



Fuzzy Logic Homework4

Covid detection rule-base system

Instructed by Dr. Tahayori

Reza Tahmasebi

saeed77t@gmail.com, Stu ID: 40160957

Table of Contents

1	<i>About the project</i>	3
2	<i>The project flow</i>	3
2.1	Feature completion and extraction	3
2.2	Feature fuzzification	5
2.3	Extract rules from the dataset.....	7
2.4	Use rule-based system.....	8
2.5	Predict the results.....	8
3	<i>Results</i>	9
4	<i>Type2 fuzzy systems</i>	9
4.1	Rules	13

1 About the project

In this project, we have a dataset of 250 people for a blood test, and each row of our data set has 14 features; this project aims to detect covid infection based on blood test results.

2 The project flow

The project contains a few steps: feature completion and extraction, making fuzzy sets, extracting rules, using a rule system, predict the results.

2.1 Feature completion and extraction

Our data set has many empty spaces; the *KNNImputer* method fills the dataset with free slots.

	SESSO	AGE	WBC	Piastrine	Neutrofili	Linfociti	Monociti	Eosinofili	Basofili	PCR	AST	ALT	ALP	GGT	LDH	TARGET
0	0.0	56	290.0	128.0	1.9	80.0	20.0	0.0	0.0	29.0	36.0	18.0	43.0	21.0	257.0	1
1	0.0	56	350.0	151.0	2.1	90.0	40.0	0.0	0.0	16.5	25.0	14.0	50.0	17.0	207.0	1
2	0.0	72	460.0	206.0	NaN	NaN	NaN	NaN	NaN	193.7	31.0	22.0	NaN	NaN	NaN	1
3	0.0	72	1650.0	316.0	14.0	120.0	30.0	0.0	0.0	318.7	96.0	33.0	80.0	42.0	651.0	1
4	0.0	77	490.0	198.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1

Example of the dataset before imputation

	SESSO	AGE	WBC	Piastrine	Neutrofili	Linfociti	Monociti	Eosinofili	Basofili	PCR	AST	ALT	ALP	GGT	LDH	TARGET
0	0.0	56.0	290.0	128.0	1.900	80.0	20.0	0.0	0.000	29.0	36.00	18.0	43.00	21.00	257.00	1.0
1	0.0	56.0	350.0	151.0	2.100	90.0	40.0	0.0	0.000	16.5	25.00	14.0	50.00	17.00	207.00	1.0
2	0.0	72.0	460.0	206.0	3.500	55.0	42.5	0.0	0.000	193.7	31.00	22.0	63.50	23.25	384.75	1.0
3	0.0	72.0	1650.0	316.0	14.000	120.0	30.0	0.0	0.000	318.7	96.00	33.0	80.00	42.00	651.00	1.0
4	0.0	77.0	490.0	198.0	3.775	67.5	42.5	0.0	0.000	34.0	35.75	24.5	85.50	91.00	330.00	1.0

Example of the dataset after imputation

Because we had 14 features and some of them were not important for covid detection, we used two types of correlation to reduce the number of features,

1- correlation point biserial model

2- normal correlation

```

corolation wbc = -0.19315379527313117 ||||| pval wbc = 0.0011852460064474524
corolation piastrine = -0.045659001525092766 ||||| pval piastrine = 0.4474769964029412
corolaion noeutrofill = -0.1285552758421598 ||||| pval noeutrofill = 0.031827477406032416
corolation Linfociti = -0.3150368992014245 ||||| pval Linfociti = 7.614108248719836e-08
corolation Monociti = -0.27498450409444375 ||||| pval Monociti = 3.120507342842616e-06
corolation Eosinofili = -0.2040137843659165 ||||| pval Eosinofili = 0.000606854725026091
corolation Basofili = -0.12819360199132807 ||||| pval Basofili = 0.032315901509511594
#####corolation PCR = 0.21622276584480724 ||||| pval PCR = 0.00027417405438361097
#####corolation AST = 0.21212250982505995 ||||| pval AST = 0.0003598195755476327
#####corolation ALT = 0.20258515992213294 ||||| pval ALT = 0.0006640400278052779
#####corolation ALP = -0.03983363355233881 ||||| pval ALP = 0.5075683393674252
#####corolation GGT = 0.11268937118077087 ||||| pval GGT = 0.06013193690173554
#####corolation LDH = 0.34667032615384424 ||||| pval LDH = 2.6802829229794086e-09
#####corolation AGE = 0.08764175995129661 ||||| pval AGE = 0.14425029798375155

