau cluster dan memilih pasangan terdekat untuk digabungkan hingga tidak ada lagi pasangan yang dapat digabungkan. Gunanya adalah untuk melakukan eksplorasi dan visualisasi data, dan untuk melakukan kompresi data dengan mengurangi jumlah fitur menjadi representasi cluster dari setiap hierarki.

9. Apa itu Apriori Algorithm dan apa gunanya?

Apriori Algorithm adalah sebuah algoritma untuk menemukan itemset yang sering muncul (frequent itemset) pada transaksi data. Algoritma ini menggunakan prinsip Apriori, bahwa jika sebuah itemset tidak sering muncul maka itemset lain yang terkait dengan itemset tersebut juga tidak akan sering muncul. Gunanya adalah untuk melakukan analisis market basket pada data transaksi untuk menemukan pola belanja yang sering terjadi.

10. Apa itu Self Organizing Map dan apa gunanya?

Self Organizing Map (SOM) adalah sebuah algoritma neural network untuk melakukan pemetaan data ke dalam bentuk 2D atau jaringan kartesian. Algoritma ini memodelkan data sebagai vektor dan membuat pemetaan dengan memperbaharui bobot dari jaringan sampai jarak antar vektor dan bobot menjadi minimal. Gunanya adalah untuk melakukan visualisasi dan explorasi data yang memiliki banyak fitur dan tidak linier, serta untuk melakukan kompresi data menjadi representasi pemetaan yang mudah dipahami.

Bagian II

Classification with Logistic Regression

```
In [1]: import pandas as pd
```

```
In [2]: dataLiga = pd.read_csv("../Machine-Learning/Liga120192021.csv")
```

```
In [3]: dataLiga.head()
```

```
Out[3]:
```

	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 [4]: dataLiga.tail()
```

```
Out[4]:
```

	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

```
In [5]: dataLiga.isna().values.any()
```

```
Out[5]: False
```

```
In [6]: print(dataLiga.dtypes)
```

```
Pass1    int64
Pass2    int64
Pass3    int64
Pass4    int64
Pass5    int64
Pass6    int64
Pass7    int64
Pass8    int64
Pass9    int64
Pass10   int64
dtype: object
```

```
In [8]: import matplotlib.pyplot as plt
```

```
In [9]: x = dataLiga.iloc[:, :-1]
        y = dataLiga.iloc[:, -1]
```

```
In [10]: plt.xlabel("Features")
plt.ylabel("Pass10")

pltX = dataLiga.loc[:, "Pass1"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "blue", label = "Pass1")

pltX = dataLiga.loc[:, "Pass2"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "aqua", label = "Pass2")

pltX = dataLiga.loc[:, "Pass3"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "green", label = "Pass3")

pltX = dataLiga.loc[:, "Pass4"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "red", label = "Pass4")

pltX = dataLiga.loc[:, "Pass5"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "yellow", label = "Pass5")

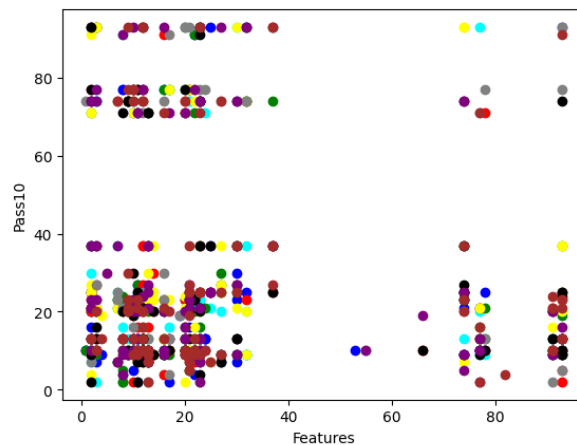
pltX = dataLiga.loc[:, "Pass6"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "gray", label = "Pass6")

pltX = dataLiga.loc[:, "Pass7"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "black", label = "Pass7")

pltX = dataLiga.loc[:, "Pass8"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "purple", label = "Pass8")

pltX = dataLiga.loc[:, "Pass9"]
pltY = dataLiga.loc[:, "Pass10"]
plt.scatter(pltX, pltY, color = "brown", label = "Pass9")
```

Out[10]: <matplotlib.collections.PathCollection at 0x295904762d0>



