

Laporan Case Base 1

Pembalajaran Mesin

Disusun untuk memenuhi tugas Mata Kuliah Pembelajaran Mesin

Dosen Pengampu : Bedy Purnama (BDP)



oleh:

Syahdan Naufal N I

1301204110

IF4401

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

Saya Syahdan Naufal mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi

1. Ikhtisar

- Alur program

Secara garis besar nanti saya akan menggunakan dataset yang di namakan data data_arrhythmia setelah itu saya akan memilah dan mengambil beberapa kolom saja dan mencari kemungkinan korelasi yang mungkin memiliki kesinambungan lalu saat pengolahan data saya akan menggunakan algoritma supervised learning yaitu Multilayer Perceptron(MLP) merupakan algoritma yang mengadopsi cara kerja jaringan saraf pada makhluk hidup dimana nantinya terdapat 3 layer input layer, hidden layer, output layer.

- DataSet

Diberikan sebuah data set dengan nama data_arrhythmia.csv yang isi 452 record dan 280 columns

	age	sex	height	weight	qrs_duration	p-r_interval	q-t_interval	t_interval	p_interval	qrs	...	KY	KZ	LA	LB	LC	LD	LE	LF	LG	diagnosis	
0	75	0	190	80	91	193	371	174	121	-16	...	0.0	9.0	-0.9	0.0	0	0.9	2.9	23.3	49.4	8	
1	56	1	165	64	81	174	401	149	39	25	...	0.0	8.5	0.0	0.0	0	0.2	2.1	20.4	38.8	6	
2	54	0	172	95	138	163	386	185	102	96	...	0.0	9.5	-2.4	0.0	0	0.3	3.4	12.3	49.0	10	
3	55	0	175	94	100	202	380	179	143	28	...	0.0	12.2	-2.2	0.0	0	0.4	2.6	34.6	61.6	1	
4	75	0	190	80	88	181	360	177	103	-16	...	0.0	13.1	-3.6	0.0	0	-0.1	3.9	25.4	62.8	7	
...
447	53	1	160	70	80	199	382	154	117	-37	...	0.0	4.3	-5.0	0.0	0	0.7	0.6	-4.4	-0.5	1	
448	37	0	190	85	100	137	361	201	73	86	...	0.0	15.6	-1.6	0.0	0	0.4	2.4	38.0	62.4	10	
449	36	0	166	68	108	176	365	194	116	-85	...	0.0	16.3	-28.6	0.0	0	1.5	1.0	-44.2	-33.2	2	
450	32	1	155	55	93	106	386	218	63	54	...	-0.4	12.0	-0.7	0.0	0	0.5	2.4	25.0	46.6	1	
451	78	1	160	70	79	127	364	138	78	28	...	0.0	10.4	-1.8	0.0	0	0.5	1.6	21.3	32.8	1	
452 rows x 280 columns																						

2. Pra-Pemrosesan data

Di bagian ini kita masuk kedalam proses pengolahan data, di awali kita memilah data lalu memperbaiki data, tujuannya agar data yang di pakai lebih efektif dari data yang saya gunakan terdapat banyak data yang null maka dari itu saya mengambil beberapa atribut yaitu

```
#Melihat seberapa banyak missing values
data.isna().sum()

age      0
sex      0
height   0
weight   0
qrs_duration  0
p-r_interval  0
q-t_interval  0
t_interval  0
p_interval  0
qrs      0
QRST     1
q_wave   0
r_wave   0
s_wave   0
diagnosis  0
dtype: int64
```

dari *screenshot* diatas terdapat data yang saya ambil dan pada atribut “QRST” terdapat nilai yang null, korelasi antar atribut

```
# melihat data mean untuk nantinya di isi ke data yang kosong
data.mean()
```

```
age          46.471239
sex          0.550885
height       166.188053
weight       68.170354
qrs_duration 88.920354
p-r_interval 155.152655
q-t_interval 367.207965
t_interval   169.949115
p_interval   90.004425
qrs          33.676991
QRST         36.716186
q_wave       5.628319
r_wave       51.628319
s_wave       20.920354
diagnosis    3.880531
dtype: float64
```

Data mean yang nanti di masukan ke data yang null

```
data['QRST'] = data['QRST'].fillna(37)
data.isna().sum()
```

