

# **Machine Learning**

## **UJIAN AKHIR SEMESTER**



**Riki Kurniawan**

**41155050190083**

**Teknik Informatika B**

**FAKULTAS TEKNIK**

**UNIVERSITAS LANGLANGBUANA**

**2023**

## Bagian 1

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

**Regresi logistik** adalah teknik analisis data yang menggunakan matematika untuk menemukan hubungan antara dua faktor data.

**Regresi linear** adalah teknik analisis data yang memprediksi nilai data yang tidak diketahui dengan menggunakan nilai data lain yang terkait dan diketahui.

Regresi linier digunakan untuk variabel dependen yang memiliki skala data kontinyu, sedangkan variabel dependen yang memiliki skala data kategori menggunakan regresi logistik. Regresi logistik digunakan untuk skala data kategori baik skala data ordinal maupun skala data nominal

2. Apa itu Support Vector Machine dan apa gunanya?

**Support Vector Machine (SVM)** merupakan salah satu metode dalam supervised learning yang biasanya digunakan untuk klasifikasi (seperti Support Vector Classification) dan regresi (Support Vector Regression).

SVM digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas. Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas.

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

**K-Nearest Neighbor (KNN)** adalah sebuah metode untuk melakukan klasifikasi terhadap objek yang berdasarkan dari data pembelajaran yang jaraknya paling dekat dengan objek tersebut.

K-Nearest Neighbor digunakan untuk menentukan nilai bobot setiap kriteria dengan mengklasifikasikan dengan baik atau buruk.

4. Apa itu Naive Bayes dan apa gunanya?

**Naive Bayes** adalah algoritma yang banyak digunakan dan sering digunakan untuk keperluan klasifikasi atau pengelompokan data.

Naive Bayes digunakan untuk mengklasifikasikan dokumen teks seperti teks berita ataupun teks akademis, sebagai metode machine learning yang menggunakan probabilitas, untuk membuat diagnosis medis secara otomatis, dan mendeteksi atau menyaring spam.

5. Apa itu Decision Tree dan apa gunanya?

**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.

Decision Tree digunakan untuk mengambil suatu keputusan. Untuk bisa mengambil keputusan dengan struktur ini, Anda perlu menyiapkan beberapa variabel yang nantinya berfungsi sebagai bahan pertimbangan keputusan yang diambil.

6. Apa itu Random Forest dan apa gunanya?

**Random Forest** adalah algoritma machine learning yang menggabungkan keluaran dari beberapa decision tree untuk mencapai satu hasil.

Random Forest membangun beberapa decision tree dan menggabungkannya untuk mendapatkan prediksi yang lebih akurat dan stabil. Satu keuntungan besar dari Random Forest adalah dapat digunakan untuk masalah klasifikasi dan regresi, yang merupakan mayoritas sistem pembelajaran mesin saat ini.

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.

8. Apa itu Agglomerative Clustering dan apa gunanya?

**Agglomerative Clustering** adalah strategi pengelompokan hirarki yang dimulai dengan setiap objek dalam satu cluster yang terpisah kemudian membentuk cluster yang semakin membesar.

9. Apa itu Apriori Algorithm dan apa gunanya?

**Algoritma Apriori** adalah algoritma yang digunakan untuk menghitung aturan asosiasi antar objek. Aturan asosiasi menjelaskan bagaimana dua atau lebih objek terkait satu sama lain.

Algoritma Apriori pertama kali diperkenalkan oleh Agrawal dan Shrikant (1994) yang berguna untuk menentukan frequent itemset pada sekumpulan data.

10. Apa itu Self Organizing Map dan apa gunanya?

**Self Organizing Map** merupakan salah satu teknik dalam Neural Network yang bertujuan untuk melakukan visualisasi data dengan cara mengurangi dimensi data melalui penggunaan self-organizing neural networks sehingga manusia dapat mengerti high-dimensional data yang dipetakan dalam bentuk low-dimensional data.

