

LAPORAN
CASEBASE PEMBELAJARAN MESIN DENGAN MODEL ANN (JARINGAN
SARAF TIRUAN) MENGGUNAKAN DATASET ARRHYTHMIA
2021/2022



Disusun oleh :
Muhammad Rafli Ramadhan (1301200204)

(IF 44 04)

S1 INFORMATIKA
FAKULTAS INFORMATIKA

BAB I

IKHTISAR KUMPULAN DATA YANG DIPILIH

- 1.1 Dataset
- 1.2 Pra-Processing data

BAB II

RINGKASAN PRA-PEMROSESAN DATA YANG DIUSULKAN

- 2.1 Data Cleaning
- 2.2 Normalisasi Data
- 2.3 Data Reduction

BAB III

MENERAPKAN ALGORITMA YANG DI PILIH

- 3.1 Algoritma ANN (Artificial Neural Network)

BAB IV

EVALUASI HASIL

- 4.1 Evaluasi Model

BAB V

LINK & PRESENTASI VIDEO

- 5.1 Link Presentasi Video
- 5.2 Link Colab
- 5.3 Link Github

BAB I

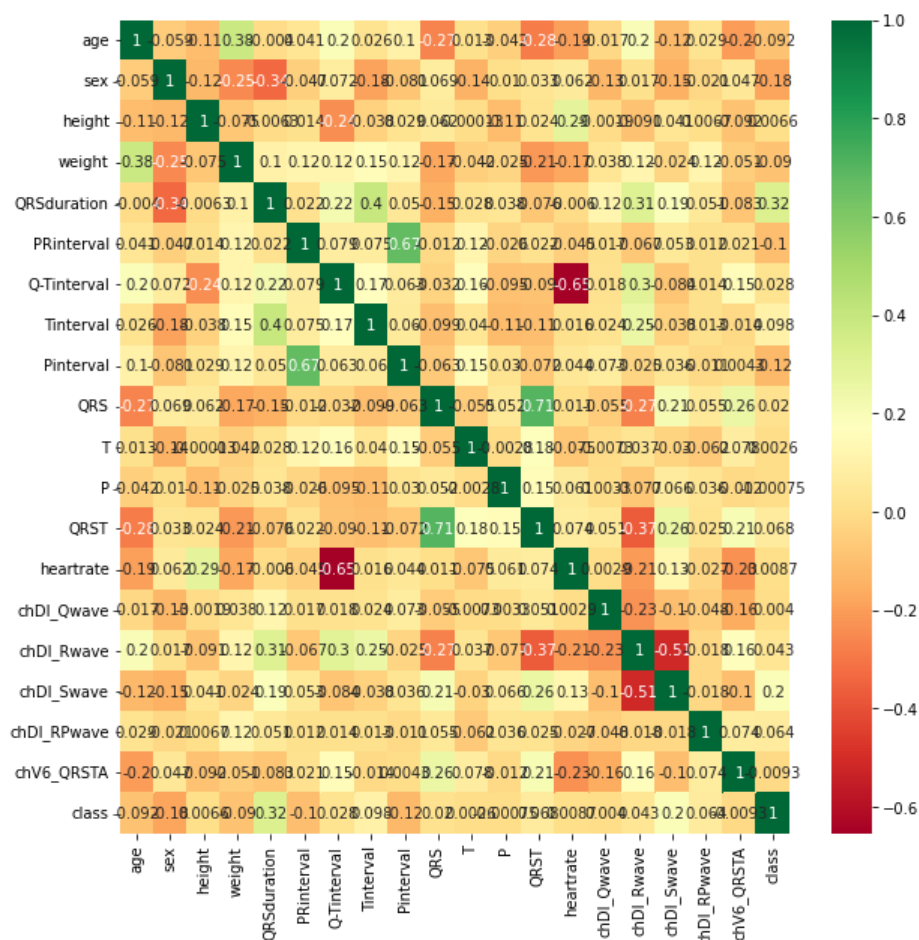
IKHTISAR KUMPULAN DATA YANG DIPILIH

1.1 Dataset

Data yang diambil adalah data arrhythmia, yaitu data gangguan irama jantung. Datanya memiliki banyak kolom 280 dan record datanya kurang lebih 400 yang membuat datanya akan kurang saat di training, karena data mentahnya sangat terbatas.

1.2 Pra-Processing data

Data yang dimiliki untuk pra-processing data agak mendominasi banyaknya kolom pada data dan hanya 400 data mentahnya. Pendekatan dengan korelasi. Korelasi dengan data lainnya akan membuat data yang kita train akan lebih baik, karena datanya bisa dilihat berkorelasi baik atau tidaknya.



gambar grafik korelasi data

Terlihat bahwa data masih lumayan berkorelasi dengan data lainnya, jadi bisa disimpulkan bahwa data tersebut masih bisa untuk di latih.

BAB II

RINGKASAN PRA-PEMROSESAN DATA YANG DIUSULKAN

2.1 Data Cleaning

Karena banyaknya kolom pada data, maka akan kemungkinan data yang kosong atau tidak teratur terjadi, maka akan dilakukan data cleaning, berikut adalah tahapan yang dilakukan



QRS	0
T	8
P	22
J	376
QRST	1

data kosong

Terlihat pada dataframe T, P, J dan QRST terdapat nilai kosong, maka akan dilakukan penghapusan pada dataframe J karena sudah terlalu banyak. Dan untuk dataframe T, P, QRST akan di isi dengan nilai rata-rata seperti ini

```
df['T'].fillna(df['T'].mean(), inplace = True)
df['P'].fillna(df['P'].mean(), inplace = True)
df['QRST'].fillna(df['QRST'].mean(), inplace = True)
df['heartrate'].fillna(df['heartrate'].mean(), inplace = True)
df.drop(columns = ['J'], inplace = True)
df.isna().sum()
```

data akan diisikan dengan nilai rata-rata, sedangkan data J dihapus



QRS	0
T	0
P	0
QRST	0

maka data setelah di cleaning akan seperti ini

```
df.loc[df["height"] == 608, "height"] = 168
df.loc[df["height"] == 780, "height"] = 178
```

memperbaiki data yang typo

Data yang setelah di cleaning akan bersih tanpa adanya nilai kosong dan typo, data akan bagus untuk dilakukan training

2.2 Normalisasi Data

Setelah data diperbaiki dengan data cleaning maka data akan diolah dan sebaiknya di normalisasikan agar mempermudah penghitungan dengan model yang ingin dipakai. Sebelum dilakukan normalisasi sebaiknya mengubah tipe datanya menjadi float.

```
df = df.astype(float)
df.info()
```

mengubah tipe data menjadi float

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

x_train, x_test, y_train, Y_test = train_test_split(x, y, train_size=0.3)
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.fit_transform(x_test)
```

kodingan normalisasi data

Setelah melakukan normalisasi data, data akan siap di training dengan model yang ingin digunakan.

2.3 Data Reduction

Selanjutnya data yang sudah di normalisasi, akan di reduksikan datanya menggunakan metode PCA untuk mengringkas dimensi pada data.

```
from sklearn.decomposition import PCA

pca = PCA(n_components=0.98)

x_train_reduced = pca.fit_transform(x_train)
x_test_reduced = pca.fit_transform(x_test)
```

kodingan data reduction menggunakan PCA

metode PCA sangat berguna untuk model ANN yang menggunakan dimensi pada data hidingnya.

BAB III

MENERAPKAN ALGORITMA YANG DI PILIH

3.1 Algoritma ANN (Artificial Neural Network)

Pada casebase pembelajaran mesin, algoritma atau model yang dipakai adalah ANN. ANN adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan sistem saraf manusia.

```
from keras.models import Sequential
from keras.layers import *

model = Sequential()
model.add(Dense(126, input_dim= 12, input_shape = ([15]), activation = 'relu'))
model.add(Dense(64))
model.add(Dense(32))
model.add(Dense(16))
model.add(Dense(1))

model.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

Kodingan model ANN menggunakan library keras

ANN ini menggunakan sequential model dengan menggunakan 3 hidden layer, 1 input layer dan 1 output layer. input layer menggunakan relu activation function.

```
model.summary()

Model: "sequential_5"

```

Layer (type)	Output Shape	Param #
dense_25 (Dense)	(None, 126)	2016
dense_26 (Dense)	(None, 64)	8128
dense_27 (Dense)	(None, 32)	2080
dense_28 (Dense)	(None, 16)	528
dense_29 (Dense)	(None, 1)	17

```

=====
Total params: 12,769
Trainable params: 12,769
Non-trainable params: 0

```

summary dari hasil training menggunakan model ANN

BAB IV

EVALUASI HASIL

4.1 Evaluasi Model

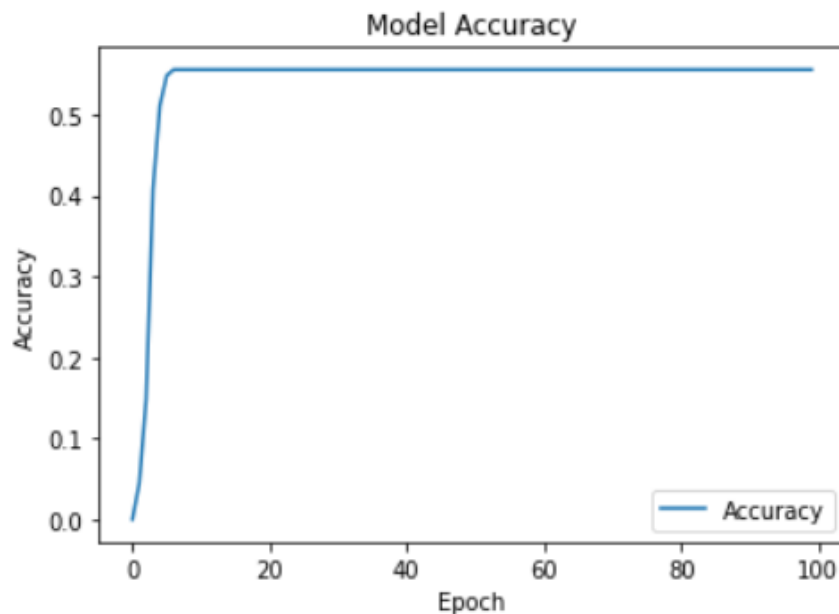
Hasil dari data yang telah ditraining dengan model ANN sebagai berikut

- Menghitung akurasi dan loss pada model

```
Epoch 2/100  
3/3 [=====] - 0s 6ms/step - loss: 3.7116 - accuracy: 0.0444  
Epoch 3/100  
3/3 [=====] - 0s 5ms/step - loss: -0.0604 - accuracy: 0.1481  
Epoch 4/100  
3/3 [=====] - 0s 6ms/step - loss: -4.8989 - accuracy: 0.4074  
Epoch 5/100  
3/3 [=====] - 0s 9ms/step - loss: -8.1780 - accuracy: 0.5111  
Epoch 6/100  
3/3 [=====] - 0s 8ms/step - loss: -15.4040 - accuracy: 0.5481  
Epoch 7/100  
3/3 [=====] - 0s 5ms/step - loss: -19.3137 - accuracy: 0.5556  
Epoch 8/100  
3/3 [=====] - 0s 5ms/step - loss: -23.5787 - accuracy: 0.5556  
Epoch 9/100  
3/3 [=====] - 0s 5ms/step - loss: -26.5056 - accuracy: 0.5556  
Epoch 10/100  
3/3 [=====] - 0s 5ms/step - loss: -30.2713 - accuracy: 0.5556  
Epoch 11/100  
3/3 [=====] - 0s 5ms/step - loss: -35.9617 - accuracy: 0.5556  
Epoch 12/100  
3/3 [=====] - 0s 5ms/step - loss: -36.7175 - accuracy: 0.5556  
Epoch 13/100
```

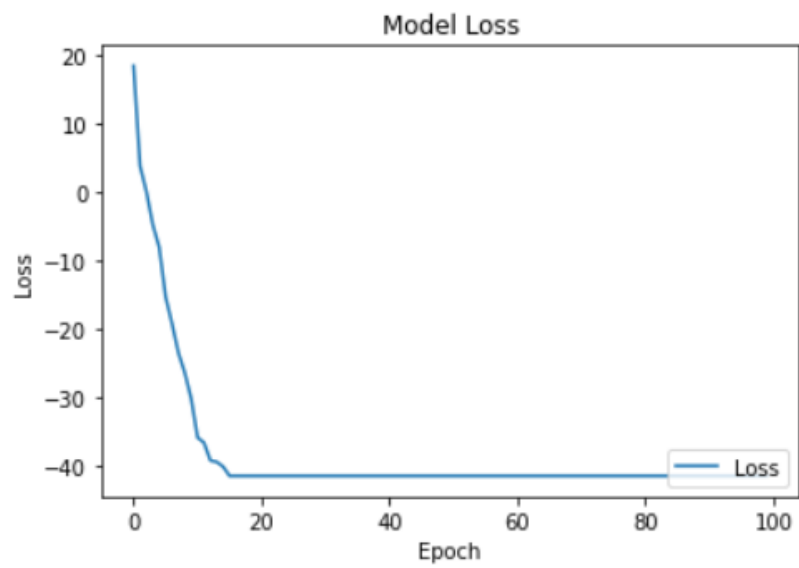
hasil dari akurasi dan loss

- Akurasi pada model



grafik akurasi pada model

- Loss pada model



grafik loss pada model

Bisa disimpulkan bahwa model masih kurang bagus untuk akurasinya, kemungkinan pada data dengan kolom yang banyak tetapi recordnya masih sedikit. Dilihat dari loss pada model adalah high learning rate.

BAB V

LINK & PRESENTASI VIDEO

5.1 Link Presentasi Video

<https://drive.google.com/file/d/1i1D1chywLl49uodBkRbApedJp-BwS3oN/view?usp=sharing>

5.2 Link Colab

https://colab.research.google.com/drive/1TyO4qFW8zSNmMsWuKdfNy1vELY_CqTTI?usp=sharing

5.3 Link Github

<https://github.com/raflidev/Machine-Learning-Case-Base-1>