Fuzzy Logic Homework3

Digit recognition rule-base system

Instructed by Dr.Tahayori

Reza Tahmasebi

saeed77t@gmail.com, Stu ID: 40160957

Table of Contents

1 About the project

In this project, we have a dataset containing digit handwriting; the project aims to create a fuzzy rule-base system based on Mamdani's implication to detect the digits.

2 The project flow

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

2.1 Feature extraction

In this project, two methods have been used to extract features from images .

1. Principle component analysis (PCA)
2. Linear discriminant analysis (LDA), known as FLDA.

In both methods, we scaled our dataset in a way to prevent negative numbers (the absolute value of the minimum of features was added to each feature)

	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	TARGET
0	1984.645002	969.895426	1252.397818	1306.806034	1609.600045	487.846550	1233.436574	1034.410874	343.251669	1354.069788	0
1	119.172658	1027.150738	1224.746258	1019.664889	631.094075	525.777564	902.143949	829.375135	227.674356	781.166812	9
2	559.194747	957.824848	2116.003324	1779.546871	767.505280	137.592277	409.936745	1013.321172	584.813072	789.365974	7
3	264.297649	462.975423	1938.883201	1977.230672	689.736531	891.918932	755.949327	240.553503	579.399573	1268.774321	9
4	1068.260778	1702.968479	1338.236302	1336.333955	718.841500	778.408472	723.490951	13.959123	1695.807920	1099.003585	2
...	...	...	...	...	...	...	...	...	...	...	...
145	524.596690	1673.650032	1589.360883	1354.717863	1746.754071	812.047351	546.856144	1582.895405	832.403838	542.659853	7
146	1387.615856	205.199243	802.793137	1510.108144	1082.573974	1704.342550	413.908959	601.938531	188.049296	690.091009	8
147	1014.425300	639.166369	827.039872	1145.220796	746.327735	518.649548	1245.649439	1574.190565	702.660616	1349.028939	5
148	779.498395	317.544596	850.986342	585.728055	690.320950	1193.072423	139.027002	0.000000	213.671828	979.274319	2
149	1584.490386	1439.393084	2325.170260	61.065837	783.529510	880.277932	1343.034561	672.042979	809.197117	409.806026	0

Example of PCA feature extraction on the dataset

	f1	f2	f3	f4	f5	f6	f7	f8	f9	TARGET
0	11.418653	3.751230	11.981976	4.738933	7.227988	5.790101	3.540473	5.297204	3.083235	0
1	2.983005	9.873780	7.820374	4.899786	6.678031	7.153805	6.736518	2.210640	5.091382	9
2	1.915534	7.037214	10.214927	5.326232	1.394789	4.904078	1.703361	3.640719	2.178594	7
3	4.702640	10.766252	6.857375	4.883366	6.744726	9.980633	6.107834	2.153912	7.599105	9
4	9.503366	7.044473	8.777936	9.279561	2.091095	0.413912	6.762935	8.149234	8.501042	2
...	...	...	...	...	...	...	...	...	...	...
145	0.902193	7.420871	9.918783	5.414055	2.086517	2.784350	1.375413	3.602570	1.697238	7
146	10.422931	11.372033	8.324603	6.109905	2.829766	10.555594	2.129882	7.025751	5.271638	8
147	12.395030	7.653852	5.336362	0.197707	3.036044	4.499732	5.889301	5.793696	3.356799	5
148	9.632242	6.647781	9.052528	8.965568	1.892256	0.867180	6.906464	8.442490	8.417439	2
149	11.424841	2.239495	13.457929	4.173008	7.121447	6.523065	3.717694	6.034560	4.306156	0

Example of FLDA feature extraction on the dataset

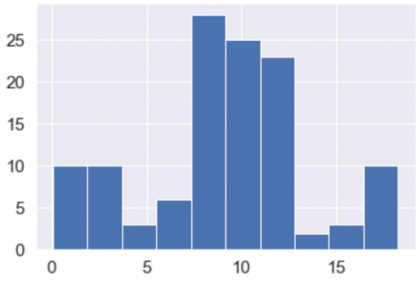
2.2 Feature fuzzification

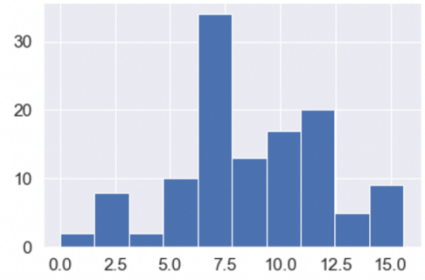
To create our fuzzy sets from each feature, first, we calculate the domain of each feature; then, with the help of the ***skfuzzy*** library, we define each feature as our antecedent. With the *trimf* method from the library, we create the triangulated fuzzy sets.

