**UTS Machine Learning**

Nama : Hurin Salimah

NIM : 1103200021

* Prompt chatGPT:  
  berikan saya kode library untuk klasifikasi menggunakan metode XGBoost
* Hasil kode chatGPT:

#import library untuk manipulasi data numerik

import numpy as np

#import library untuk manipulasi data tabular

import pandas as pd

#import library untuk visualisasi data

import seaborn as sns

#import library untuk visualisasi data

import matplotlib.pyplot as plt

#

from sklearn.model\_selection import train\_test\_split

#

from sklearn.metrics import accuracy\_score, classification\_report

#

from xgboost import XGBClassifier

* Prompt chatGPT:

berikan saya kode untuk menyambungkan google drive dengan google colab

* Hasil kode chatGPT:

#Import google drive API

from google.colab import drive

#Login ke google drive

drive.mount('/content/drive')

* Hasil running:

Mounted at /content/drive

* Penjelasan hasil running:

Google Colab berhasil terhubung dengan google drive

* Prompt chatGPT:

berikan saya kode untuk load dataset college

* Hasil kode chatGPT:

#mengambil dataset dari google drive

dataset = pd.read\_csv('/content/drive/MyDrive/machine learning/breast cancer.csv')

* Prompt chatGPT:

berikan saya kode untuk menampilkan dataset

* Hasil kode chatGPT:

#menampilan 5 baris pertama yang menggambarkan dataset yang diambil

print(dataset.head())

* Hasil running:

Unnamed: 0 mean radius mean texture mean perimeter mean area \

0 0 17.99 10.38 122.80 1001.0

1 1 20.57 17.77 132.90 1326.0

2 2 19.69 21.25 130.00 1203.0

3 3 11.42 20.38 77.58 386.1

4 4 20.29 14.34 135.10 1297.0

mean smoothness mean compactness mean concavity mean concave points \

0 0.11840 0.27760 0.3001 0.14710

1 0.08474 0.07864 0.0869 0.07017

2 0.10960 0.15990 0.1974 0.12790

3 0.14250 0.28390 0.2414 0.10520

4 0.10030 0.13280 0.1980 0.10430

mean symmetry ... worst texture worst perimeter worst area \

0 0.2419 ... 17.33 184.60 2019.0

1 0.1812 ... 23.41 158.80 1956.0

2 0.2069 ... 25.53 152.50 1709.0

3 0.2597 ... 26.50 98.87 567.7

4 0.1809 ... 16.67 152.20 1575.0

worst smoothness worst compactness worst concavity worst concave points \

0 0.1622 0.6656 0.7119 0.2654

1 0.1238 0.1866 0.2416 0.1860

2 0.1444 0.4245 0.4504 0.2430

3 0.2098 0.8663 0.6869 0.2575

4 0.1374 0.2050 0.4000 0.1625

worst symmetry worst fractal dimension target

0 0.4601 0.11890 0.0

1 0.2750 0.08902 0.0

2 0.3613 0.08758 0.0

3 0.6638 0.17300 0.0

4 0.2364 0.07678 0.0

[5 rows x 32 columns]

* Penjelasan hasil running:
* Unnamed: 0 : nomor baris dari dataset asli sebelumnya
* mean radius, mean texture, mean perimeter, mean area, mean smoothness: mewakili berbagai atribut atau fitur dari data medis yang digunakan dalam analisis kanker payudara.
* target : kolom yang menandakan label atau kelas dari data, contoh 0 untuk kanker jinak dan 1 untuk kanker ganas)
* Prompt chatGPT:

berikan saya kode untuk mengambil dataset dari library sklearn

* Hasil kode chatGPT:

#memuat dataset kanker payudara yang sudah tersedia pada sklearn

from sklearn.datasets import load\_breast\_cancer

#untuk memuat dataset kanker payudara

data = load\_breast\_cancer()

#dataset kanker payudara dimuat ke dalam dataframe pandas

X = pd.DataFrame(data.data, columns=data.feature\_names)

#diisi dengan target atau label dari dataset kanker payudara

y = pd.Series(data.target, name='target')

* Prompt chatGPT:

berikan saya kode EDA untuk klasifikasi menggunakan metode XGBoost

* Hasil kode chatGPT:
* #EDA (Exploratory Data Analysis)
* #Menampilkan informasi dasar tentang dataset

print(X.info())

#Menampilkan statistik deskriptif

print(X.describe())

* Hasil running:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 569 entries, 0 to 568