```

Correlation point biserial for each feature and target

	SESSO	AGE	WBC	Piastrine	Neutrofilli	Linfociti	Monociti	Eosinofili	Basofili	PCR	AST	ALT	ALP	GGT	LDH	TARGET
SESSO	1.00	-0.04	-0.04	0.06	-0.08	0.20	0.05	0.09	0.03	-0.27	-0.14	-0.11	0.07	-0.07	-0.12	-0.25
AGE	-0.04	1.00	0.10	-0.17	0.14	-0.21	0.08	-0.09	-0.01	0.11	-0.00	-0.08	-0.04	-0.03	0.16	0.09
WBC	-0.04	0.10	1.00	0.41	0.96	0.16	0.66	0.18	0.54	0.26	0.17	0.07	0.14	0.09	0.35	-0.19
Piastrine	0.06	-0.17	0.41	1.00	0.36	0.14	0.31	0.25	0.31	0.09	0.03	0.15	0.18	0.20	0.13	-0.05
Neutrofilli	-0.08	0.14	0.96	0.36	1.00	-0.04	0.55	0.09	0.49	0.33	0.21	0.10	0.11	0.10	0.39	-0.13
Linfociti	0.20	-0.21	0.16	0.14	-0.04	1.00	0.32	0.26	0.24	-0.33	-0.18	-0.15	0.06	-0.07	-0.25	-0.32
Monociti	0.05	0.08	0.66	0.31	0.55	0.32	1.00	0.24	0.44	-0.04	-0.07	-0.04	0.12	-0.02	-0.03	-0.27
Eosinofili	0.09	-0.09	0.18	0.25	0.09	0.26	0.24	1.00	0.42	-0.03	0.04	0.03	0.12	0.04	-0.05	-0.20
Basofili	0.03	-0.01	0.54	0.31	0.49	0.24	0.44	0.42	1.00	0.01	0.17	0.13	0.03	0.01	0.12	-0.13
PCR	-0.27	0.11	0.26	0.09	0.33	-0.33	-0.04	-0.03	0.01	1.00	0.29	0.10	0.01	0.10	0.50	0.22
AST	-0.14	-0.00	0.17	0.03	0.21	-0.18	-0.07	0.04	0.17	0.29	1.00	0.71	0.31	0.51	0.55	0.21
ALT	-0.11	-0.08	0.07	0.15	0.10	-0.15	-0.04	0.03	0.13	0.10	0.71	1.00	0.26	0.51	0.31	0.20
ALP	0.07	-0.04	0.14	0.18	0.11	0.06	0.12	0.12	0.03	0.01	0.31	0.26	1.00	0.80	0.09	-0.04
GGT	-0.07	-0.03	0.09	0.20	0.10	-0.07	-0.02	0.04	0.01	0.10	0.51	0.51	0.80	1.00	0.25	0.11
LDH	-0.12	0.16	0.35	0.13	0.39	-0.25	-0.03	-0.05	0.12	0.50	0.55	0.31	0.09	0.25	1.00	0.35
TARGET	-0.25	0.09	-0.19	-0.05	-0.13	-0.32	-0.27	-0.20	-0.13	0.22	0.21	0.20	-0.04	0.11	0.35	1.00

Normal correlation for each feature and target

After checking the correlations, seven features out of fourteen were chosen. ‘WBC,’ ‘Linfociti,’ ‘Monociti’ , ‘Eosinofili’, ‘AST’, ‘ALT’ and ‘LDH’