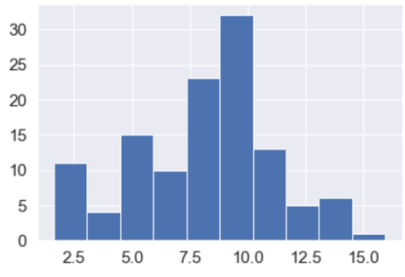
The histogram of each feature helped to understand each feature's ‘normal,’ ‘low,’ and ‘high’ range.

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

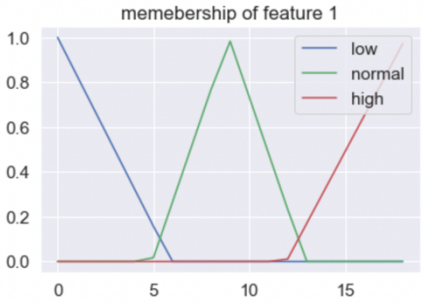
The *Consequent* method from the ***skfuzzy*** library was used to fuzzy-fie the labels.

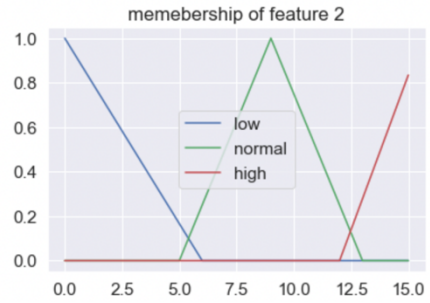


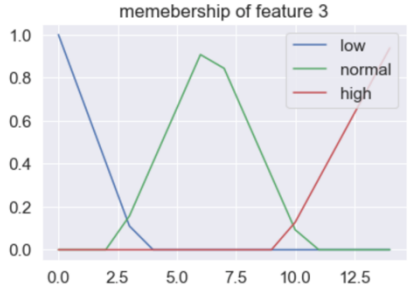


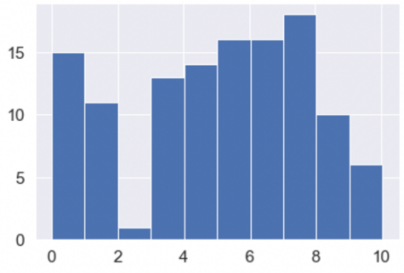


1.628054088383342

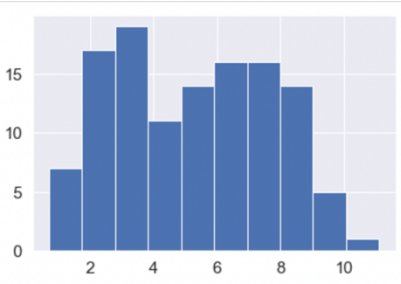




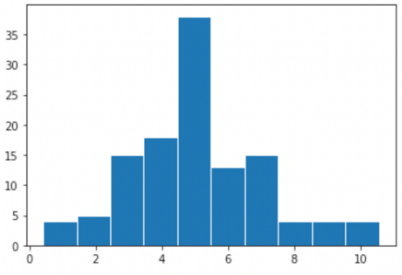




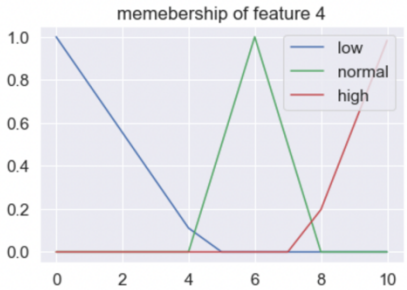
0.0

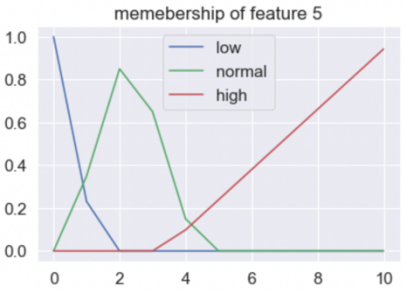


