UTS Pembelajaran Mesin

April 17, 2022

1 UTS Pembelajaran Mesin

Nurmalik Fajar 1197050101 C

2 Mempelajari Data Pasien Penyakit Jantung

Mini Riset ini mengenai klasifikasi pasien yang memiliki penyakit jantung dengan beberapa faktor menggunakan algoritma SVM

3 Business Understanding

3.1 Objektif Bisnis

3.1.1 Latar Belakang

Penyakit jantung adalah kondisi ketika jantung mengalami gangguan. Bentuk gangguan itu sendiri bermacam-macam, bisa berupa gangguan pada pembuluh darah jantung, katup jantung, atau otot jantung. Penyakit jantung juga dapat disebabkan oleh infeksi atau kelainan lahir. Penyakit Jantung masih menjadi penyebab kematian tertinggi di USA hingga saat ini. Oleh karena itu memprediksi apakah pasien akan mengalami penyakit jantung di masa depan sangatlah dibutuhkan untuk pasien dan dokter. Pada Dataset ini dilakukan percobaan untuk memprediksi apakah seorang pasien akan memiliki penyakit jantung dari berbagai faktor mulai dari BMI hingga status Kanker kulit pasien.

3.1.2 Identifikasi Masalah

Dari latar belakang di atas maka identifikasi masalah yang dapat diambil ialah: -Apakah ada faktor yang dapat memprediksi penyakit jantung ?

3.1.3 Metode/Pendekatan Penyelesaian Masalah

Pendekatan Penyelesaian Masalah pada Mini Riset ini ialah sebagai berikut. 1. Memahami masalah 2. Membuat rencana untuk menyelesaikan masalah 3. Melaksanakan rencana pada tahap 2 4. Memeriksa ulang hasil yang didapatkan

dengan Metode penyelesaian masalah sebagai berikut. 1. Data Understanding 2. Data Preparation 3. Modeling 4. Evaluation

3.2 Tujuan Teknis dan Kriteria Kesuksesan

Mini Riset ini bertujuan untuk mengetahui apakah ada faktor yang dapat dijadikan prediksi apakah pasien akan menderita penyakit jantung. Kriteria kesuksesan Mini Riset ini adalah ketika Dataset ini dapat memberikan kesimpulan apakah faktor-faktor dalam dataset dapat melakukan prediksi ya atau tidak penyakit jantung pada pasien.

4 Data Understanding

Membahas Kebutuhan Data; Pengambilan Data; Integrasi Data; Telaah Data; Analisis Karakteristik Data; Validasi Data

Dataset ini berasal dari data CDC (Center for Disease Control). Dataset ini memiliki 17 dimensi dan 320000 data dari rekam medis. Dimensi yang ada yaitu sebagai berikut.

```
ini rincian atribut/fitur/independent variable/kriteria:

BMI - merepresentasikan besar angka BMI (float)
Smoking - merepresentasikan apakah pasien perokok (ya/tidak)
AlcoholDrinking - merepresentasikan apakah pasien mengkonsumsi alkohol (ya/tidak)
Stroke - merepresentasikan apakah pasien menderita stroke (ya/tidak)
PhysicalHealth - merepresentasikan nilai kesehatan jasmani (float)
MentalHealth - merepresentasikan nilai kesehatan mental (float)
DiffWalking - merepresentasikan apakah pasien mengalami kesulitan berjalan (ya/tidak)
Sex - merepresentasikan jenis kelamin pasien (perempuan/laki-laki)
AgeCategory - merepresentasikan kategori umur pasien (per 10 tahun, 80 ke atas)
Race - merepresentasikan ras pasien (white, black, latin, etc)
Diabetic - merepresentasikan apakah pasien menderita diabetes (ya/tidak)
PhysicalActivity - merepresentasikan apakah pasien melakukan aktifitas fisik (ya/tidak)
GenHealth - merepresentasikan kesehatan pasien secara umum (very bad - very good)
SleepTime - merepresentasikan berapa lama pasien tidur setiap harinya (float)
Asthma - merepresentasikan apakah pasien menderita asthma (ya/tidak)
KidneyDisease - merepresentasikan apakah pasien menderita penyakit ginjal (ya/tidak)
SkinCancer - merepresentasikan apakah pasien menderita kanker kulit (ya/tidak)
```

kolom yang dijadikan prediksi adalah kolom yang bertipe data float yaitu kolom 1,5,6, dan 14.

```
ini variabel target/dependet variable/class/label
HeartDisease - merepresentasikan apakah pasien menderita penyakit jantung (ya/tidak)
```

