HACETTEPE UNIVERSITY

BBM407

FUZZY LOGIC

ANFIS for Sepsis Disease

Gizem Aleyna Tuzcu

2200356816

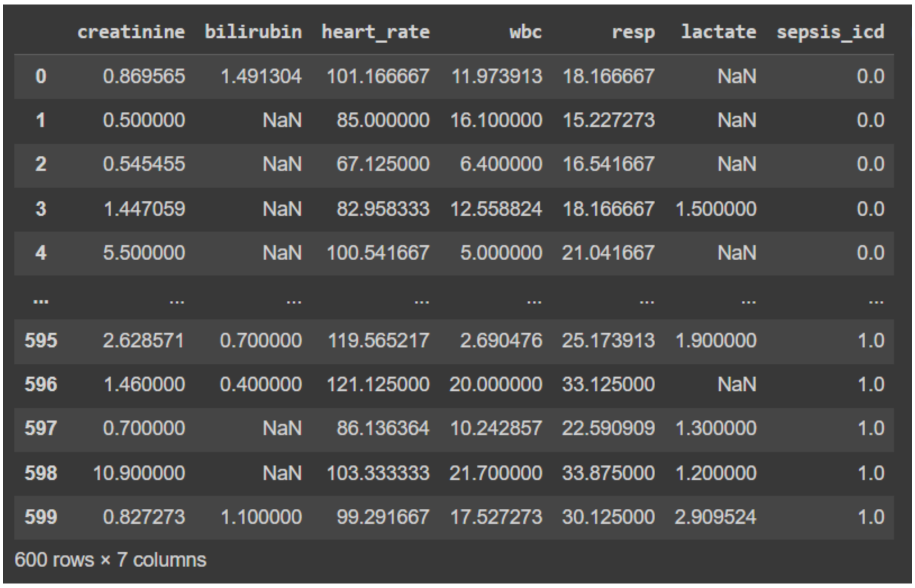
1. DATA SET

The data is taken by 300 patiens from both sepsis and non-sepsis patients. At total, there are 600 patients.

The chosen metrics for model are;

Creatinine, Bilirubin, Heart_rate, WBC, Resp, Lactate

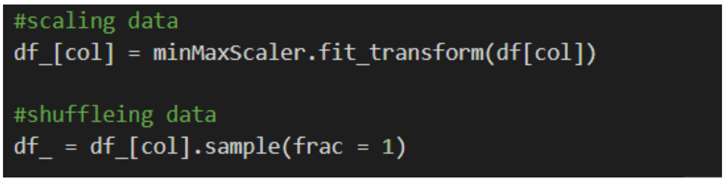
And the chosen metric for output is **Sepsis_icd**



	creatinine	bilirubin	heart_rate	wbc	resp	lactate	sepsis_icd
0	0.869565	1.491304	101.166667	11.973913	18.166667	NaN	0.0
1	0.500000	NaN	85.000000	16.100000	15.227273	NaN	0.0
2	0.545455	NaN	67.125000	6.400000	16.541667	NaN	0.0
3	1.447059	NaN	82.958333	12.558824	18.166667	1.500000	0.0
4	5.500000	NaN	100.541667	5.000000	21.041667	NaN	0.0
...	...	...	...	...	...	...	...
595	2.628571	0.700000	119.565217	2.690476	25.173913	1.900000	1.0
596	1.460000	0.400000	121.125000	20.000000	33.125000	NaN	1.0
597	0.700000	NaN	86.136364	10.242857	22.590909	1.300000	1.0
598	10.900000	NaN	103.333333	21.700000	33.875000	1.200000	1.0
599	0.827273	1.100000	99.291667	17.527273	30.125000	2.909524	1.0

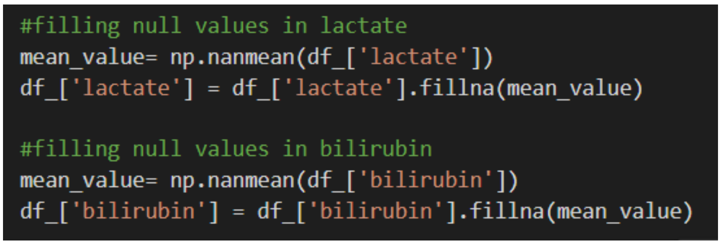
600 rows × 7 columns

After scaling data with minMaxScaler, it is shuffeled by fraction 1 to distribute patients uniformly as their sepsis output.