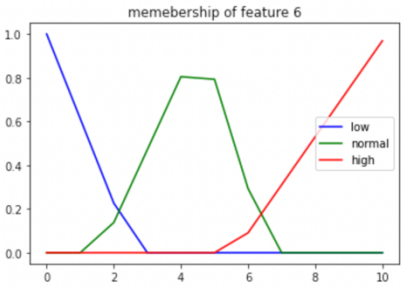
0.6994185364413621

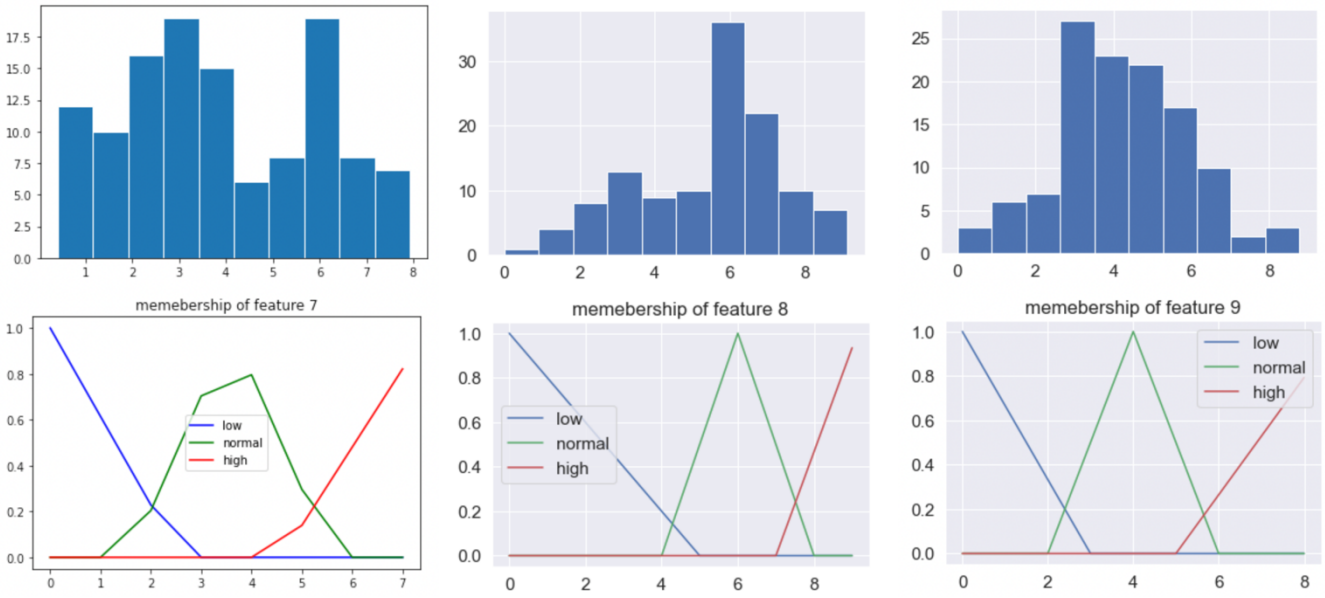


0.4139120289198699

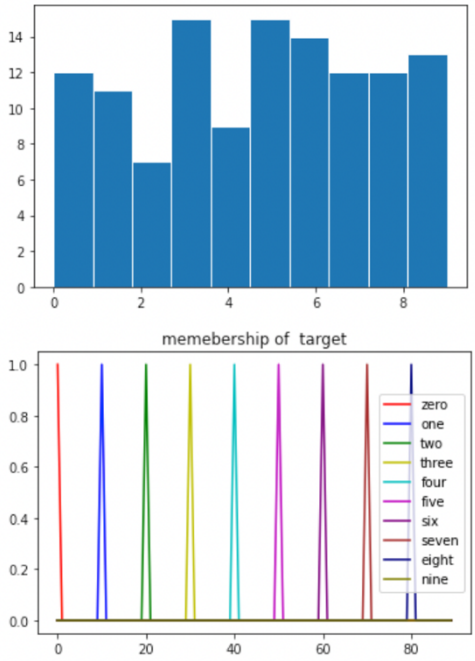








Features fuzzy sets



Labels 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.

```
[[1, 1, 1, 0, 1, 1, 2, 1, 1, 5.0],
 [0, 2, 1, 0, 2, 2, 1, 0, 2, 9.0],
 [1, 1, 1, 1, 0, 2, 1, 2, 1, 8.0],
 [1, 0, 1, 0, 2, 0, 0, 1, 2, 4.0],
 [2, 1, 1, 1, 2, 1, 1, 0, 1, 6.0],
 [1, 1, 1, 2, 1, 2, 1, 2, 1, 8.0],
 [1, 2, 1, 0, 2, 1, 0, 1, 1, 4.0],
 [1, 1, 1, 0, 1, 0, 2, 1, 0, 2.0],
 [0, 1, 1, 1, 1, 0, 0, 0, 1, 7.0],
 [2, 1, 1, 1, 1, 1, 1, 0, 1, 6.0],
```