5 Data Preparation

Melakukan Load data, pemilahan data, hingga integrasi data. Pada mini riset ini data yang diambil berasal dari kolom data bertipe data float. Jumlah baris pada Dataset ini sebanyak 320000, pada riset ini hanya diambil 100000 data pertama saja.

[1]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
```

```
# load data

url = 'https://raw.githubusercontent.com/Malik-hub/UTS-Pembelajaran-Mesin/main/
↪heart_2020_cleaned.csv'
df = pd.read_csv(url, sep=',', nrows=100000) #membaca 100000 data pertama

df.head()
```

[1]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth
0	No	16.60	Yes	No	No	3.0
1	No	20.34	No	No	Yes	0.0
2	No	26.58	Yes	No	No	20.0
3	No	24.21	No	No	No	0.0
4	No	23.71	No	No	No	28.0

	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic
0	30.0	No	Female	55-59	White	Yes
1	0.0	No	Female	80 or older	White	No
2	30.0	No	Male	65-69	White	Yes
3	0.0	No	Female	75-79	White	No
4	0.0	Yes	Female	40-44	White	No

	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer
0	Yes	Very good	5.0	Yes	No	Yes
1	Yes	Very good	7.0	No	No	No
2	Yes	Fair	8.0	Yes	No	No
3	No	Good	6.0	No	No	Yes
4	Yes	Very good	8.0	No	No	No

[2]:
```
df.info()

# pastikan tidak ada yg null
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   HeartDisease      100000 non-null  object
 1   BMI               100000 non-null  float64
 2   Smoking           100000 non-null  object
 3   AlcoholDrinking   100000 non-null  object
 4   Stroke            100000 non-null  object
 5   PhysicalHealth    100000 non-null  float64
 6   MentalHealth      100000 non-null  float64
 7   DiffWalking       100000 non-null  object
```

```
 8   Sex               100000 non-null  object
 9   AgeCategory       100000 non-null  object
 10  Race              100000 non-null  object
 11  Diabetic          100000 non-null  object
 12  PhysicalActivity  100000 non-null  object
 13  GenHealth         100000 non-null  object
 14  SleepTime         100000 non-null  float64
 15  Asthma            100000 non-null  object
 16  KidneyDisease     100000 non-null  object
 17  SkinCancer        100000 non-null  object
dtypes: float64(4), object(14)
memory usage: 13.7+ MB
```

[3]:
```
# set independent variable yaitu BMI, PhysicalHealth, MentalHealth, dan␣
 ↪SleepTime
X = df.iloc[:, [1,5,6,14]]
#memilih type data float saja

# set dependent variable/clas/target/label
y = df['HeartDisease']
```

[4]:
```
X
```

[4]:

	BMI	PhysicalHealth	MentalHealth	SleepTime
0	16.60	3.0	30.0	5.0
1	20.34	0.0	0.0	7.0
2	26.58	20.0	30.0	8.0
3	24.21	0.0	0.0	6.0
4	23.71	28.0	0.0	8.0
…	…	…	…	…
99995	31.09	30.0	15.0	6.0
99996	29.53	0.0	0.0	7.0
99997	38.65	0.0	0.0	7.0
99998	50.84	30.0	0.0	4.0
99999	24.02	25.0	25.0	6.0

[100000 rows x 4 columns]

[5]:
```
y
```

[5]:
```
0         No
1         No
2         No
3         No
4         No
        …
99995     No
```