Data columns (total 30 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 mean radius 569 non-null float64

1 mean texture 569 non-null float64

2 mean perimeter 569 non-null float64

3 mean area 569 non-null float64

4 mean smoothness 569 non-null float64

5 mean compactness 569 non-null float64

6 mean concavity 569 non-null float64

7 mean concave points 569 non-null float64

8 mean symmetry 569 non-null float64

9 mean fractal dimension 569 non-null float64

10 radius error 569 non-null float64

11 texture error 569 non-null float64

12 perimeter error 569 non-null float64

13 area error 569 non-null float64

14 smoothness error 569 non-null float64

15 compactness error 569 non-null float64

16 concavity error 569 non-null float64

17 concave points error 569 non-null float64

18 symmetry error 569 non-null float64

19 fractal dimension error 569 non-null float64

20 worst radius 569 non-null float64

21 worst texture 569 non-null float64

22 worst perimeter 569 non-null float64

23 worst area 569 non-null float64

24 worst smoothness 569 non-null float64

25 worst compactness 569 non-null float64

26 worst concavity 569 non-null float64

27 worst concave points 569 non-null float64

28 worst symmetry 569 non-null float64

29 worst fractal dimension 569 non-null float64

dtypes: float64(30)

memory usage: 133.5 KB

None

mean radius mean texture mean perimeter mean area \

count 569.000000 569.000000 569.000000 569.000000

mean 14.127292 19.289649 91.969033 654.889104

std 3.524049 4.301036 24.298981 351.914129

min 6.981000 9.710000 43.790000 143.500000

25% 11.700000 16.170000 75.170000 420.300000

50% 13.370000 18.840000 86.240000 551.100000

75% 15.780000 21.800000 104.100000 782.700000

max 28.110000 39.280000 188.500000 2501.000000

mean smoothness mean compactness mean concavity mean concave points \

count 569.000000 569.000000 569.000000 569.000000

mean 0.096360 0.104341 0.088799 0.048919

std 0.014064 0.052813 0.079720 0.038803

min 0.052630 0.019380 0.000000 0.000000

25% 0.086370 0.064920 0.029560 0.020310

50% 0.095870 0.092630 0.061540 0.033500

75% 0.105300 0.130400 0.130700 0.074000

max 0.163400 0.345400 0.426800 0.201200

mean symmetry mean fractal dimension ... worst radius \

count 569.000000 569.000000 ... 569.000000

mean 0.181162 0.062798 ... 16.269190

std 0.027414 0.007060 ... 4.833242

min 0.106000 0.049960 ... 7.930000

25% 0.161900 0.057700 ... 13.010000

50% 0.179200 0.061540 ... 14.970000

75% 0.195700 0.066120 ... 18.790000

max 0.304000 0.097440 ... 36.040000

worst texture worst perimeter worst area worst smoothness \

count 569.000000 569.000000 569.000000 569.000000

mean 25.677223 107.261213 880.583128 0.132369

std 6.146258 33.602542 569.356993 0.022832

min 12.020000 50.410000 185.200000 0.071170

25% 21.080000 84.110000 515.300000 0.116600

50% 25.410000 97.660000 686.500000 0.131300

75% 29.720000 125.400000 1084.000000 0.146000

max 49.540000 251.200000 4254.000000 0.222600

worst compactness worst concavity worst concave points \

count 569.000000 569.000000 569.000000

mean 0.254265 0.272188 0.114606

std 0.157336 0.208624 0.065732

min 0.027290 0.000000 0.000000

25% 0.147200 0.114500 0.064930

50% 0.211900 0.226700 0.099930

75% 0.339100 0.382900 0.161400

max 1.058000 1.252000 0.291000

worst symmetry worst fractal dimension

count 569.000000 569.000000

mean 0.290076 0.083946

std 0.061867 0.018061

min 0.156500 0.055040

25% 0.250400 0.071460

50% 0.282200 0.080040

75% 0.317900 0.092080

max 0.663800 0.207500

[8 rows x 30 columns]

* Penjelasan hasil running:

Hasil yang ditampilkan adalah informasi tentang dataset yang memiliki 569 baris (entri) dan 30 kolom (fitur). Dataset ini terdiri dari fitur-fitur yang tampaknya numerik, masing-masing diukur dalam float64.

X.info():

* Menginformasikan struktur dataset dengan 569 entri dan 30 kolom.
* Menunjukkan bahwa tidak ada nilai yang hilang (non-null count = 569 untuk setiap kolom).
* Semua kolom berisi data numerik dengan tipe data float64.

