

Implementasi Data Science Menggunakan Algoritma Regresi Logistik Untuk Memprediksi Hubungan Gejala Dengan Penyakit Jantung

Nabila Asshafa Putri

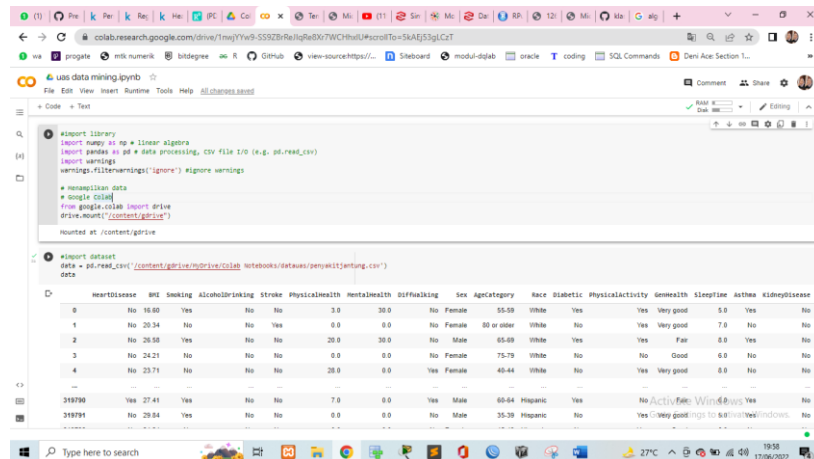
Penyakit jantung adalah salah satu penyakit penyebab kematian nomor satu di Indonesia. Sistem perawatan kesehatan di seluruh dunia mengalami kesulitan karena kurangnya keahlian staf medis dalam menentukan dan memprediksi penyakit ini. Tentunya dengan kemajuan teknologi angka tersebut dapat diminimalisir. Salah satu teknologi yang ada sekarang yaitu machine learning yang dapat digunakan untuk mendeteksi sebuah penyakit. Machine Learning merupakan sebuah metode berbasis komputer yang tidak perlu diatur dahulu oleh manusia dan dapat belajar dengan bantuan data dan akan semakin pintar seiring dengan banyaknya data yang telah diolah (belajar melalui pengalaman). Metode ini sering digunakan dalam menyelesaikan kasus klasifikasi dan clustering dan biasanya digunakan untuk menangani data dalam skala besar atau big data. Machine learning sendiri bukanlah sebuah teknologi yang memiliki keakuratan 100% untuk melakukan analisa data dan mendapat kesimpulan berdasarkan analisa data tersebut. Namun keakuratan yang dihasilkan cukup efektif sehingga machine learning telah terbukti membantu di Bidang Kesehatan. Ada banyak algoritma atau metode klasifikasi yang dapat dipakai di machine learning salah satunya yaitu Regresi Logistik. Regresi logistik digunakan untuk menggambarkan data dan untuk menjelaskan hubungan antara satu variabel biner dependen dan satu atau lebih variabel independen.

Cara yang digunakan untuk menyelesaikan masalah ini dikatakan dengan metodologi. Kerangka metodologi dijelaskan pada gambar dibawah ini :