```
99996     No
99997     No
99998     No
99999    Yes
Name: HeartDisease, Length: 100000, dtype: object
```

[6]:
```
# terdapat fitur yang gak sama skalanya seperti BMI dan SleepTime jadi perlu␣
↪dibuat sama skalanya dengan StandardScaler()

from sklearn.preprocessing import StandardScaler

# standarisasi nilai-nilai dari dataset
scaler = StandardScaler()
scaler.fit(X)
X = scaler.transform(X)
```

[7]:
```
X
```

[7]:
```
array([[-1.82236734, -0.05648764,  3.2814416 , -1.42311743],
       [-1.23583865, -0.43012437, -0.49122076, -0.06756739],
       [-0.25724533,  2.06078711,  3.2814416 ,  0.61020763],
       …,
       [ 1.6356427 , -0.43012437, -0.49122076, -0.06756739],
       [ 3.54734984,  3.30624284, -0.49122076, -2.10089245],
       [-0.65871952,  2.68351498,  2.65266454, -0.74534241]])
```

6 Modeling

Dilakukan fitting model, memisahkan data untuk training 80% dan testing 20%. Dilakukan juga fitting algoritma svm dengan rbf kernel.

[8]:
```
import warnings
warnings.filterwarnings('ignore')

# training 80% testing 20%

from sklearn.model_selection import train_test_split

# memisahkan data untuk training dan testing
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=100)
```

[9]:
```
# fiting algoritma SVM dengan rbf kernal

from sklearn.svm import SVC
```

```
# membuat objek SVC dan memanggil fungsi fit untuk melatih model
clf = SVC(kernel='rbf')
clf.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

7 Evaluasi

Membahas mengenai hasil pengujian terhadap model.

[10]:
```
# Menampilkan skor akurasi prediksi
clf.score(X_test, y_test)
```

[10]: 0.91325

[11]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

          No       0.91      1.00      0.95     18266
         Yes       0.00      0.00      0.00      1734

    accuracy                           0.91     20000
   macro avg       0.46      0.50      0.48     20000
weighted avg       0.83      0.91      0.87     20000
```

[12]:
```
# coba classify single row

single = df.head(4)
single
```

[12]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer
0	No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Yes	Very good	5.0	Yes	No	Yes
1	No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Yes	Very good	7.0	No	No	No
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Yes	Fair	8.0	Yes	No	No
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	No	Good	6.0	No	No	Yes

```
0              Yes  Very good        5.0    Yes             No         Yes
1              Yes  Very good        7.0     No             No          No
2              Yes       Fair        8.0    Yes             No          No
3               No       Good        6.0     No             No         Yes
```

[13]:
```
S = single.iloc[:,[1,5,6,14]]
S
```

[13]:
```
     BMI  PhysicalHealth  MentalHealth  SleepTime
0  16.60             3.0          30.0        5.0
1  20.34             0.0           0.0        7.0
2  26.58            20.0          30.0        8.0
3  24.21             0.0           0.0        6.0
```

[14]:
```
# standarisasi nilai-nilai dari dataset
scaler = StandardScaler()
scaler.fit(S)
S = scaler.transform(S)
S
```

[14]:
```
array([[-1.40331529, -0.3306122 ,  1.        , -1.34164079],
       [-0.41908666, -0.69128005, -1.        ,  0.4472136 ],
       [ 1.22304881,  1.71317231,  1.        ,  1.34164079],
       [ 0.59935313, -0.69128005, -1.        , -0.4472136 ]])
```

[15]:
```
y_pred = clf.predict(S)
y_pred
```

[15]:
```
array(['No', 'No', 'No', 'No'], dtype=object)
```

Dari pemodelan dan evaluasi data di atas dapat disimpulkan bahwa keempat faktor yaitu BMI, MentalHealth, PhysicalHealth, dan SleepTime hanya dapat memprediksi apakah seseorang tidak menderita penyakit jantung (HeartDisease). Pada penelitian selanjutnya diharapkan untuk melakukan modeling dan evaluasi pada faktor-faktor lain yang ada dalam dataset ini agar bisa lebih memahami apakah faktor-faktor yang ada dalam dataset ini bisa memprediksi penyakit jantung atau tidak.