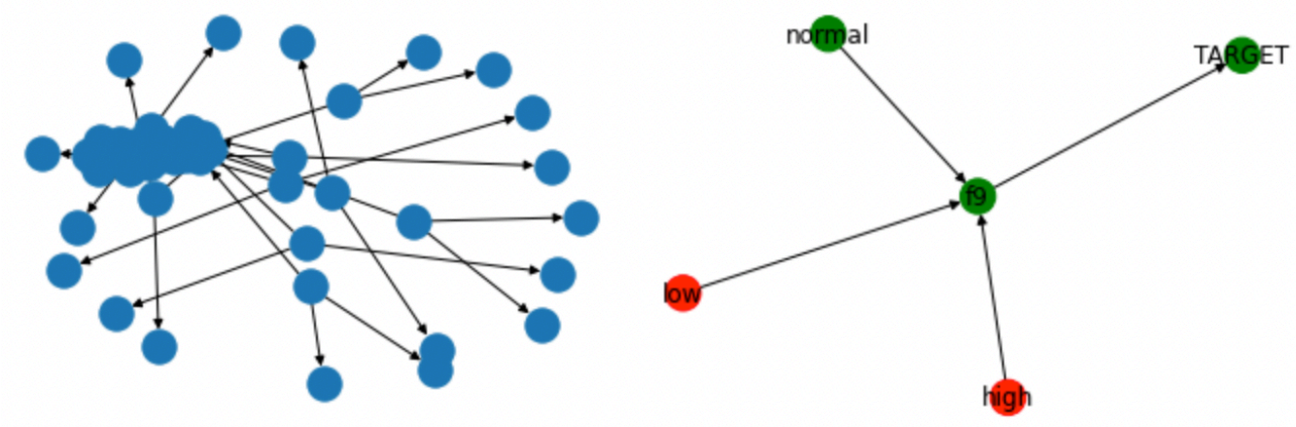
The output of the rule extraction cycle

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 (((((((f1[normal] AND f2[normal]) AND f3[normal]) AND f4[low]) AND f5[normal]) AND f6[normal]) AND f7[high]) AND
f8[normal]) AND f9[normal] THEN TARGET[five]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[low] AND f2[high]) AND f3[normal]) AND f4[low]) AND f5[high]) AND f6[high]) AND f7[normal]) AND f8[low]
) AND f9[high] THEN TARGET[nine]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[normal] AND f2[normal]) AND f3[normal]) AND f4[normal]) AND f5[low]) AND f6[high]) AND f7[normal]) AND
f8[high]) AND f9[normal] THEN TARGET[eight]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[normal] AND f2[low]) AND f3[normal]) AND f4[low]) AND f5[high]) AND f6[low]) AND f7[low]) AND f8[normal