1) Pengumpulan data

Dataset yang digunakan merupakan dataset publik yang merupakan contoh nyata dari penyakit jantung. Dataset yang digunakan berasal dari Kaggle dataset repository (Heart Failure Prediction) yang dapat diunduh dari <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease>. Dataset ini memiliki 18 atribut dengan total sample valid sebanyak 319795 sample.



```
# Import library
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

# Menampilkan data
# Sample Colab
from google.colab import drive
drive.mount('/content/gdrive')

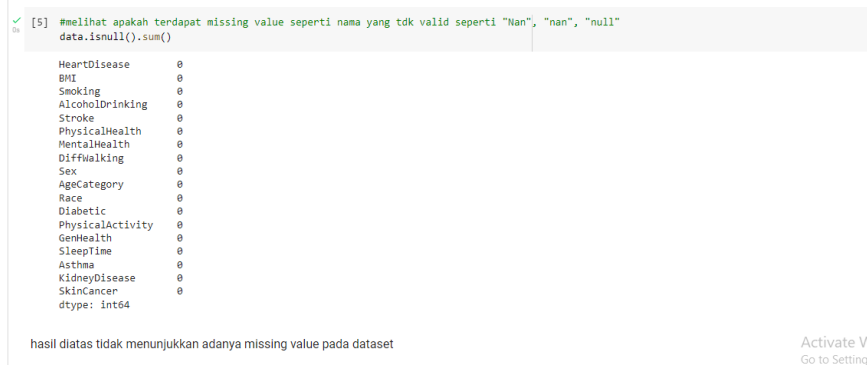
# Import dataset
data = pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/dataset/penyakitjantung.csv')
data
```

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease
0	No	35.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Yes	Very good	5.0	Yes	No
1	No	20.34	No	No	No	0.0	0.0	No	Female	60 or older	White	No	Yes	Very good	7.0	No	No
2	No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Yes	Fair	6.0	Yes	No
3	No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	No	Good	6.0	No	No
4	No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Yes	Very good	8.0	No	No
...
319790	Yes	27.41	Yes	No	No	7.0	0.0	Yes	Male	60-64	Hispanic	Yes	No	Active	6.0	Yes	No
319791	No	29.04	Yes	No	No	0.0	0.0	No	Male	35-39	Hispanic	No	Yes	Very good	6.0	Yes	No

2) Pembersihan data

Pembersihan data adalah suatu prosedur untuk memastikan kebenaran, konsistensi, dan kegunaan suatu data yang ada dalam dataset. Caranya adalah dengan mendeteksi adanya error atau corrupt pada data, kemudian memperbaiki atau menghapus data jika memang diperlukan.

- Cek missing value



```
[5] #melihat apakah terdapat missing value seperti nama yang tdk valid seperti "NaN", "nan", "null"
data.isnull().sum()
```

HeartDisease	0
BMI	0
Smoking	0
AlcoholDrinking	0
Stroke	0
PhysicalHealth	0
MentalHealth	0
DiffWalking	0
Sex	0
AgeCategory	0
Race	0
Diabetic	0
PhysicalActivity	0
GenHealth	0
SleepTime	0
Asthma	0
KidneyDisease	0
SkinCancer	0
dtype:	int64

hasil diatas tidak menunjukkan adanya missing value pada dataset

- Pada tahap ini akan dilakukan penanganan data kategoris dengan melakukan konversi ke biner/encode ordinal data(label encoder)

Nilai 0 = No; 1= Yes

```
[9] #melihat beberapa baris data
data.head()
```

	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

Dalam dataset, beberapa variabel kategori masih memiliki nilai yes atau no. maka perlu dilakukan konversi/ubah ke biner. keterangan: yes = 1
no = 0

```
[10] column_yesno = ["HeartDisease", "Smoking", "AlcoholDrinking", "Stroke", "DiffWalking", "Diabetic", "PhysicalActivity", "Asthma", "KidneyDisease", "Skincancer"]
data[column_yesno] = data[column_yesno].apply(lambda x: x.map({'Yes':1, 'No':0})) # Mengubah ke biner

#melihat data kembali
data.head()
```

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	Skincancer
0	0	16.60	1	0	0	3.0	30.0	0	Female	55-59	White	1	1	Very good	5.0	1	0	1
1	0	20.34	0	0	1	0.0	0.0	0	Female	80 or older	White	0	1	Very good	7.0	0	0	0
2	0	26.58	1	0	0	20.0	30.0	0	Male	65-69	White	1	1	Fair	8.0	1	0	0