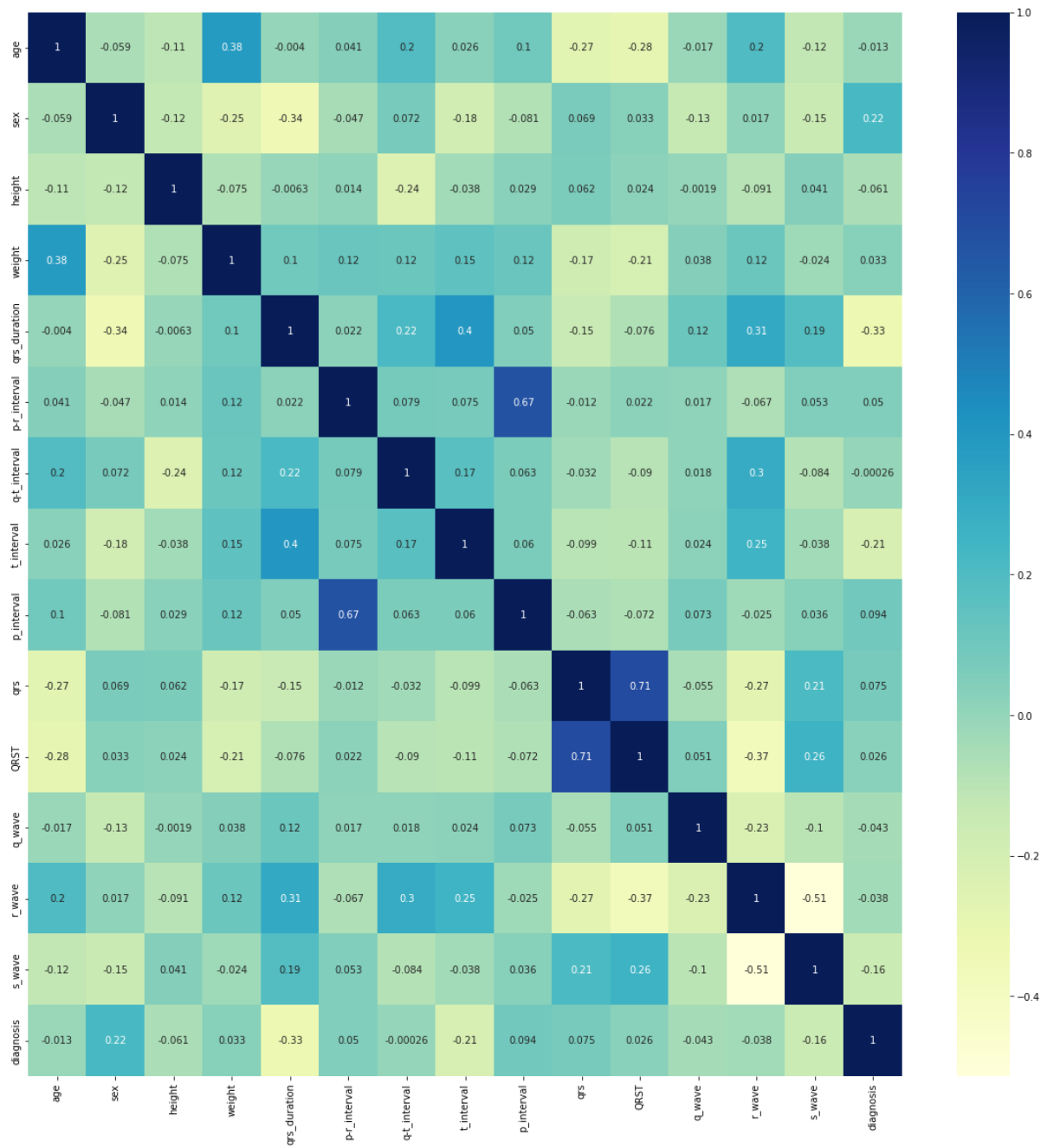
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide

"""Entry point for launching an IPython kernel.

```
age          0
sex          0
height       0
weight       0
qrs_duration 0
p-r_interval 0
q-t_interval 0
t_interval   0
p_interval   0
qrs          0
QRST         0
q_wave       0
r_wave       0
s_wave       0
diagnosis    0
dtype: int64
```