]) AND f9[high] THEN TARGET[four]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[high] AND f2[normal]) AND f3[normal]) AND f4[normal]) AND f5[high]) AND f6[normal]) AND f7[normal]) AND
f8[low]) AND f9[normal] THEN TARGET[six]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[normal] AND f2[normal]) AND f3[normal]) AND f4[high]) AND f5[normal]) AND f6[high]) AND f7[normal]) AND
f8[high]) AND f9[normal] THEN TARGET[eight]
        AND aggregation function : fmin
        OR aggregation function  : fmax,
 IF (((((((f1[normal] AND f2[high]) AND f3[normal]) AND f4[low]) AND f5[high]) AND f6[normal]) AND f7[low]) AND f8[no
rmal]) AND f9[normal] THEN TARGET[four]
```

Example of extracted rules



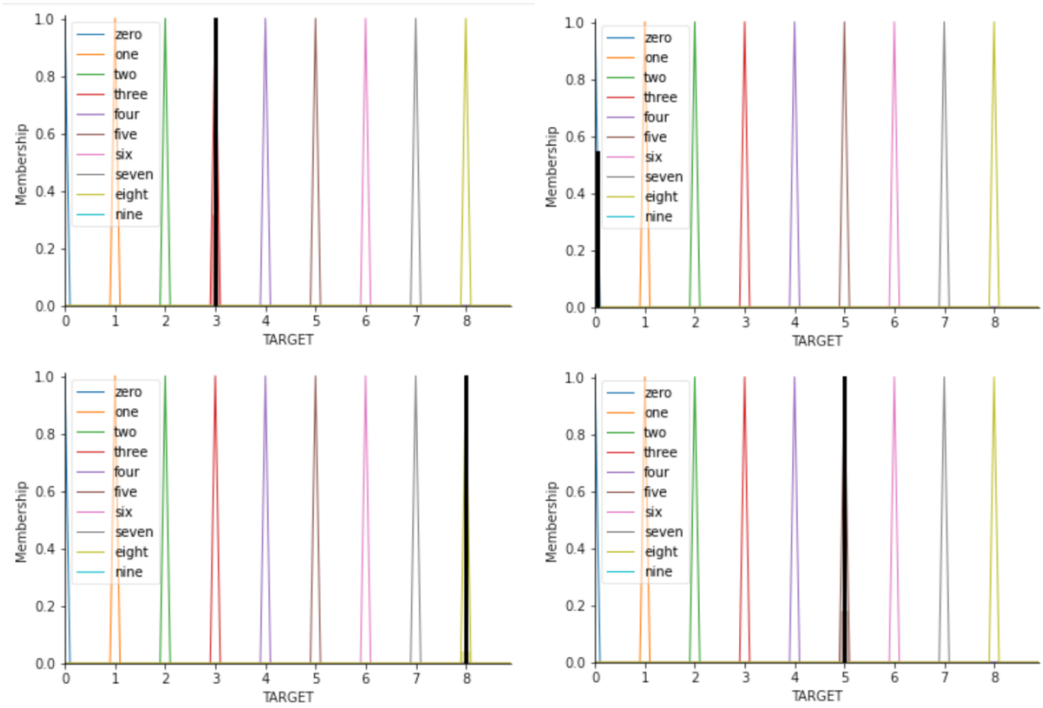
A graph view of a single rule from two different sides.