## Bagian 2

### Import pandas dan numpy dan testing

```
[9] import pandas as pd
import numpy as np

[10] df = pd.read_csv("/content/Ligal20192021.csv", encoding='latin-1')

[11] df.head()
```

	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

```
[13] df.shape

(103, 10)

[14] df.info()
```

completed at 8:59 PM

Informasi tipe data, jumlah kolom dan memastikan bahwa tidak ada data yang duplikasi

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 10 columns):
#   Column    Non-Null Count  Dtype  
---  -
0    Pass1     103 non-null    int64  
1    Pass2     103 non-null    int64  
2    Pass3     103 non-null    int64  
3    Pass4     103 non-null    int64  
4    Pass5     103 non-null    int64  
5    Pass6     103 non-null    int64  
6    Pass7     103 non-null    int64  
7    Pass8     103 non-null    int64  
8    Pass9     103 non-null    int64  
9    Pass10    103 non-null    int64  
dtypes: int64(10)
memory usage: 8.2 KB
```

```
[16] df.duplicated().sum()

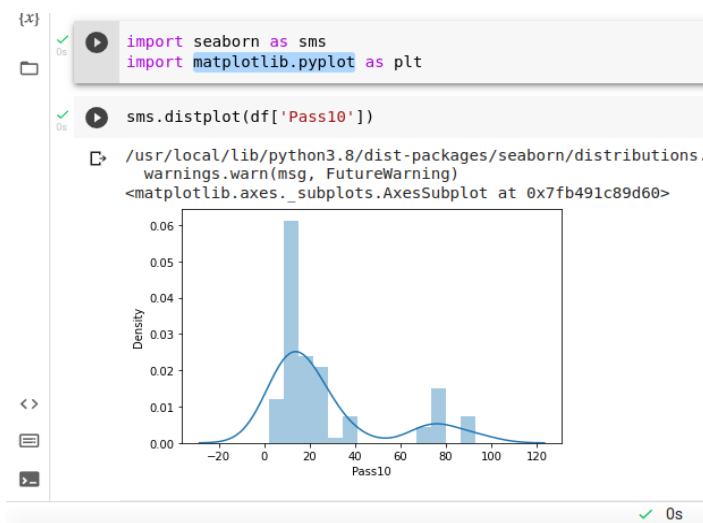
0
```

Memastikan bahwa semua data tidak ada yang kosong

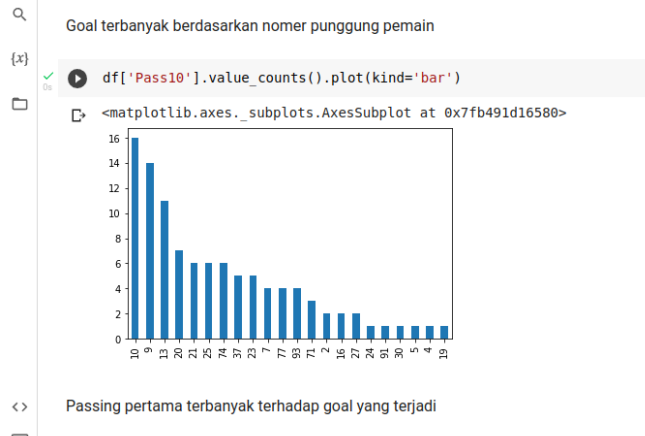
```
df.isnull().sum()
```

Pass1	0
Pass2	0
Pass3	0
Pass4	0
Pass5	0
Pass6	0
Pass7	0
Pass8	0
Pass9	0
Pass10	0
dtype: int64	

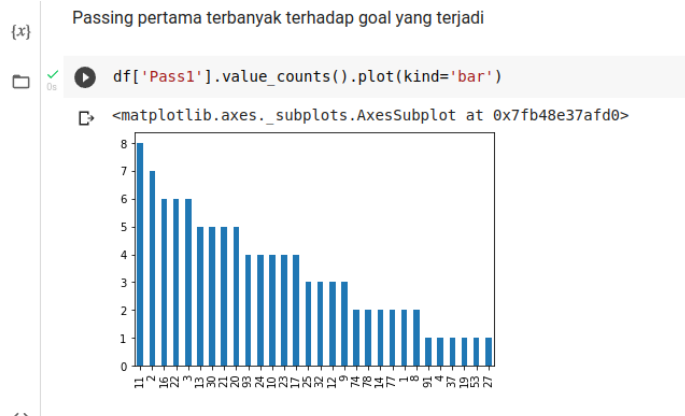
Mengimport seaborn dan matplotlib.pyplot untuk menghitung data dan mengubahnya ke dalam chart



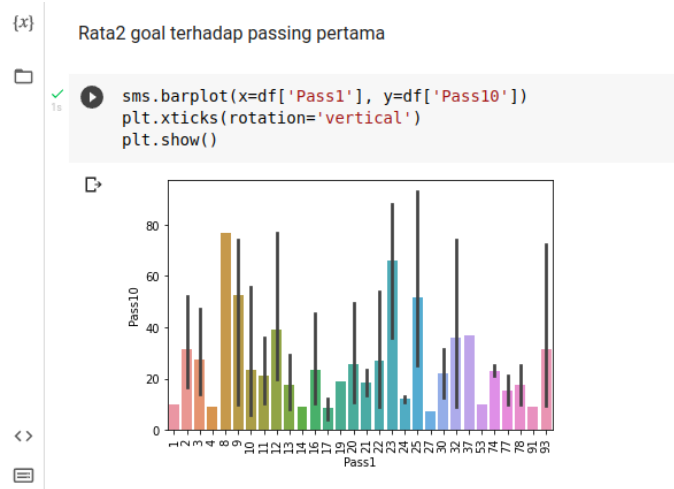
## Hasil goal terbanyak berdasarkan Pass10



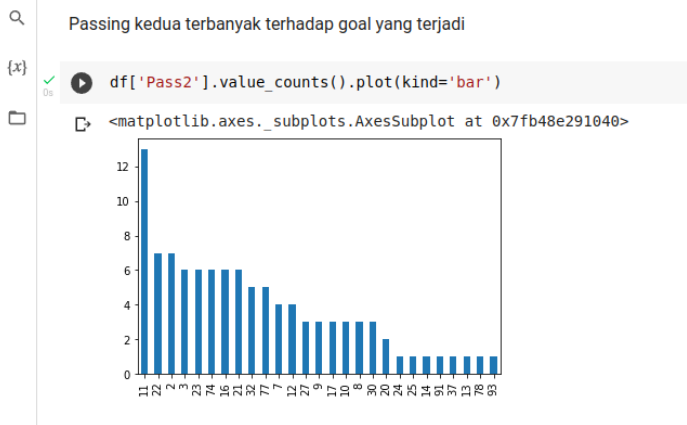
## Passing pertama terbanyak terhadap goal yang terjadi



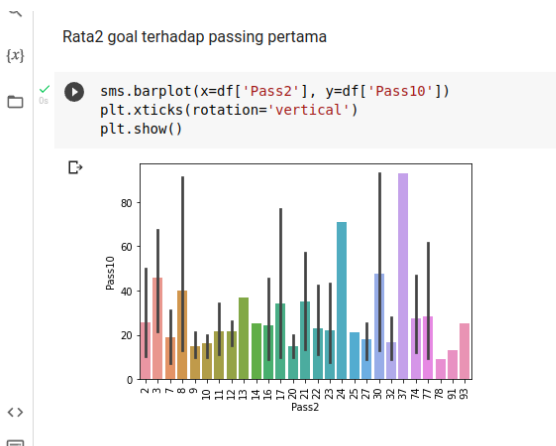
## Rata2 goal terhadap passing pertama



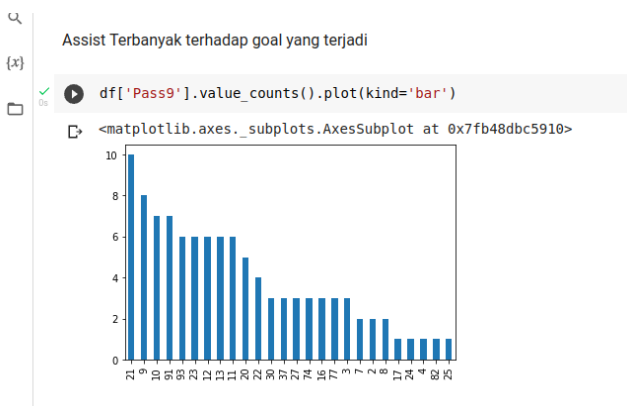
## Passing ke-2 terbanyak



Rata2 goal yang terjadi terhadap passing ke-2

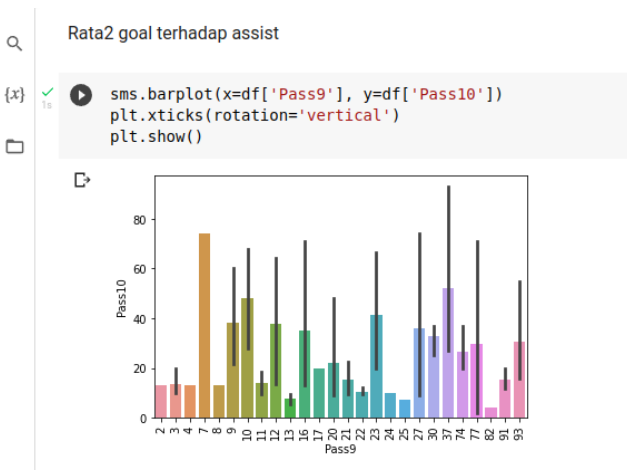


Assist(Passing ke-9) terbanyak

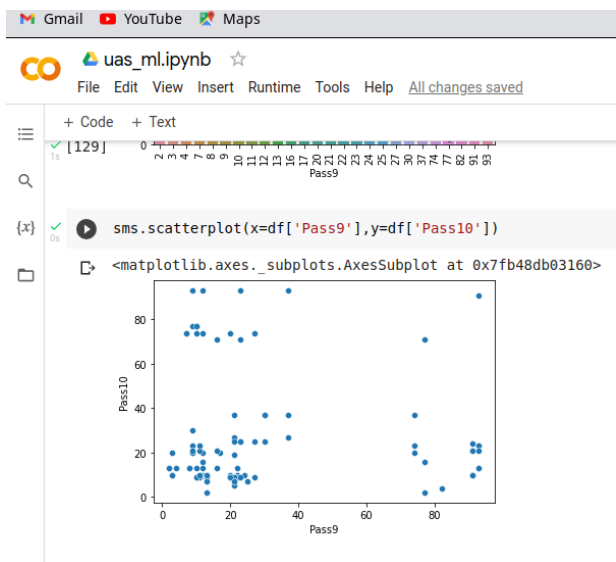




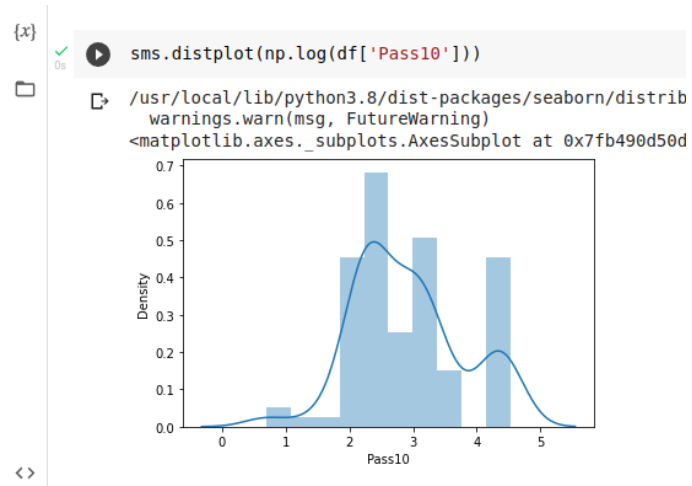
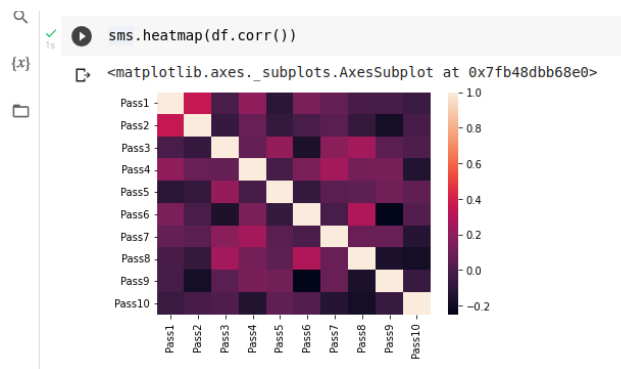
## Rata2 goal terhadap assist(Passing ke-9)



## Scatterplot assist terhadap goal yang terjadi



## Heatmap data csvnya



## Model data 9 terhadap data 10

Mendeklarasikan data\_X (Pass 9)