```
In [13]: from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
from sklearn.preprocessing import StandardScaler
```

```
In [15]: X_train, X_test, y_train, y_test = train_test_split(x, y, random_state=42)
```

```
In [17]: model = LogisticRegression()
model.fit(X_train, y_train)

C:\Users\HP\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:444: 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_i = _check_optimize_result(
```

```
Out[17]: LogisticRegression
LogisticRegression()
```

```
In [18]: predictions = model.predict(X_test)
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 [19]: print(classification_report(y_test, predictions))
print("accuracy: ", accuracy_score(y_test, predictions))
```

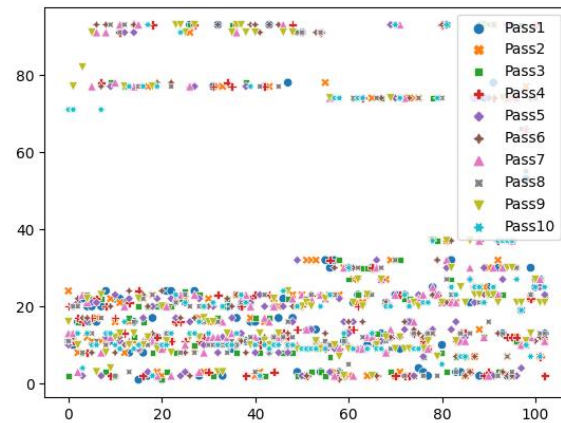
	precision	recall	f1-score	support
2	0.00	0.00	0.00	1.0
5	0.00	0.00	0.00	0.0
7	0.00	0.00	0.00	1.0
9	0.00	0.00	0.00	2.0
10	0.00	0.00	0.00	7.0
13	0.00	0.00	0.00	4.0
16	0.00	0.00	0.00	1.0
19	0.00	0.00	0.00	1.0
20	0.00	0.00	0.00	2.0
21	0.00	0.00	0.00	2.0
24	0.00	0.00	0.00	0.0
25	0.00	0.00	0.00	1.0
30	0.00	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

```
In [20]: #Scatter Plot
```

```
In [21]: import seaborn as sns
```

```
In [22]: sns.scatterplot(data = dataliga)
plt.show()
```



Clustering with K-Means

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
```

```
In [2]: dataset = pd.read_csv('../Machine-Learning/Liga120192021.csv')
```

```
In [3]: dataset.tail()
```

```
Out[3]:
```

	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

```
In [4]: dataset.keys()
```

```
Out[4]: Index(['Pass1', 'Pass2', 'Pass3', 'Pass4', 'Pass5', 'Pass6', 'Pass7', 'Pass8',
              'Pass9', 'Pass10'],
              dtype='object')
```

```
In [5]: dataku = pd.DataFrame(dataset)
dataku.head()
```

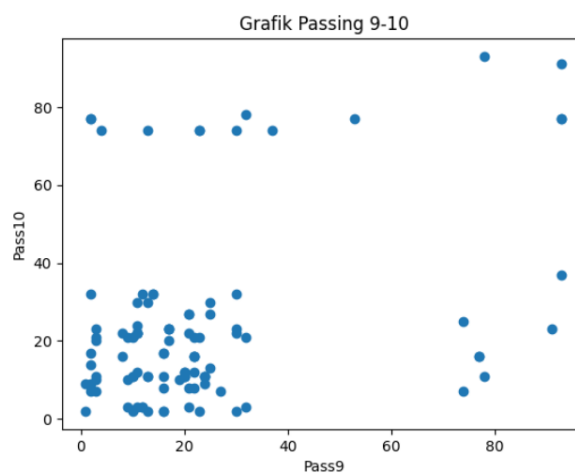
```
Out[5]:
```

	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 [6]: X = np.asarray(dataset)
print(X)

[[11 24  2 ... 11 16 71]
 [10 11 13 ... 20 77 71]
 [16  8 16 ... 20 10 13]
 ...
 [25 27 74 ... 74 27 25]
 [27  7 27 ... 13 21  7]
 [13 11 23 ... 13 21 25]]
```

```
In [7]: plt.scatter(X[:,0],X[:,1], label='True Position')
plt.xlabel("Pass9")
plt.ylabel("Pass10")
plt.title("Grafik Passing 9-10")
plt.show()
```