3) Persiapan data

Langkah selanjutnya yaitu melakukan persiapan data. Dimana dalam persiapan data, data yang akan digunakan dilakukan pembagian variabel menjadi variabel dependen dan variabel independent.

```
us = us.data

HeartDisease BMI Smoking AlcoholDrinking Stroke PhysicalHealth MentalHealth DiffWalking Diabetic PhysicalActivity ... AgeCategory_80 or older Race_Asiian Race_black Race_hispanic Race_Other Race_white GenHealth_I
0 0 16.60 1 0 0 3.0 30.0 0 1 1 ... 0 0 0 0 0 0 1
1 0 20.34 0 0 1 0.0 0.0 0 0 1 ... 1 0 0 0 0 0 1
2 0 26.58 1 0 0 20.0 30.0 0 1 1 ... 0 0 0 0 0 0 1
3 0 24.21 0 0 0 0.0 0.0 0 0 0 ... 0 0 0 0 0 0 1
4 0 23.71 0 0 0 20.0 0.0 1 0 1 ... 0 0 0 0 0 0 1
5 rows x 36 columns
```

```
# kemungkinan memiliki penyakit jantung berdasarkan variabel lain.
# kolom 'HeartDisease' akan dijadikan variabel dependen
y = data.pop('HeartDisease')
x = data
x.head()
```

	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Diabetic	PhysicalActivity	SleepTime	AgeCategory_80 or older	Race_Asiian	Race_black	Race_hispanic	Race_Other	Race_white	GenHealth_Fair
0	16.60	1	0	0	3.0	30.0	0	1	1	5.0	...	0	0	0	0	0	1
1	20.34	0	0	1	0.0	0.0	0	0	1	7.0	...	1	0	0	0	0	1
2	26.58	1	0	0	20.0	30.0	0	1	1	8.0	...	0	0	0	0	0	1
3	24.21	0	0	0	0.0	0.0	0	0	0	6.0	...	0	0	0	0	0	1
4	23.71	0	0	0	20.0	0.0	1	0	1	8.0	...	0	0	0	0	0	1

4) Implementasi algoritma

- Splitting data

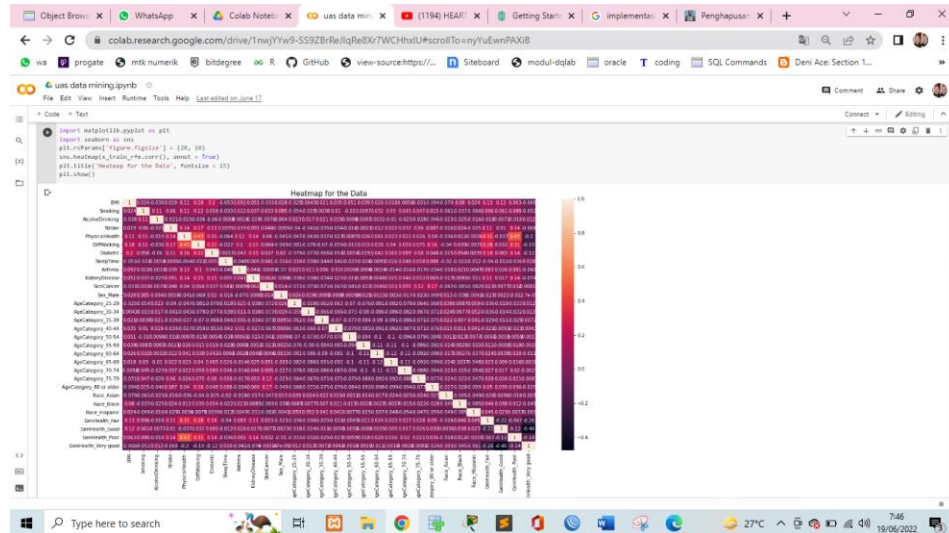
Untuk melakukan prediksi pada testing data, maka bagi data menjadi 75% training dan 25% testing.

```
[14] from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, train_size=0.75, test_size=0.25, random_state=100)

x_train.head()
```