```
data_X = df.Pass9[:, np.newaxis]
print(data_X)
```



Mendeklarasikan data X train, X\_test, y\_train, y test kemudian membuat regresi terhadap data X train dan y train, Lalu mendeklarasikan y\_predict dan menentukan accuracy berdasarkan mean squared\_error

```
[169] data_X_train = data_X[:-30]
      data_X_test = data_X[-20:]

      data_y_train = df.Pass10[:-30]
      data_y_test = df.Pass10[-20:]

[180] from sklearn import linear_model
      reg = linear_model.LinearRegression()
      reg.fit(data_X_train, data_y_train)

      y_predict = reg.predict(data_X_test)

      from sklearn.metrics import mean_squared_error
      accuracy = mean_squared_error(data_y_test, y_predict)
      print(accuracy)

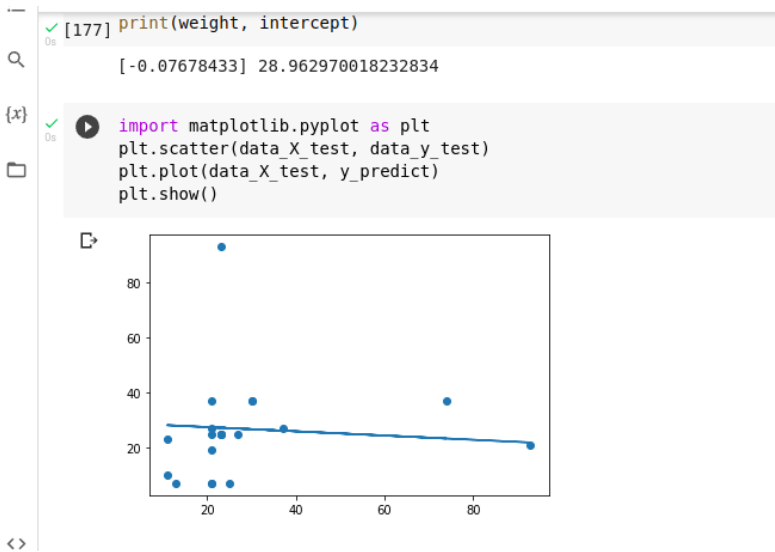
347.20578040133756

[177] weight = reg.coef_
      intercept = reg.intercept_
      print(weight, intercept)