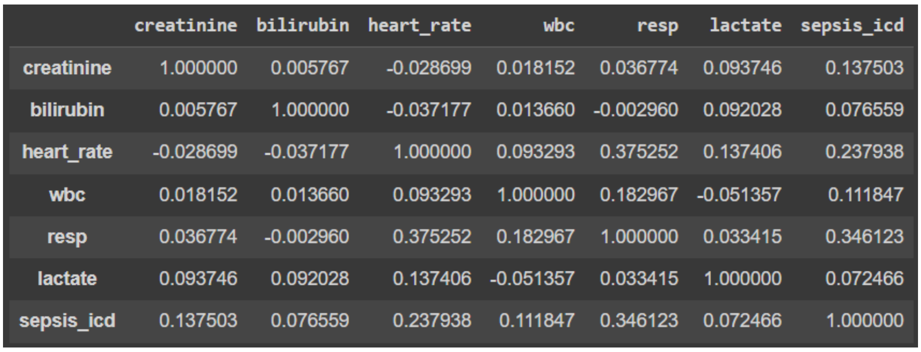
```
#scaling data
df_[col] = minMaxScaler.fit_transform(df[col])

#shuffleing data
df_ = df_[col].sample(frac = 1)
```

Since the columns *Lactate* and *Bilirubin* consists null values, they are filled with mean value.

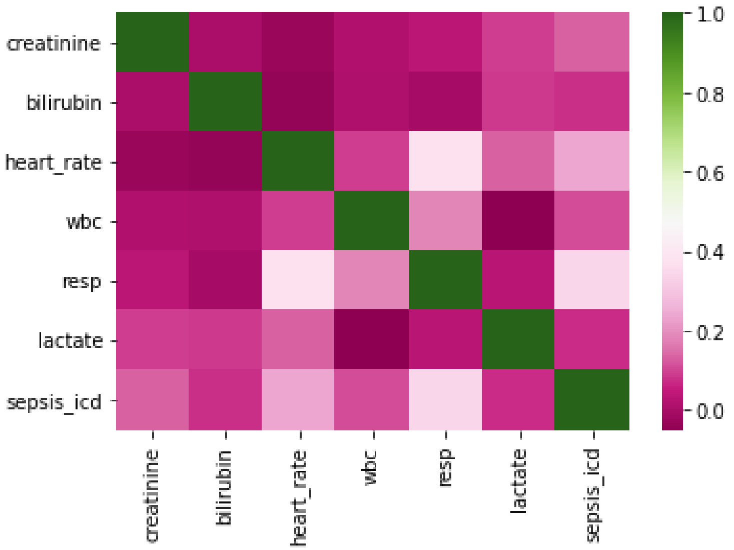


```
#filling null values in lactate
mean_value= np.nanmean(df_['lactate'])
df_['lactate'] = df_['lactate'].fillna(mean_value)

#filling null values in bilirubin
mean_value= np.nanmean(df_['bilirubin'])
df_['bilirubin'] = df_['bilirubin'].fillna(mean_value)
```