	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Diabetic	PhysicalActivity	SleepTime	AgeCategory_80 or older	Race_Asiian	Race_black	Race_hispanic	Race_Other	Race_white	GenHealth_I
224032	30.23	0	0	0	3.0	30.0	0	0	1	7.0	...	0	0	0	1	0	0
69666	29.21	1	0	0	30.0	25.0	1	0	0	8.0	...	0	0	0	0	0	1
202674	24.03	0	0	0	10.0	3.0	0	0	1	7.0	...	0	0	0	0	0	1
301567	33.89	1	0	0	27.0	30.0	1	1	1	6.0	...	0	0	0	0	0	1
188559	20.18	0	0	0	5.0	0.0	0	0	1	8.0	...	0	0	0	0	0	1

- Visualisasi data menggunakan heatmap
Grafik pada gambar dibawah ini menunjukkan hubungan, yang erat hubungannya berwarna semakin terang, sedangkan yang semakin jauh berwarna gelap.



- Membangun model linear dengan GLM(Generalized Linear Model)

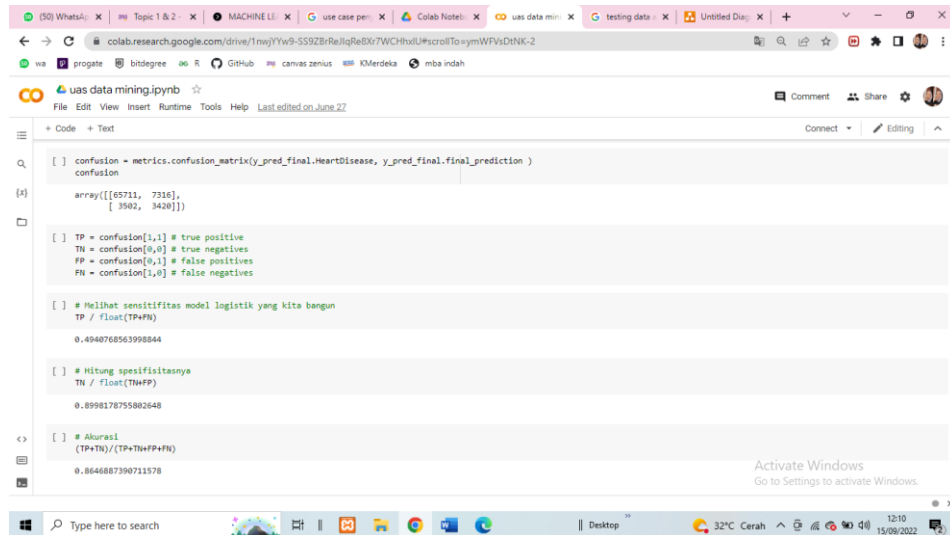
```
# Membangun model linier dengan statsmodel

import statsmodels.api as sm
logit = sm.GLM(list(y_train), (sm.add_constant(x_train_rfe)), family = sm.families.Binomial())
logit.fit().summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	y	No. Observations:	239849		
Model:	GLM	DF Residuals:	239810		
Model Family:	Binomial	DF Model:	29		
Link Function:	logit	Scale:	1.0000		
Method:	IRLS	Log-Likelihood:	-54341		
Date:	Fri, 17 Jun 2022	Deviance:	1.0868e+05		
Time:	09:23:03	Pearson chi2:	2.22e+05		
No. Iterations:	8				
Covariance Type:	nonrobust				
	coef	std err	z	P> z	[0.025 0.975]
const	-3.1533	0.013	-234.144	0.000	-3.180 -3.127
BMI	0.0912	0.008	7.349	0.000	0.045 0.078
Smoking	0.1899	0.008	23.258	0.000	0.174 0.206
AlcoholDrinking	-0.0554	0.010	-5.782	0.000	-0.076 -0.037
Stroke	0.2018	0.005	40.397	0.000	0.192 0.212
PhysicalHealth	0.0243	0.008	3.107	0.002	0.009 0.040
DiffWalking	0.0728	0.007	10.108	0.000	0.059 0.087
Diabetic	0.1601	0.008	25.051	0.000	0.148 0.173
SleepTime	-0.0520	0.007	-7.185	0.000	-0.066 -0.038
Asthma	0.0908	0.008	11.937	0.000	0.078 0.105
KidneyDisease	0.1072	0.005	20.177	0.000	0.097 0.118
SkinCancer	0.0328	0.007	5.003	0.000	0.020 0.046
Sex_Male	0.3495	0.008	41.526	0.000	0.330 0.363
AgeCategory_25-29	-0.2035	0.024	-8.332	0.000	-0.251 -0.156
AgeCategory_30-34	-0.1187	0.020	-5.943	0.000	-0.168 -0.080
AgeCategory_35-39	-0.1011	0.010	-9.410	0.000	-0.128 -0.064
AgeCategory_40-44	0.2043	0.013	16.328	0.000	0.180 0.229
AgeCategory_45-49	0.2795	0.012	22.713	0.000	0.256 0.304
AgeCategory_50-54	0.3799	0.012	31.205	0.000	0.358 0.404
AgeCategory_55-59	0.4535	0.012	37.923	0.000	0.430 0.477
AgeCategory_60-64	0.5203	0.011	45.606	0.000	0.498 0.543

Setelah mendapatkan model, selanjutnya melakukan testing dengan memasukkan data testing kedalam model dan mendapatkan hasilnya. Hasil tersebut dapat diolah dalam confusion matrix untuk mendapatkan analisa terhadap keakuratan dalam prediksi model tersebut.