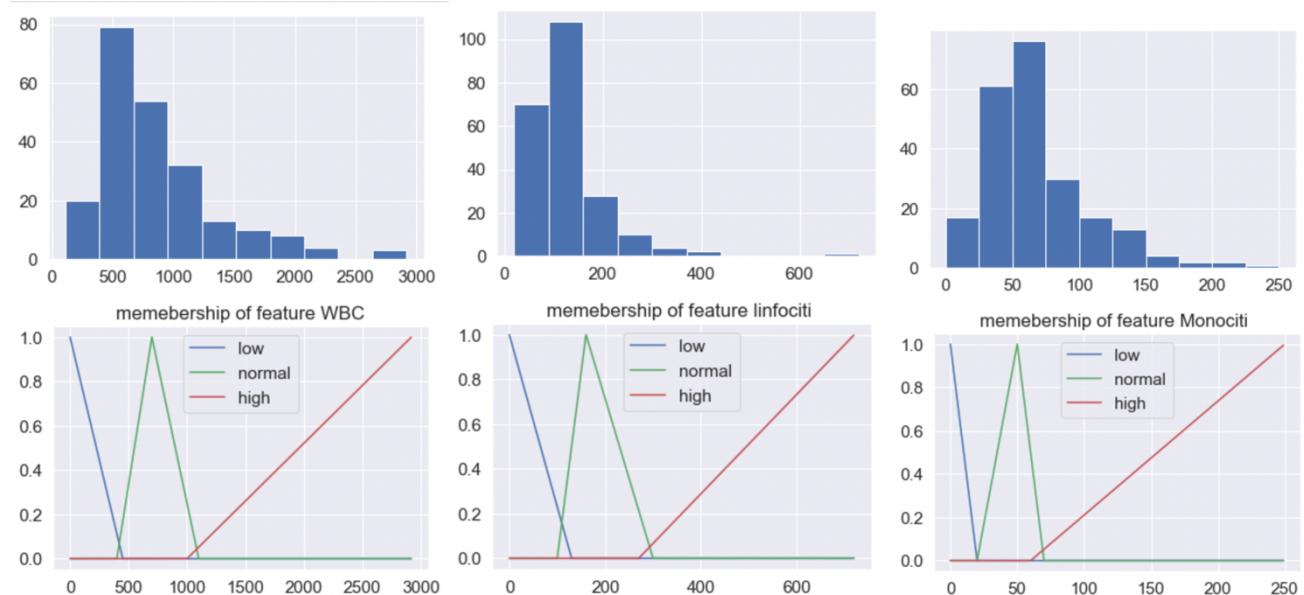
2.2 Feature fuzzification

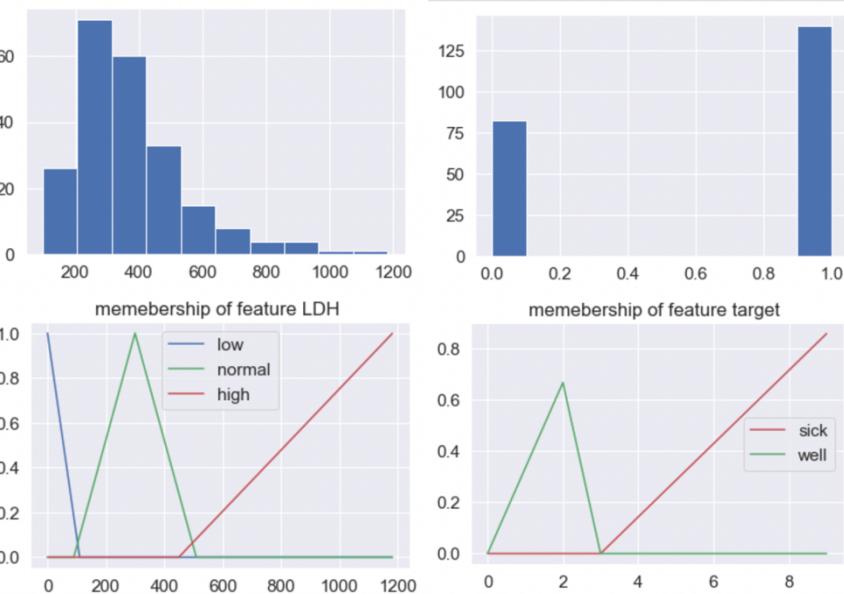
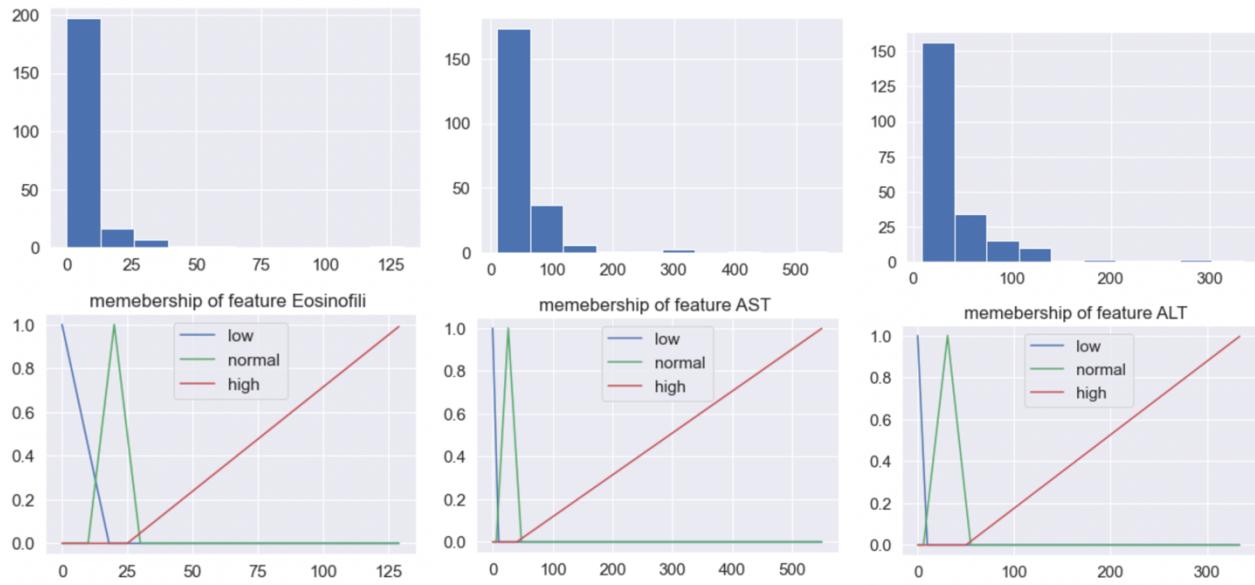
To create our fuzzy sets from each feature, first, we calculate the domain of each feature; then, with the help of the *skfuzzy* library, we define each feature as our antecedent. With the *trimf* method from the library, we create the triangulated fuzzy sets.