Pada ScreenShot di atas data kosong pada “QRST” sudah terisi dengan data mean



gambar diatas merupakan korelasi antar atribut

KORELASI

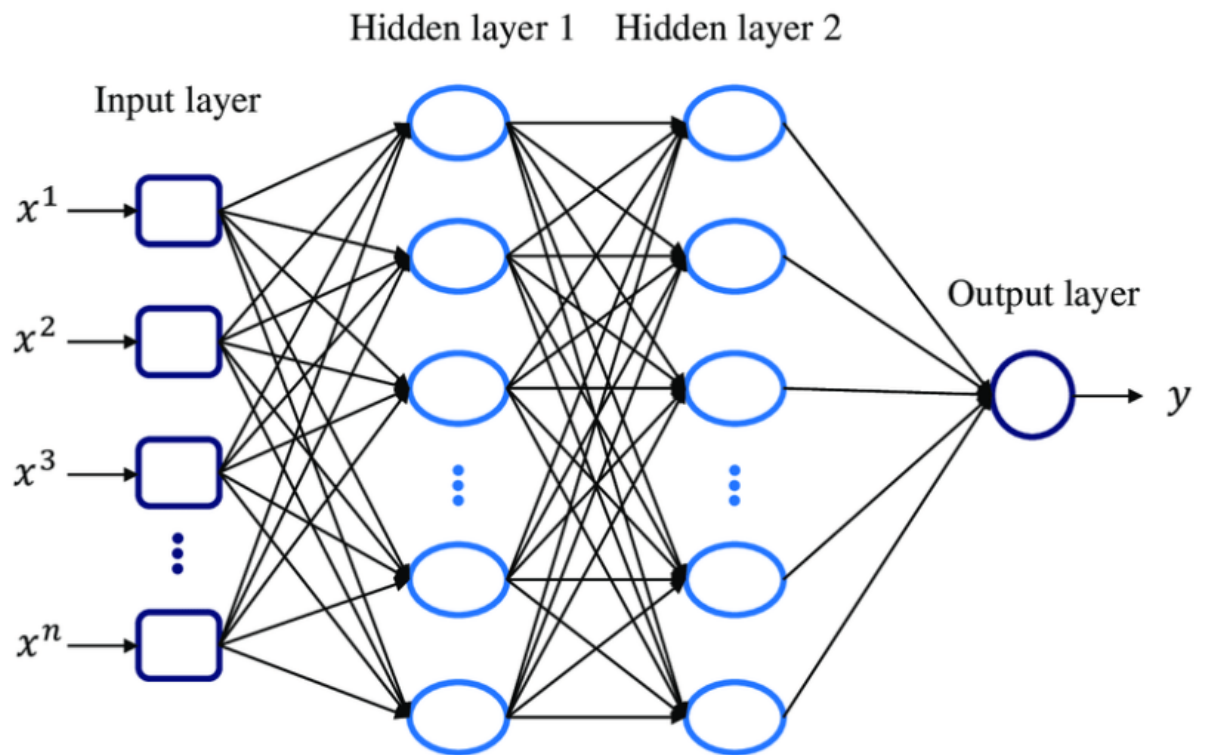
```

corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
g=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="YlGnBu")

```

3. Penerapan Algoritma

MLP merupakan jaringan saraf tiruan feedforward yang menghasilkan serangkaian output dari serangkaian input, MLP memiliki ciri khas dimana terdapat hidden layer yang berada di antara lapisan output dan input yang jumlah layer-nya berubah ubah seperti gambar di bawah.



selanjutnya saya akan melakukan model, sebelum melakukan modeling saya akan melakukan labeling dan membagi dataset menjadi training set dan testing set, saya menggunakan library yaitu `train_test_split` dengan memabagi 20%:80% dan random state nya adalah 3.

- Labeling dan Split data

```
#extracting x and y
x = data.drop(["diagnosis"], axis=1).values

y = data["diagnosis"].values
```

```
#Proses Split data

# x = data[['age','sex','height','weight','qrs_duration','p-r_interval','q-t_interval','t_interval','P_in
# y = data[['diagnosis']]

x_train, x_test, y_train, y_test = train_test_split(x,y , test_size=0.2, random_state=3, shuffle=True)
```

selanjutnya merupakan modeling dengan metode MLP classification

Modelling

```
▶ #Pembuatan model dengan MLP classification
clasf = MLPClassifier(hidden_layer_sizes=(150,100,50),
                      max_iter=300,
                      solver = 'lbfgs',
                      verbose = True,
                      random_state=5)

clasf.fit(x_train,y_train)
```

4. Evaluasi

```
# Make prediction on test dataset
y_predic = clasf.predict(x_test)
y_train_pred = clasf.predict(x_train)
print("Hasil Prediksi : ")
print(y_predic)

#Calculate accuracy with accuracy library
hasil = accuracy_score(y_predic,y_test)
print(f"Hasil akurasi testing: {hasil * 100}")

hasil_train = accuracy_score(y_train_pred,y_train)
print(f"Hasil Akurasi train: {hasil_train} ")

persen = hasil * 100
persen_train = hasil_train * 100
print("Hasil Testing = ",round(persen,2),"%")
print("Hasil Training = ",round(persen_train,2),"%")
```

```
Hasil Prediksi :
[0 0 0 1 0 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 1 1 0 1 0 0 1 0 0 1 1 1
 0 0 0 1 1 1 1 1 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 1 0 0 1 1 0 0 0 0 1 1
 0 1 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1]
Hasil akurasi testing: 65.93406593406593
Hasil Akurasi train: 0.8975069252077562
Hasil Testing = 65.93 %
Hasil Training = 89.75 %
```

dari hasil split data dengan perbandingan 20%:80% ternyata setelah diolah hanya memunculkan hasil prediksi testing 66% dan Training 90%

PRESENTASI VIDEO

- Link google Collab:
https://colab.research.google.com/drive/1BxYfIIY9HPYeiW2FRp6o_2WrZ6KiA_p?usp=sharing
- Link Presentasi:
https://youtu.be/MJAu_35nhVo