```
[ ] confusion = metrics.confusion_matrix(y_pred_final.HeartDisease, y_pred_final.final_prediction )
confusion
array([[65711, 7316],
       [ 3502, 3420]])

[ ] TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives

[ ] # Melihat sensitivitas model logistik yang kita bangun
TP / float(TP+FN)
0.4940768563998844

[ ] # Hitung spesifisitasnya
TN / float(TN+FP)
0.8998178755802648

[ ] # Akurasi
(TP+TN)/(TP+TN+FP+FN)
0.8648887390711578
```

5) Evaluasi hasil

Untuk dataset yang digunakan berasal dari situs Kaggle dataset repository (Heart Failure Prediction) yang dapat diunduh dari <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease>. Dataset ini memiliki 18 atribut dengan total sample valid sebanyak 319795 sample. Berdasarkan hasil eksperimen, terdiri dari beberapa tahap diantaranya data preprocessing, split data, model fitting dengan GLM (Generalized Linear Model) dan evaluasi hasil menggunakan confusion matrix untuk menentukan performa model tersebut. Berdasarkan hasil testing prediksi hubungan penyakit jantung dengan gejala menggunakan algoritma regresi logistik tampaknya memiliki sensitivitas dan spesifisitas yang layak. Oleh karena itu model ini tampaknya dapat digunakan untuk memprediksi, karena diperoleh hasil sensitivitas 49.4%, spesifisitas 89.98% dan akurasi 86.46%.

KESIMPULAN

Dari hasil analisa prediksi penyakit jantung dengan menggunakan regresi logistic dapat diambil kesimpulan diagnosis penyakit jantung dengan menggunakan logistik regresi memiliki keunggulan yang berbeda beda terhadap metode lainnya pada model analisa confusion matrix. Dengan hasil yang diperoleh tampaknya model tersebut dapat digunakan untuk memprediksi. Penerapan data science sangat membantu dibidang kesehatan salah satunya dapat memprediksi hubungan antara gejala dengan penyakit jantung.

Github : <https://github.com/nabilaass/nabilaass-github.io.git>

LinkedIn : <https://www.linkedin.com/in/nabila-asshafa-putri-0769b8246/>