The histogram of each feature and an expert opinion helped to understand each feature's ‘normal,’ ‘low,’ and ‘high’ range.

The plots below show each image's histogram and fuzzy sets.

The *Consequent* method from the *skfuzzy* library was used to fuzzy-fie the labels.





Features and target fuzzy sets

2.3 Extract rules from the dataset.

We used all of our training data to extract rules; we used the *interp_membership* method from the library to calculate the degree of each feature in low, normal, and high ranges of its fuzzy set, then we used the *argmax* method to define each feature of the training data sets belongs to what range.

The output of the cycle is like the figure below.

```
[ [2, 1, 2, 0, 2, 2, 2, 1.0],
  [1, 1, 2, 1, 1, 1, 1, 1.0],
  [2, 1, 0, 0, 1, 2, 1, 0.0],
  [2, 0, 2, 0, 1, 1, 1, 0.0],
  [1, 0, 1, 0, 2, 1, 1, 1.0],
  [1, 1, 1, 0, 2, 2, 1, 1.0],
  [1, 2, 1, 2, 1, 1, 1, 0.0],
  [1, 1, 2, 0, 1, 1, 1, 1.0],
  [1, 0, 1, 0, 1, 1, 1, 1.0],
  [1, 0, 1, 1, 2, 2, 2, 1.0],
  [1, 0, 1, 0, 1, 1, 1, 0.0],
```

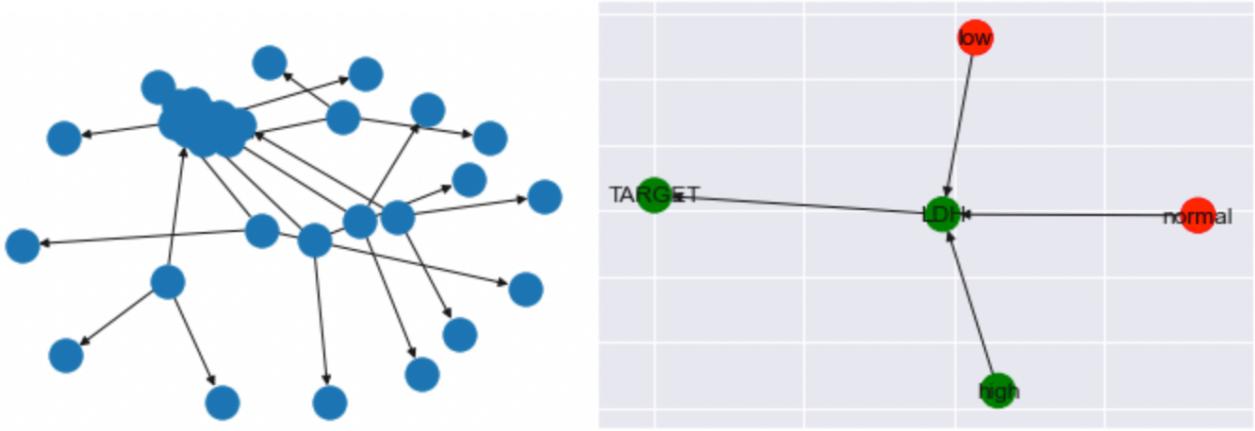
The output of the rule extraction cycle

With the *defineRulec* function, we replace each of the numbers (0, 1, 2) with ‘low,’ ‘normal,’ and ‘high,’ respectively.