X.describe():

* Memberikan statistik deskriptif untuk setiap kolom numerik.
* Statistik ini mencakup count (jumlah entri), mean (rata-rata), std (standar deviasi), min (nilai minimum), quartiles (25%, 50%, 75%), dan max (nilai maksimum) dari setiap kolom numerik.
* Dari contoh yang disediakan, setiap fitur numerik memiliki rentang nilai yang berbeda-beda, seperti mean radius, mean texture, dan lainnya.

* Prompt chatGPT :

berikan saya kode menampilkan distribusi target dan korelasi antar fitur

* Hasil kode chatGPT :

#Menampilkan plot yang menunjukkan distribusi dari kelas target

sns.countplot(y)

#Menampilkan plot yang telah dibuat sebelumnya

plt.show()

#Korelasi antar fitur numerik

correlation\_matrix = X.corr()

#Memanggil heatmap dari matriks korelasi yang telah dihitung

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

#Menampilkan plot yang telah dibuat sebelumnya

plt.show()

* Hasil running :

A blue square with white text

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Prompt chatGPT :

berikan saya kode untuk menampilkan visualisasi fitur

* Hasil kode chatGPT :

# Visualisasi fitur utama

num\_features = min(len(X.columns), 10)

# Menyiapkan area gambar untuk viasualisasi

plt.figure(figsize=(20, 10))

#Melakukan iterasi melalui fitur dari dataset

for i, feature in enumerate(X.columns, 1):

#Membuat subplot dengan 2 baris dan 5 kolom

plt.subplot(2, 5, i)

#membuat diagram kotak untuk setiap fitur dalam kelas target

sns.boxplot(x=y, y=feature, data=X)

#Menampilkan plot yang telah dibuat sebelumnya

plt.show()

* Hasil running :

A group of blue and orange boxes

Description automatically generated

* Prompt chatGPT:

berikan saya kode untuk splitting data pada klasifikasi meggunakan xgboost

* Hasil kode chatGPT:

# Splitting data menjadi data latih dan data uji

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

* Prompt chatGPT:

berikan saya kode untuk splitting data pada klasifikasi meggunakan xgboost

* Hasil kode chatGPT:

# Training model XGBoost untuk klasifikasi

model = XGBClassifier()

#Melakukan prediksi pada dataset

model.fit(X\_train, y\_train)

* Hasil running:

XGBClassifier

XGBClassifier(base\_score=None, booster=None, callbacks=None,

colsample\_bylevel=None, colsample\_bynode=None,

colsample\_bytree=None, device=None, early\_stopping\_rounds=None,

enable\_categorical=False, eval\_metric=None, feature\_types=None,

gamma=None, grow\_policy=None, importance\_type=None,

interaction\_constraints=None, learning\_rate=None, max\_bin=None,

max\_cat\_threshold=None, max\_cat\_to\_onehot=None,

max\_delta\_step=None, max\_depth=None, max\_leaves=None,

min\_child\_weight=None, missing=nan, monotone\_constraints=None,

multi\_strategy=None, n\_estimators=None, n\_jobs=None,

num\_parallel\_tree=None, random\_state=None, ...)

* Penjelasan hasil running:

setiap parameter memiliki pengaruh sendiri terhadap bagaimana model XGBoost akan belajar dari dataset dan cara model tersebut akan memprediksi. Menggunakan parameter-parameter ini dengan bijak dapat mempengaruhi kinerja dan perilaku model XGBoost saat digunakan dalam tugas klasifikasi. Menyesuaikan parameter-parameter ini seringkali merupakan bagian penting dari proses tuning model untuk mendapatkan hasil yang optimal.

* Prompt chatGPT:

berikan saya kode untuk memprediksi accuracy dari suatu dataset

* Hasil kode chatGPT:

# Memprediksi label/target dari data uji berdasarkan fitur-fitur yang diberikan

y\_pred = mode.predict(X\_test)

# Menghitung akurasi dari model yang telah melakukan prediksi terhadap data uji

accuracy = accuracy\_score(Y\_test, y\_pred)

# Mencetak tingkat akurasi yang telah dihitung sebelumnya dalam format yang lebih mudah dibaca

print(f'Accuracy: {accuracy \* 100:.2f}%')

* Hasil running:

Accuracy: 95.61%

* Penjelasan hasil running: