

LAPORAN
PEMBELAJARAN MESIN



Telkom
University

Oleh :

Caesar Fannany (IF-44-11 / 1301204144)

Dosen Pengampu:

Tjokorda Agung Budi Wirayuda, S.T., M.T.

PRODI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2021

Kata Pengantar

Dengan mengucapkan puji dan rasa syukur kepada Allah SWT yang telah memberikan rahmat sehingga kita dapat menyelesaikan tugas dari mata kuliah Machine Learning dengan tema “Cased Based-1” dengan benar dan tepat waktu.

Untuk memenuhi nilai tugas pada mata kuliah Pembelajaran Mesin, maka dibuatkan tugas yang dapat saya selesaikan. Tidak hanya itu, tujuan dari pembuatan laporan dan pengerjaan tugas ini adalah untuk menambah wawasan tentang pembahasan Implementasi ANN bagi kita semua.

Saya mengucapkan terima kasih kepada semua pihak mulai dari Pak Tjokorda Agung Budi Wirayuda, S.T., M.T. selaku dosen pembimbing yang telah memberikan saya tugas besar untuk membuat Algoritma ANN.

Saya sangat menyadari laporan yang saya susun masih jauh dari kata sempurna. Tetapi saya akan terus berusaha untuk selalu menjadi lebih baik untuk kedepannya.

Pernyataan : Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi.

Bandung, 9 November 2022

Caesar Fannany

BAB I PENDAHULUAN

Scenario

Mengikuti keberhasilan tugas sebelumnya, Anda diberi kesempatan lebih lanjut untuk mengesankan atasan Anda mengenai kemampuan Anda untuk menganalisis data. Anda diminta untuk melakukan beberapa analisis dan menghasilkan seperangkat aturan yang berguna menggunakan dataset berikut:

Kumpulan data berikut tersedia online, tautan ke kumpulan data adalah sebagai berikut:

[untuk NIM akhir GENAP gunakan data ini] <https://archive.ics.uci.edu/ml/datasets/Arrhythmia>

[untuk NIM akhir GANJIL gunakan data ini] <https://archive.ics.uci.edu/ml/datasets/Audit+Data>

Anda perlu mempelajari data dengan hati-hati. Kemudian pilih teknik pra-pemrosesan data apa yang akan dilakukan untuk meningkatkan kualitas data tersebut. Akan ada banyak hal yang harus diuraikan dan kemudian Anda harus mengumpulkan case-based ini sebagai karya individu.

Tugas Anda

Tujuan dari tugas ini yaitu Anda diharapkan mampu menjelaskan, mengimplementasikan, menganalisis, dan mendesain teknik pembelajaran mesin supervised learning yaitu AAN/MLP/RNN/LSTM/CNN.

Pertama, selidiki masalah kualitas data yang telah diberikan di atas. Jelaskan keputusan Anda mengenai pendekatan pra-pemrosesan data. Jelajahi kumpulan data dengan meringkas data menggunakan statistik dan mengidentifikasi masalah kualitas data apa pun. Tidak ada batasan jumlah ringkasan yang akan dilaporkan tetapi Anda diharapkan hanya melaporkan yang paling relevan.

Kedua, pilih salah satu dari metode unsupervised learning yang telah dipelajari yaitu AAN/MLP/RNN/LSTM/CNN. Anda hanya perlu memilih satu metode untuk diterapkan. Gunakan algoritma tersebut untuk memberikan beberapa output/outcome dengan menggunakan variasi hyperparameter, kemudian menganalisis hasilnya.

BAB II PEMBAHASAN

1. Ikhtisar Data

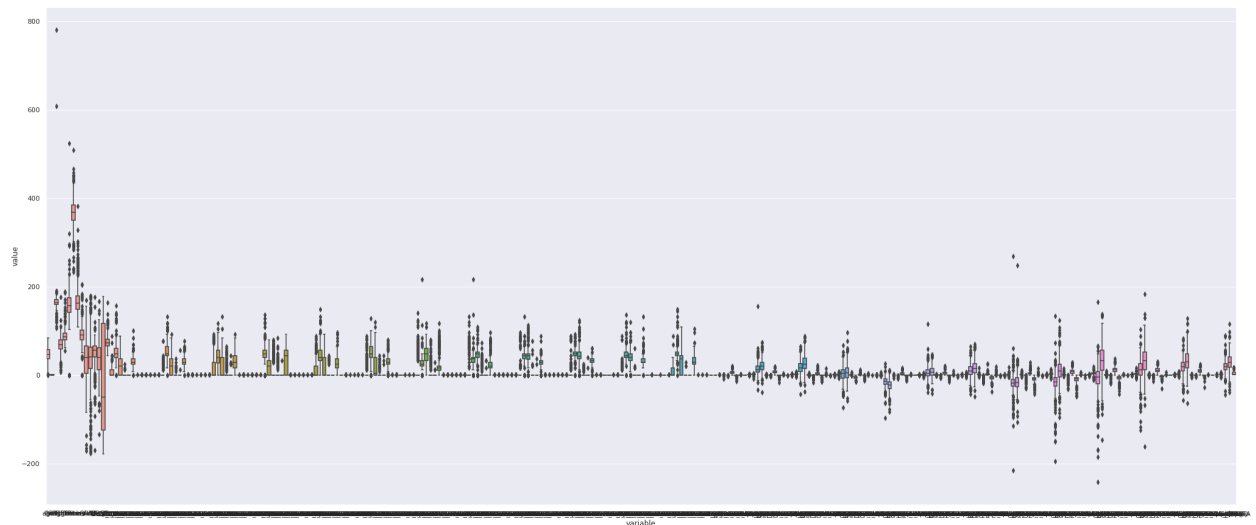
Data yang digunakan pada kasus ini merupakan data arrhythmia dikarenakan saya mendapatkan nim berakhiran 4.

```
#load the dataset
df = pd.read_csv('https://raw.githubusercontent.com/CaesarFannany/casedBased1/main/Data.csv')
```

Bagian ini merupakan input data dari github menuju google colab.

| | age | sex | height | weight | QRSduration | PRinterval | Q-Tinterval | Interval | Pinterval | QRS | ... | chV6_QwaveAmp | chV6_RwaveAmp | chV6_SwaveAmp | chV6_RPwaveAmp | chV6_SPwave |
|-----|-----|-----|--------|--------|-------------|------------|-------------|----------|-----------|-----|-----|---------------|---------------|---------------|----------------|-------------|
| 0 | 75 | 0 | 190 | 80 | 91 | 193 | 371 | 174 | 121 | -16 | ... | 0.0 | 9.0 | -0.9 | 0.0 | |
| 1 | 56 | 1 | 165 | 64 | 81 | 174 | 401 | 149 | 39 | 25 | ... | 0.0 | 8.5 | 0.0 | 0.0 | |
| 2 | 54 | 0 | 172 | 95 | 138 | 163 | 386 | 185 | 102 | 96 | ... | 0.0 | 9.5 | -2.4 | 0.0 | |
| 3 | 55 | 0 | 175 | 94 | 100 | 202 | 380 | 179 | 143 | 28 | ... | 0.0 | 12.2 | -2.2 | 0.0 | |
| 4 | 75 | 0 | 190 | 80 | 88 | 181 | 360 | 177 | 103 | -16 | ... | 0.0 | 13.1 | -3.6 | 0.0 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 447 | 53 | 1 | 160 | 70 | 80 | 199 | 382 | 154 | 117 | -37 | ... | 0.0 | 4.3 | -5.0 | 0.0 | |
| 448 | 37 | 0 | 190 | 85 | 100 | 137 | 361 | 201 | 73 | 86 | ... | 0.0 | 15.6 | -1.6 | 0.0 | |
| 449 | 36 | 0 | 166 | 68 | 108 | 176 | 365 | 194 | 116 | -85 | ... | 0.0 | 16.3 | -28.6 | 0.0 | |
| 450 | 32 | 1 | 155 | 55 | 93 | 106 | 386 | 218 | 63 | 54 | ... | -0.4 | 12.0 | -0.7 | 0.0 | |
| 451 | 78 | 1 | 160 | 70 | 79 | 127 | 364 | 138 | 78 | 28 | ... | 0.0 | 10.4 | -1.8 | 0.0 | |

Data berupa data supervised data yang sudah dikelompokkan pada data class. Dengan begitu kita dapat menentukan dengan mencari data dengan orang yang terkena penyakit arrhythmia. Data tersebut tidak memiliki data trial, data yang ditampilkan merupakan data detak jantung dari setiap pasien, dengan begitu data pada kolom sangat bervariasi.



2. Pra-pemrosesan data

```
df = df.replace('?', np.NaN)
```

Data yang sudah dibagi lalu pada “df” yang memiliki “?” atau tanda tanya akan diisi dengan NaN dengan Not A Number dengan begitu dapat terlihat perbedaan data yang ada.

```
thresh = len(df) * 0.5  
df.dropna(thresh = thresh, axis = 1, inplace = True)
```

Data yang sudah diganti dengan Not A Number, selanjutnya pada data axis = 1 atau, pada data kolom yang memiliki kekosongan data sampai dengan 50% akan dihapus dari daftar data set tersebut.

```
df = df.fillna(df.median())
```

Data yang sudah dihapus sebelumnya lanjut pada tahap pengisian data yang Not A Number diganti dengan median dari data-data kolom tersebut.

```
std_scaler = StandardScaler()  
x_scaled = std_scaler.fit_transform(df_data.values)  
df_data = pd.DataFrame(x_scaled, index = df_data.index)
```

Setelah data yang diisi dengan data median setelah itu menggunakan StandardScaler agar data bisa lebih terkontrol batas atas dengan 1 dan batas bawah dengan 0.

```
data yang sudah di kurangi  
(452, 279)
```

```
y = df["class"]  
x = df.drop(['class'], axis=1)  
x = np.array(x)  
y = np.array(y)  
x[:10], y[:10]
```

Data yang sudah diproses sebelumnya lalu data tersebut dipisah dengan data x dengan data y, dimana data x merupakan data yang sudah mendrop data class lalu pada data y disimpan kedalam bentuk array begitu juga sebaliknya data x disimpan kedalam array.

```
from sklearn.preprocessing import LabelEncoder  
from keras.utils.np_utils import to_categorical  
  
l_encode = LabelEncoder()  
l_encode.fit(y)  
y = l_encode.transform(y)  
y = to_categorical(y)  
y
```

Data yang sudah dipisah dengan data x dan data y, pada data y dijadikan encode agar hasilnya hanya 1 atau 0 saja.

```
xtrain, xval, ytrain, yval = train_test_split(x, y, test_size=0.4, random_state=0)
xtrain.shape, ytrain.shape, xval.shape, yval.shape
```

Data dibagi menjadi dua bagian yaitu bagian data train dengan data validasi, dengan pembagian data train sebanyak 40% dengan data validasi 60%, data yang sudah dipisah tidak di random pada kasus ini.

3. Menerapkan algoritma

Pada kasus ini saya menggunakan algoritma ANN(Artificial Neural Network) untuk menyelesaikan permasalahan ini, algoritma ANN sendiri merupakan algoritma dengan pendekatan menggunakan matrix. Dengan dilakukannya percobaan hasil ini saya mengharapkan hasil sebesar diatas 0.70 atau diatas 70%.

```
from keras.models import Sequential
from keras.layers import Activation, Dense
model = Sequential()

model.add(Dense(8, input_dim = 278, activation = 'relu'))
model.add(Dense(512, activation = 'relu'))
model.add(Dense(256, activation = 'relu'))
model.add(Dense(128, activation = 'relu'))
model.add(Dense(13, activation = 'softmax'))
model.summary()
```

Metode yang digunakan menggunakan model sequential dengan input hyperparameter dengan input neuron dan hidden layer yang lainnya, dan menggunakan aktivasi yang bisa diubah ubah.

```
model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
hist = model.fit(xtrain, ytrain, epochs=200, batch_size=8)
```

Pada compile modelnya menggunakan loss “categorical_crossentropy” dengan optimizer “adam”, dengan hyperparameter epochs dan batch_size.