And we use the *setrules* function to create the rules in a way that we can use in our system.

```
[IF (((((WBC[high] AND Linfociti[normal]) AND Monociti[high]) AND Eosinofili[low]) AND AST[high]) AND ALT[high]) AND LDH[high] THEN TARGET[sick]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[normal] AND Linfociti[normal]) AND Monociti[high]) AND Eosinofili[normal]) AND AST[normal]) AND ALT[normal]) AND LDH[normal] THEN TARGET[sick]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[high] AND Linfociti[normal]) AND Monociti[low]) AND Eosinofili[low]) AND AST[normal]) AND ALT[high]) AND LDH[normal] THEN TARGET[well]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[high] AND Linfociti[low]) AND Monociti[high]) AND Eosinofili[low]) AND AST[normal]) AND ALT[normal]) AND LDH[normal] THEN TARGET[well]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[normal] AND Linfociti[low]) AND Monociti[normal]) AND Eosinofili[low]) AND AST[high]) AND ALT[normal]) AND LDH[normal] THEN TARGET[sick]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[normal] AND Linfociti[normal]) AND Monociti[normal]) AND Eosinofili[low]) AND AST[high]) AND ALT[high]) AND LDH[normal] THEN TARGET[sick]
    AND aggregation function : fmin
    OR aggregation function : fmax,
IF (((((WBC[normal] AND Linfociti[high]) AND Monociti[normal]) AND Eosinofili[high]) AND AST[normal]) AND ALT[normal]) AND LDH[normal] THEN TARGET[well]
```

Example of extracted rules



A graph view of a single rule from two different sides.

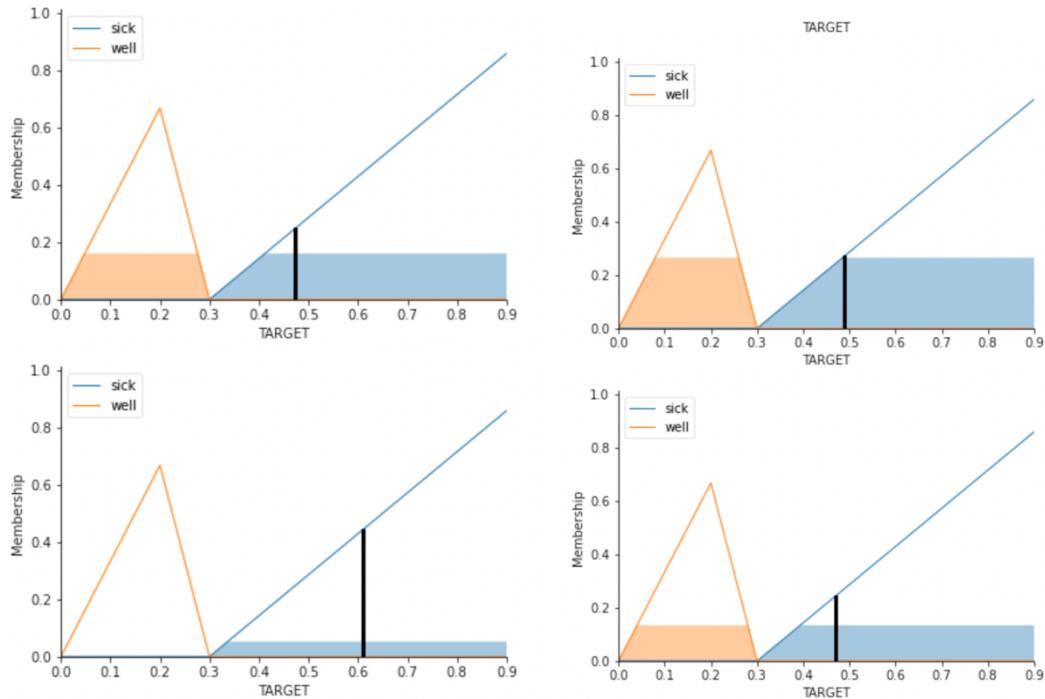
2.4 Use rule-based system

The rules created in the previous step were passed into the *ControlSystem* method from the library, got the result back, and then given into the *ControlSystemSimulation* method, which returns the simulations.

2.5 Predict the results

With a for loop on our test dataset, we set our inputs of the test dataset equal to our model features and

With *compute* method, we got back our results.



Prediction plots

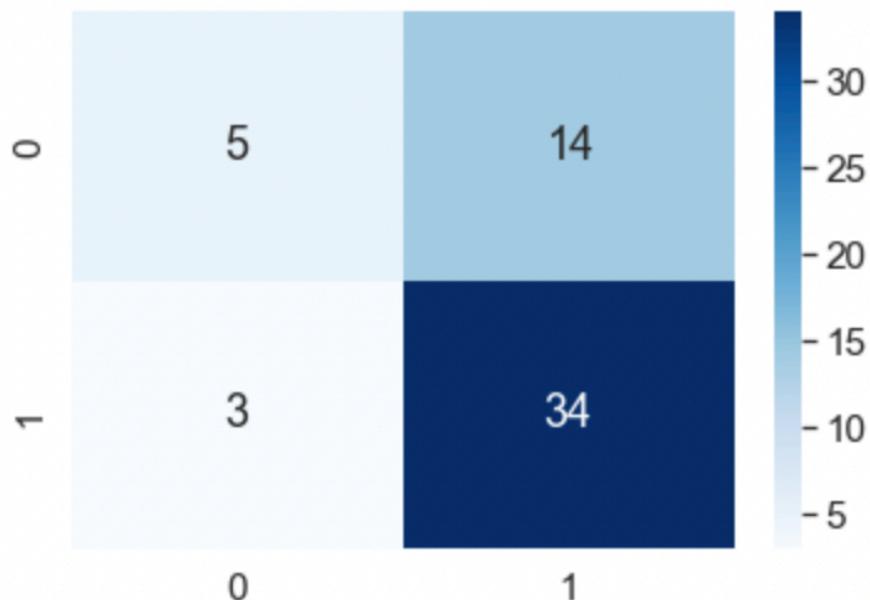
And after that, with the *prediction* method, we sort and organize the output of the fuzzy system, and then with the *accuracy_score function*, calculate the system's accuracy.

3 Results

The accuracy of our model is about 60% to 70%.

The latest run gave an accuracy of 69.62%.

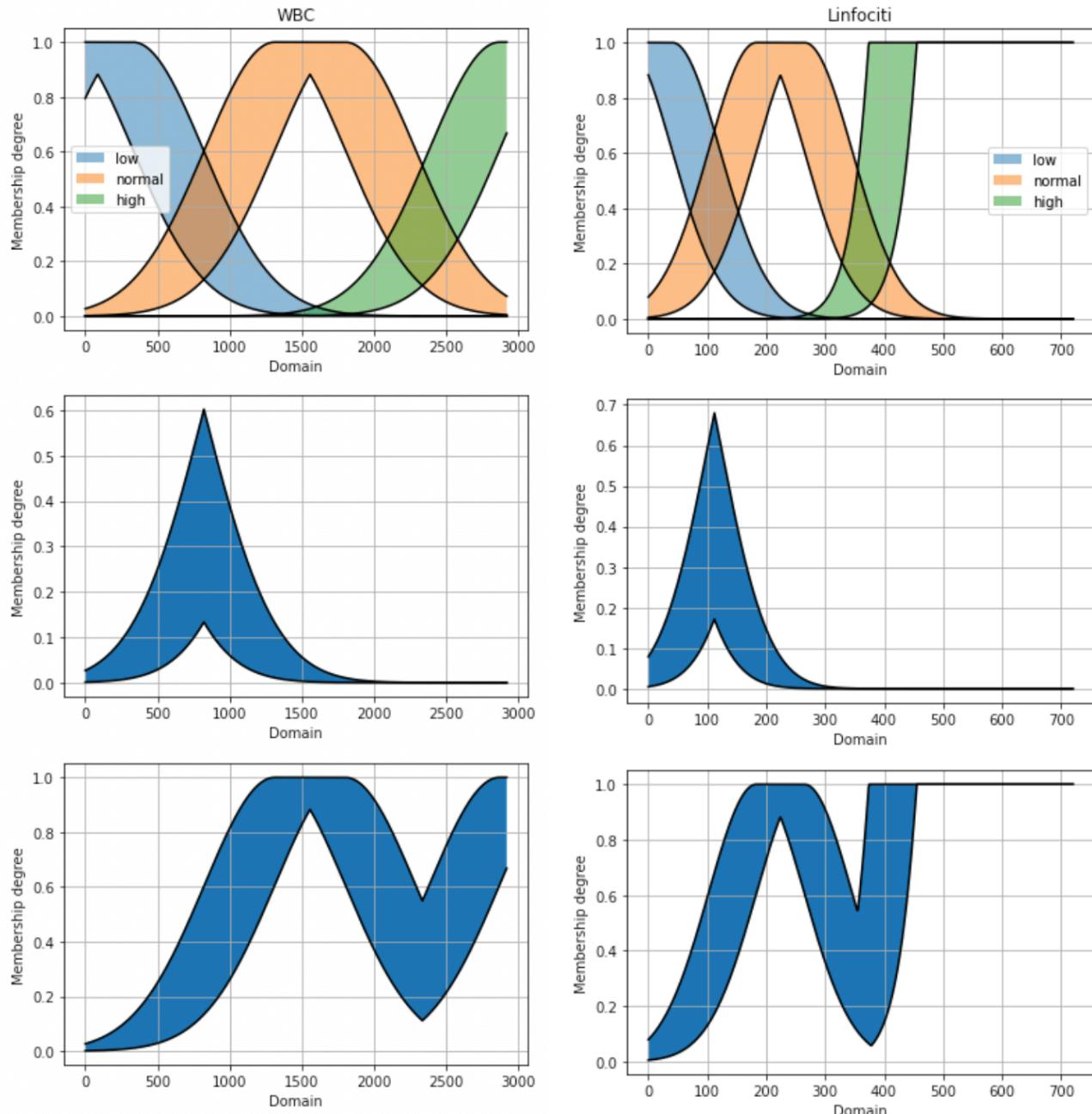
The confusion matrix is:

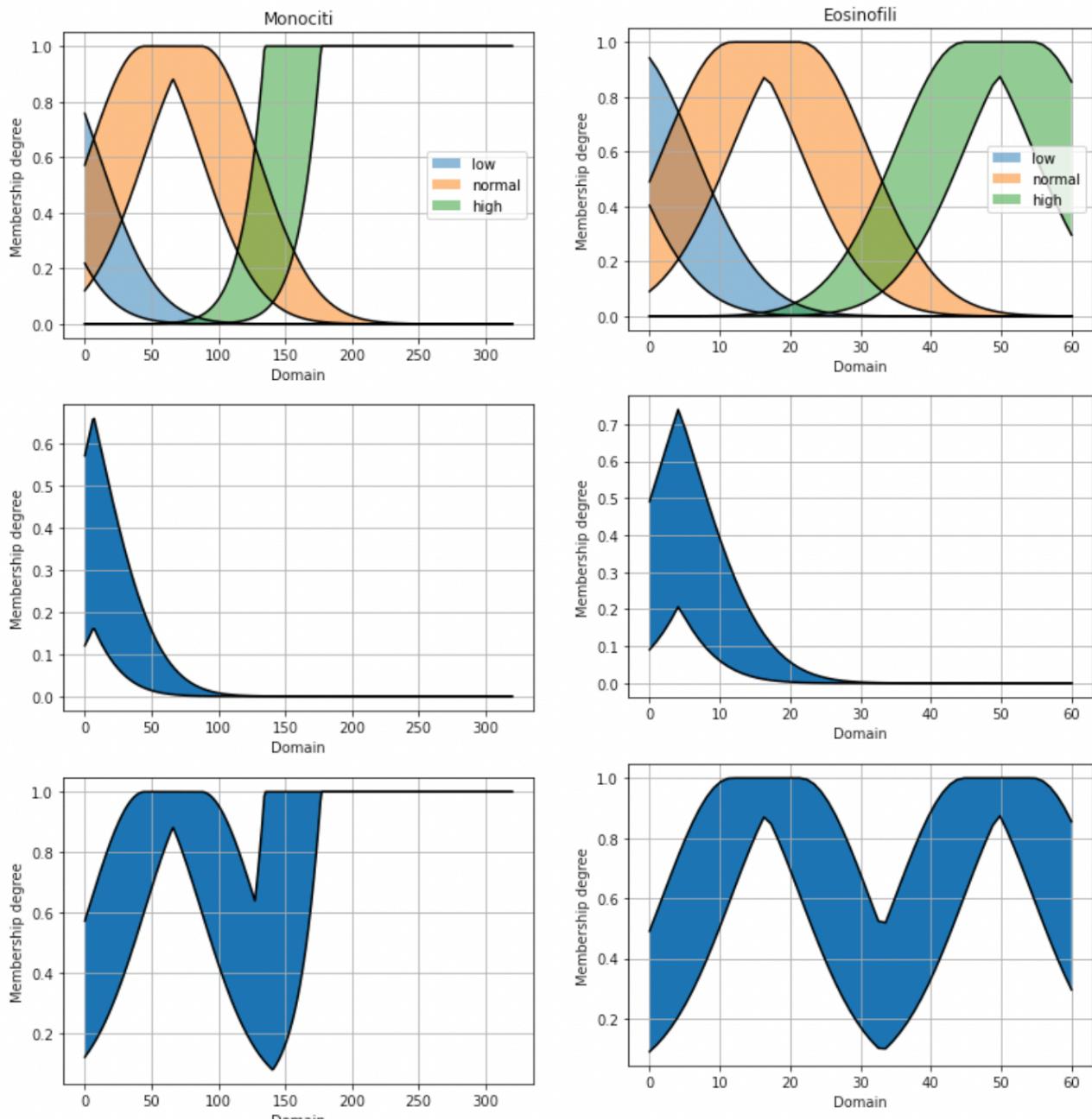


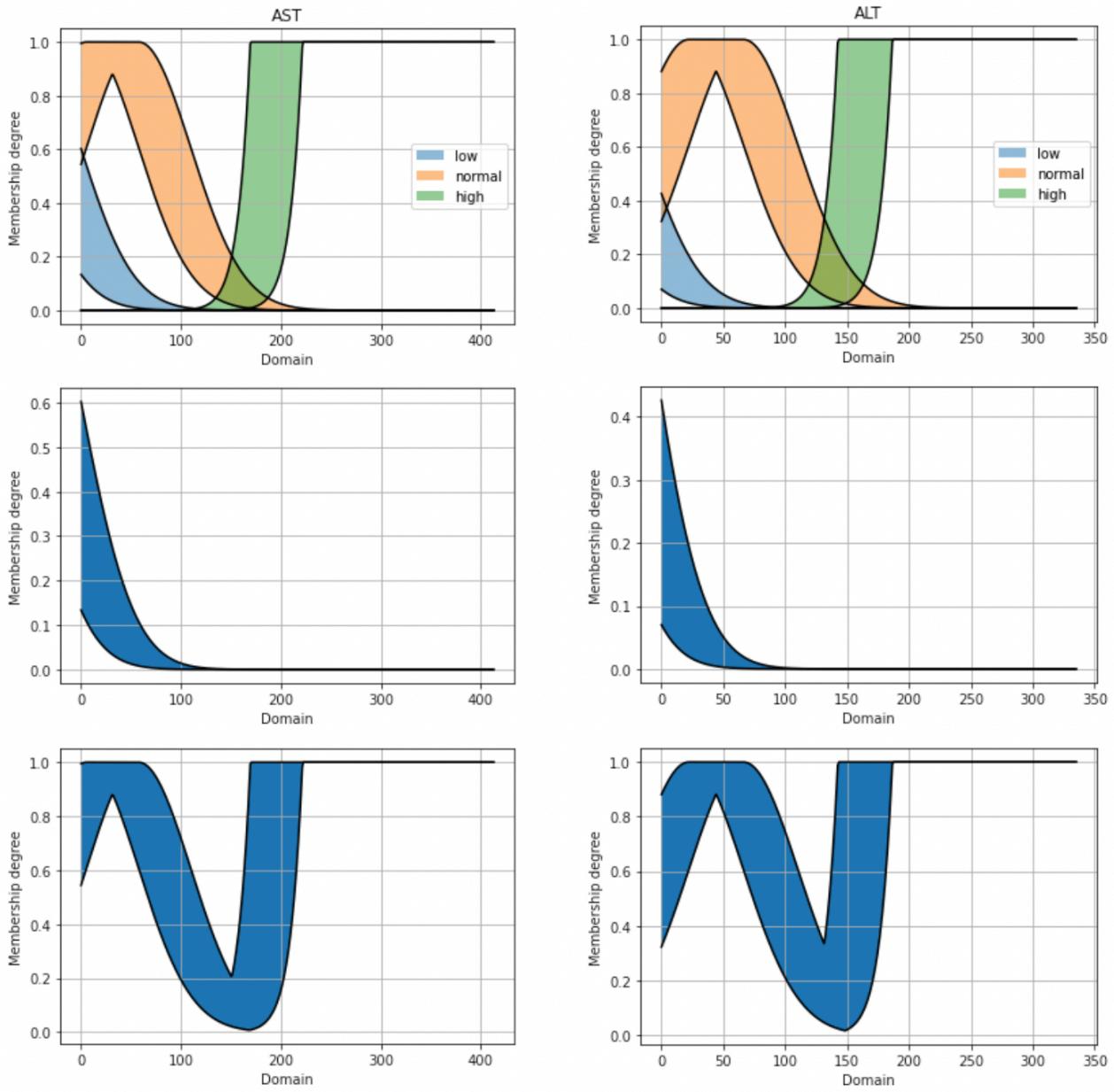
4 Type2 fuzzy systems

This part used *pyit2fls* to create type2 fuzzy sets with each feature's mean and standard deviation.

The pictures below are fuzzy sets, and join and meet them.







4.1 Rules

From the *Mamdani* class, we create an object, then with the *add_input_variable* method, we can define our inputs, and then with the *add_output_variable approach*, we represent our targets.

We can use the *add_rule* method to add rules to our system . our rules will be like this:

```
[([('x1',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [0.08081180345971414, 0.5604438019214517, 0.4796319
984617376, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [0.08081180345971414, 0.5604438019214517, 0.4796319984
617376, 1.0] parameters.),,
    ('x2',
        Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [1.042695211531463, 1.5223272099932006, 0.479631998
4617376, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [1.042695211531463, 1.5223272099932006, 0.4796319984617
376, 1.0] parameters.),
    [('y1',
        Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [1778.2489911703697, 1955.585089547119, 177.3360983
7674953, 9999] parameters, and gauss_uncert_mean_lmf LMF function with [1778.2489911703697, 1955.585089547119, 177.33609837674
953, 9999] parameters.),
    ('y2',
        Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [1778.2489911703697, 1955.585089547119, 177.3360983
7674953, 9999] parameters, and gauss_uncert_mean_lmf LMF function with [1778.2489911703697, 1955.585089547119, 177.33609837674
953, 9999] parameters.)),
    ([('x1',
        Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [0.08081180345971414, 0.5604438019214517, 0.4796319
984617376, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [0.08081180345971414, 0.5604438019214517, 0.4796319984
617376, 1.0] parameters.),,
    ('x2',
```

Example of rules in type2 systems