[-0.07678433] 28.962970018232834
```



Scatterplot Assist(Pass9) terhadap Goal(Pass10) yang terjadi



Model Pass8 terhadap Pass10

```
data_X = df.Pass8[:, np.newaxis]
print(data_X)
```

```
[183] data_X_train = data_X[:-30]
      data_X_test = data_X[-20:]

      data_y_train = df.Pass10[:-30]
      data_y_test = df.Pass10[-20:]

      from sklearn import linear_model
      reg = linear_model.LinearRegression()
      reg.fit(data_X_train, data_y_train)

      y_predict = reg.predict(data_X_test)
```

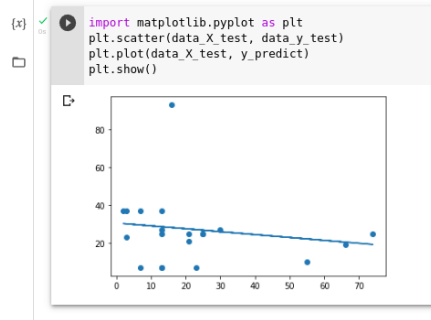
```
[185] from sklearn.metrics import mean_squared_error
      accuracy = mean_squared_error(data_y_test, y_predict)
      print(accuracy)

328.5407168847145
```

```
[186] weight = reg.coef_
      intercept = reg.intercept_
      print(weight, intercept)

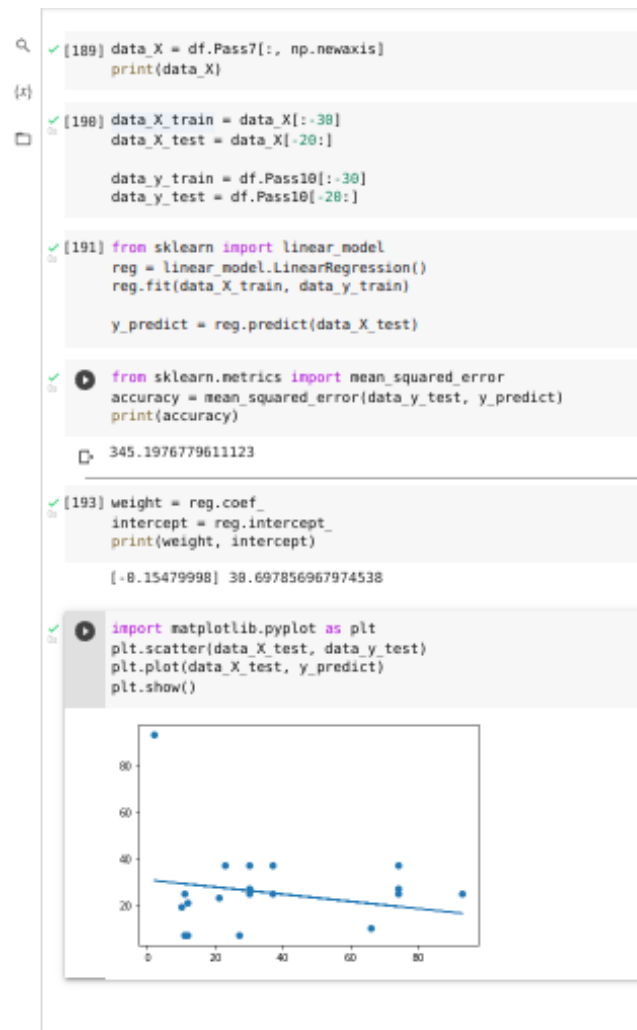
[-0.15379101] 30.590752587426625
```

## Scatterplot Pass8 terhadap Pass10



## Model Pass7 terhadap Pass10

### Scatterplot Pass7 terhadap Pass10



## Model Pass7 terhadap Pass10

### Scatterplot Pass7 terhadap Pass10