| Metode | Hasil Epoch = 100 |
|---|-------------------|
| <pre>from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'relu')) model.add(Dense(256, activation = 'relu')) model.add(Dense(128, activation = 'relu')) model.add(Dense(13, activation = 'softmax')) model.summary()</pre> | 0.9889 |

| | |
|---|--------|
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'sigmoid')) model.add(Dense(256, activation = 'relu')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.9410 |
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'sigmoid')) model.add(Dense(512, activation = 'sigmoid')) model.add(Dense(256, activation = 'sigmoid')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.6863 |
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'sigmoid')) model.add(Dense(256, activation = 'tanh')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.9151 |
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'relu')) model.add(Dense(256, activation = 'tanh')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.8007 |

4. Evaluasi

Random data = 5

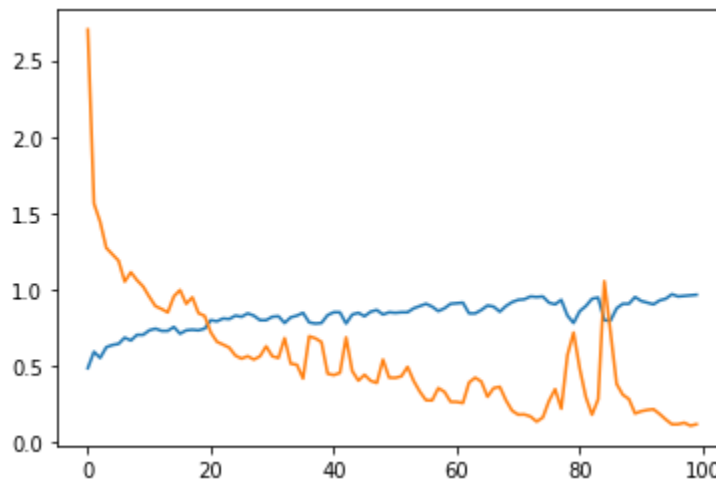
| Model | Hasil Epoch = 100 | Hasil Validasi |
|-------|----------------------|----------------|
|-------|----------------------|----------------|

| | | |
|---|--------|--------|
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'relu')) model.add(Dense(256, activation = 'relu')) model.add(Dense(128, activation = 'relu')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.8819 | 0.4972 |
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'relu')) model.add(Dense(512, activation = 'sigmoid')) model.add(Dense(256, activation = 'relu')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.9188 | 0.5856 |
| <pre> from keras.models import Sequential from keras.layers import Activation, Dense model = Sequential() model.add(Dense(8, input_dim = 278, activation = 'sigmoid')) model.add(Dense(512, activation = 'sigmoid')) model.add(Dense(256, activation = 'sigmoid')) model.add(Dense(128, activation = 'sigmoid')) model.add(Dense(13, activation = 'softmax')) model.summary() </pre> | 0.7306 | 0.6022 |

KESIMPULAN

Pada percobaan ini saya menggunakan algoritma ANN dalam menyelesaikan masalah arrhythmia, dengan begitu hasil yang didapatkan melampaui keinginan saya dengan hasil akurasi diatas 70% namun hasil validasinya dibawah dari hasil akurasi maka dalam hal ini terjadi overfitting pada data testnya, dengan begitu hal ini memungkinkan terjadi hasil yang lebih baik dengan menggunakan algoritma yang berbeda.

Algoritma ini saya menerapkan hyperparameter dimana epoch dan batch yang bisa dirubah, data random yang bisa diacak, lalu penggunaan loss dan parameter yang bisa dirubah. Hasil yang terbaik saya dapatkan ketika hidden layer menggunakan aktivasi “relu” dan output yang menggunakan aktivasi “softmax”, dengan hasil akurasi data test sebesar 0.9889 dan hasil validasi sebesar 0.5414.



Tabel tersebut menunjukkan hasil akurasi pada garis biru lalu loss yang terjadi pada garis orange. Hasil dari tabel tersebut menunjukkan bahwa akurasi disini meningkat sebanyak epoch yang dijalankan lalu hasil loss menurun sebanyak epoch yang dijalankan.

Link Colab : <https://colab.research.google.com/drive/1z5ZZde029cqgmbxxzq-Nj81WcaFt05SG?usp=sharing>

Link Video : https://youtu.be/S067xz_bOMw