The Correlation Co-efficients



	creatinine	bilirubin	heart_rate	wbc	resp	lactate	sepsis_icd
creatinine	1.000000	0.005767	-0.028699	0.018152	0.036774	0.093746	0.137503
bilirubin	0.005767	1.000000	-0.037177	0.013660	-0.002960	0.092028	0.076559
heart_rate	-0.028699	-0.037177	1.000000	0.093293	0.375252	0.137406	0.237938
wbc	0.018152	0.013660	0.093293	1.000000	0.182967	-0.051357	0.111847
resp	0.036774	-0.002960	0.375252	0.182967	1.000000	0.033415	0.346123
lactate	0.093746	0.092028	0.137406	-0.051357	0.033415	1.000000	0.072466
sepsis_icd	0.137503	0.076559	0.237938	0.111847	0.346123	0.072466	1.000000



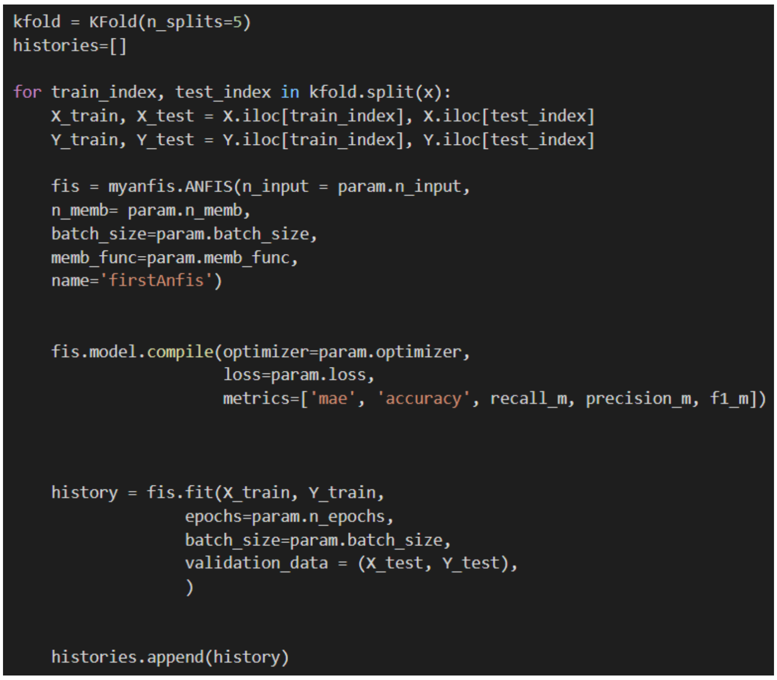
2. TRAIN – TEST INDEX

The data is splitted with 5 folds.

0 – 120	120 – 240	240 – 360	360 – 480	480 - 600
■				
	■			
		■		
			■	
				■

Train dataset is 480 patients, and test index is 120 patients.
At each iteration, test index shifted by 120 patients. By doing that, every patient involved in both test and train datasets.

3. ANFIS

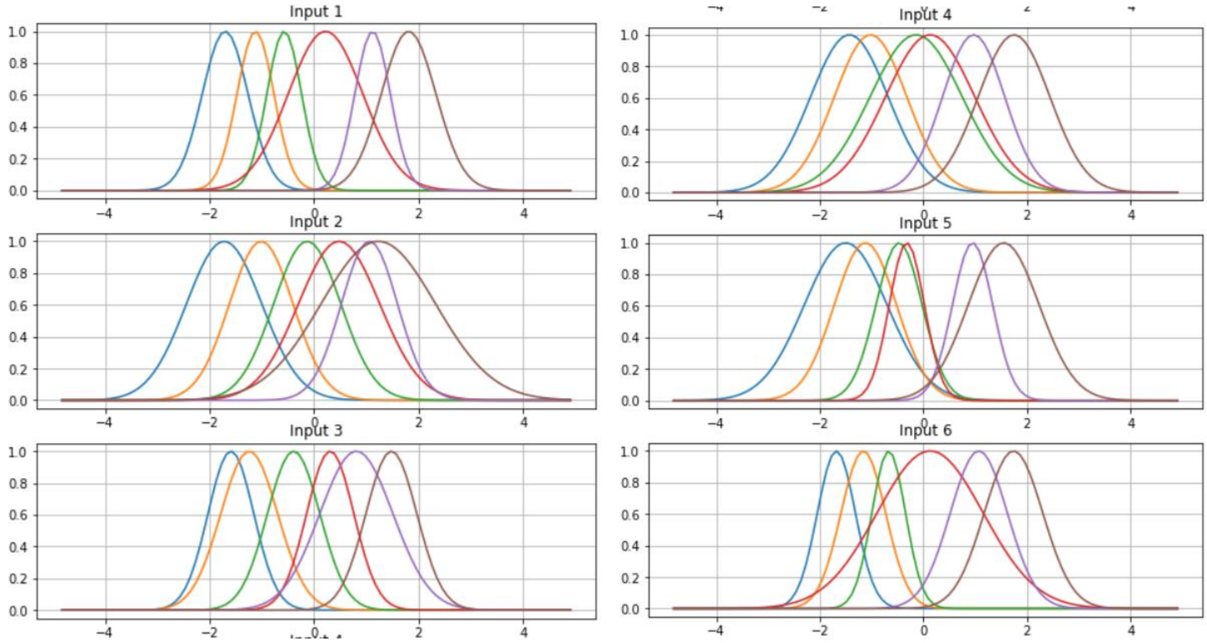


```
kfold = KFold(n_splits=5)
histories=[]

for train_index, test_index in kfold.split(x):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    Y_train, Y_test = Y.iloc[train_index], Y.iloc[test_index]

    fis = myanfis.ANFIS(n_input = param.n_input,
    n_memb= param.n_memb,
    batch_size=param.batch_size,
    memb_func=param.memb_func,
    name='firstAnfis')


    fis.model.compile(optimizer=param.optimizer,
                      loss=param.loss,
                      metrics=['mae', 'accuracy', recall_m, precision_m, f1_m])



    history = fis.fit(X_train, Y_train,
                 epochs=param.n_epochs,
                 batch_size=param.batch_size,
                 validation_data = (X_test, Y_test),
                 )


    histories.append(history)
```

In ANFIS model, used membership function is *Gaussian*, and optimizer type is *Adam*.

Number of splits in k-fold is 5 and membership function for every 6 input is divided by 6.

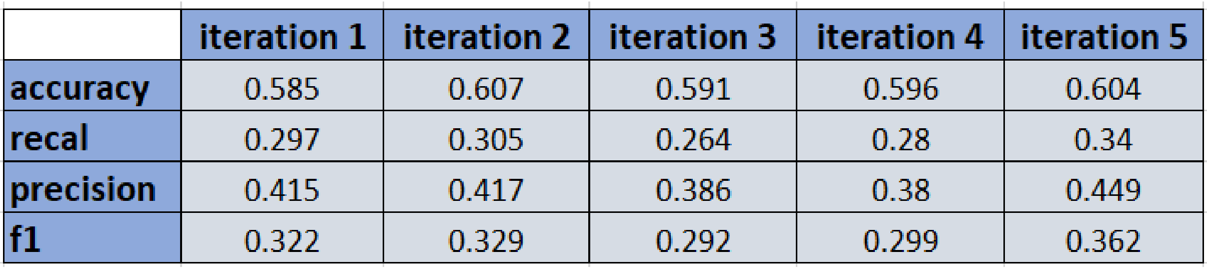
4. MEMBERSHIP FUNCTIONS



Membership functions are calculated by the model itself.

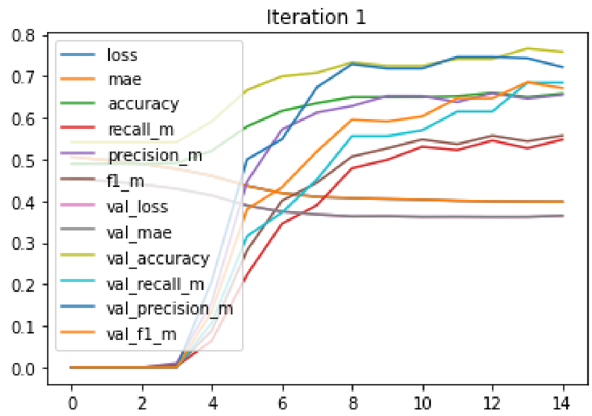
5. SCORES FOR EACH ITERATION

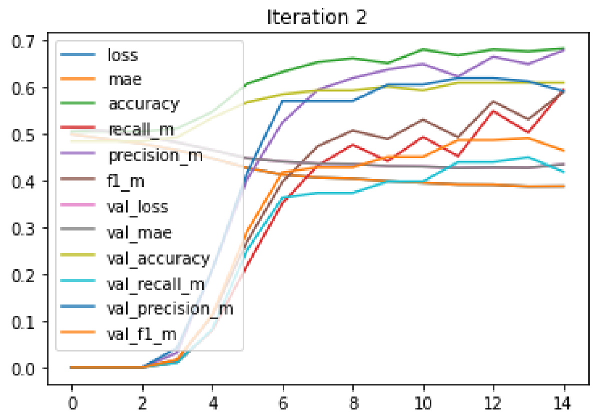
Average scores for each iteration:

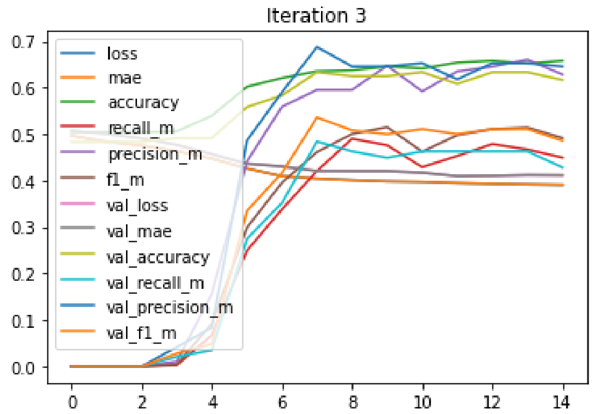


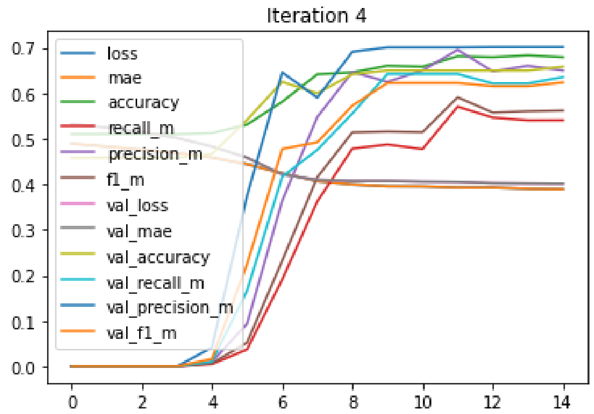
	iteration 1	iteration 2	iteration 3	iteration 4	iteration 5
accuracy	0.585	0.607	0.591	0.596	0.604
recal	0.297	0.305	0.264	0.28	0.34
precision	0.415	0.417	0.386	0.38	0.449
f1	0.322	0.329	0.292	0.299	0.362

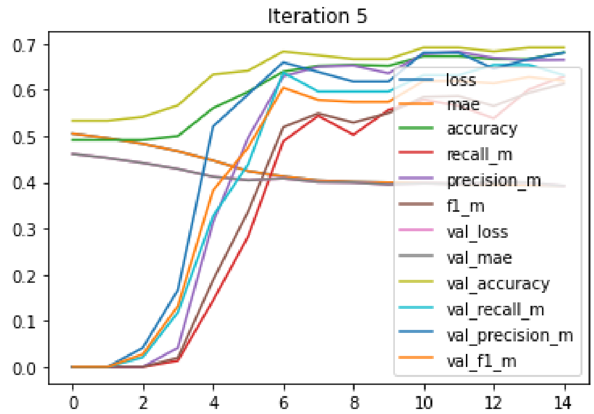
Plots for each iteration:



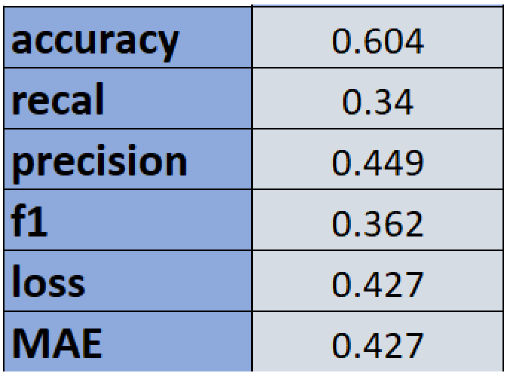








6. RESULTING SCORES



accuracy	0.604
recal	0.34
precision	0.449
f1	0.362
loss	0.427
MAE	0.427

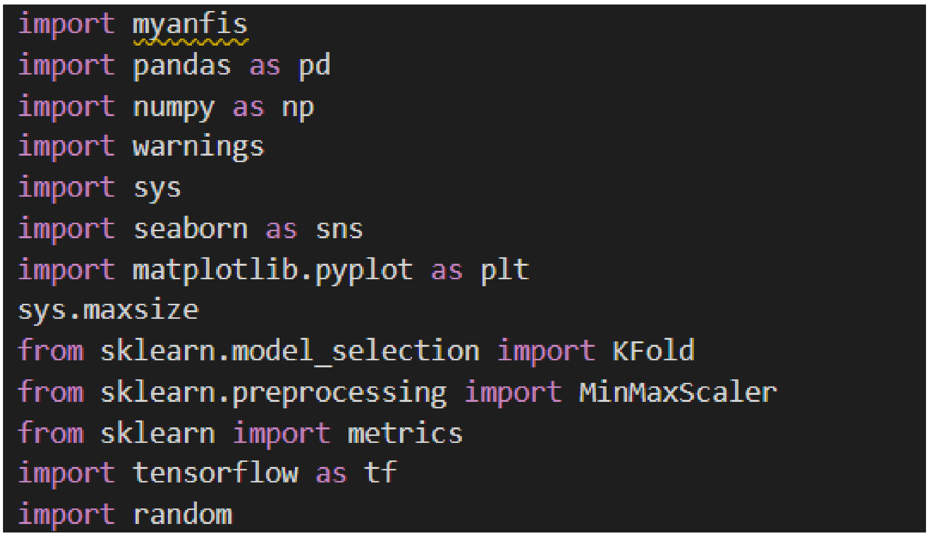
After all 5 iterations, average scores are shown above.

- Max accuracy for itearion 1 is 0.64
- Max accuracy for itearion 2 is 0.67
- Max accuracy for itearion 3 is 0.65
- Max accuracy for itearion 4 is 0.68
- Max accuracy for itearion 5 is 0.68

Comparing the results from “Midterm Project Mamdani FIS” with the results from “Final Project: ANFIS”, there is a significant difference between two models. Accuracy score taken from Mamdani FIS project was **0.59**. But the accuracy score taken from ANFIS is somewhere between **0.55-0.68**

One of the reason for this improvement is membership functions are created by model itself, and neurol network is created automatically. By doing so, human error is eliminated.

7. DEPENDENCIES



```
import myanfis
import pandas as pd
import numpy as np
import warnings
import sys
import seaborn as sns
import matplotlib.pyplot as plt
sys.maxsize
from sklearn.model_selection import KFold
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
import tensorflow as tf
import random
```

NOTES:

I created my project on Google Colab platform. That’s why reading file from drive required different file reading.