```
In [8]: kmeans = KMeans(n_clusters=2)
kmeans.fit(X)
```

```
Out[8]:
KMeans
KMeans(n_clusters=2)
```

```
In [9]: print(kmeans.cluster_centers_)

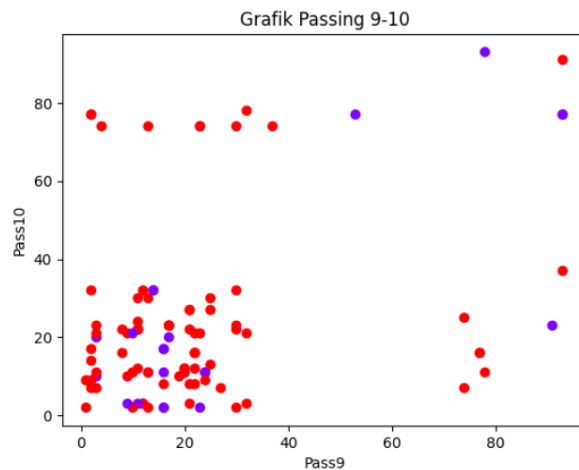
[[29.17391304 24.08695652 16.82608696 33.47826087 20.         86.43478261
 30.13043478 40.30434783 15.86956522 26.30434783]
 [21.675      23.975      24.7375     24.45        24.025      14.8625
 27.025       21.2        33.775      27.1         ]]
```

```
In [10]: print(kmeans.labels_)

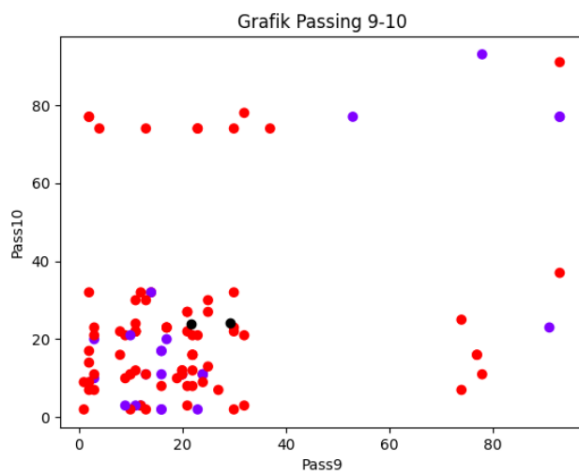
[1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
 1 1 0 0 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0
 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1]
```

```
In [11]: plt.scatter(X[:,0],X[:,1], c=kmeans.labels_, cmap='rainbow')
plt.xlabel("Pass9")
plt.ylabel("Pass10")
plt.title("Grafik Passing 9-10")
plt.show()
```

```
In [11]: plt.scatter(X[:,0],X[:,1], c=kmeans.labels_, cmap='rainbow')
plt.xlabel("Pass9")
plt.ylabel("Pass10")
plt.title("Grafik Passing 9-10")
plt.show()
```

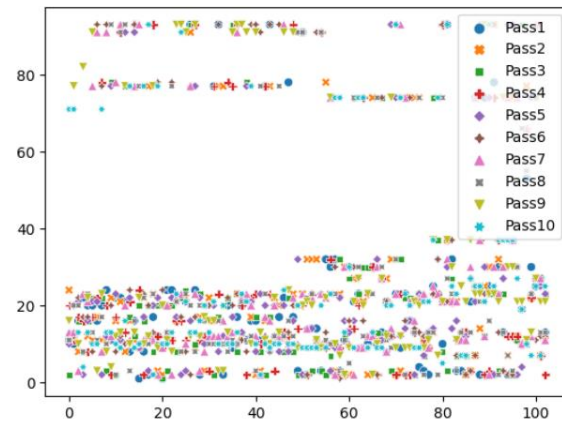


```
In [12]: plt.scatter(X[:,0],X[:,1], c=kmeans.labels_, cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[0],kmeans.cluster_centers_[1],color='black')
plt.xlabel("Pass9")
plt.ylabel("Pass10")
plt.title("Grafik Passing 9-10")
plt.show()
```



```
In [13]: import seaborn as sns
```

```
In [14]: sns.scatterplot(data = dataku)  
plt.show()
```



<https://github.com/rifkydzalbarry/Machine-Learning>