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 wet 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 project has been run on two different feature sets,

1. PCA feature sets
2. FLDA feature sets

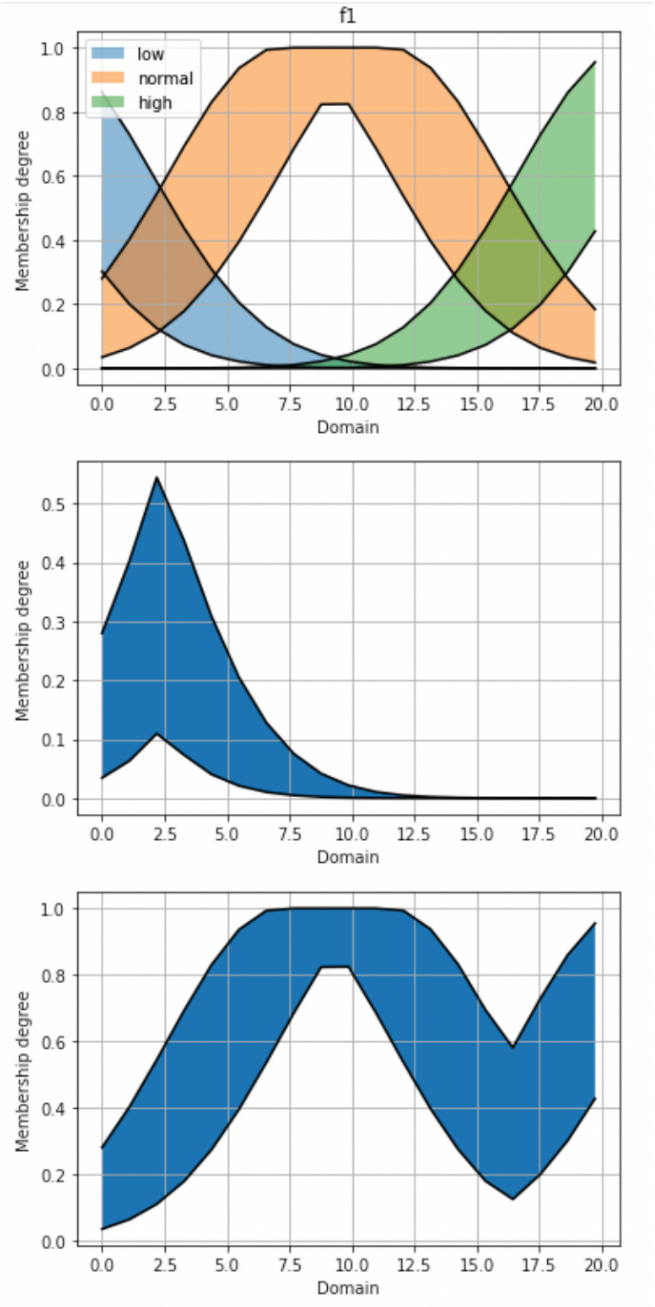
The accuracy of the PCA feature set could have been better, and it was about 10% to 20%.

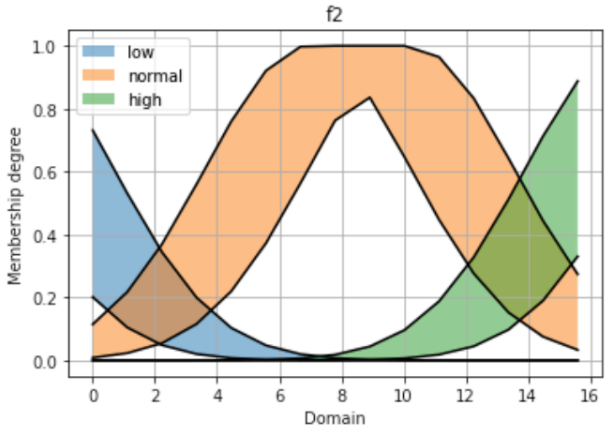
But the accuracy of the FLDA features set was ok, and it was about 60% to 80%; in the lastest run, I got 83.33% accuracy.

4 Type2 fuzzy systems

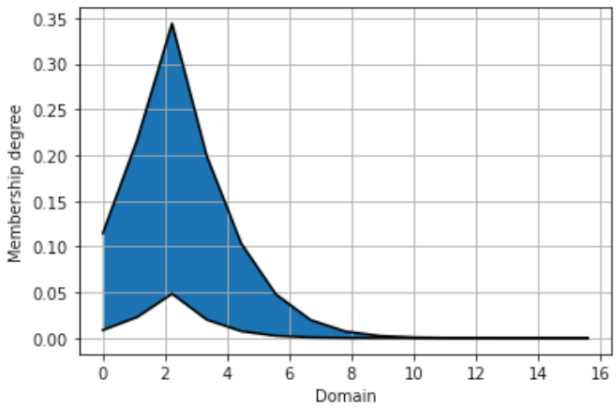
This part used ***pyit2fls*** to create type2 fuzzy sets with each feature's mean and standard deviation.

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

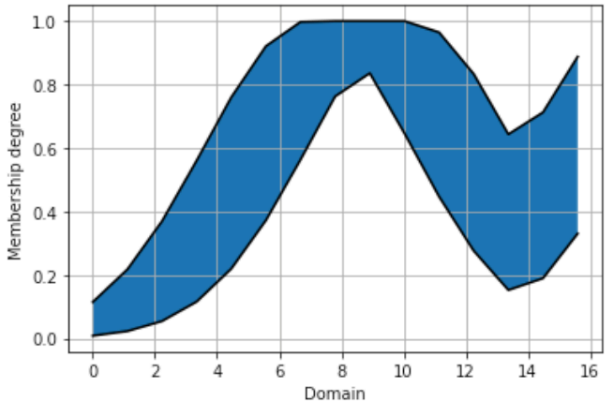

f1
low
normal
high
Membership degree
Domain

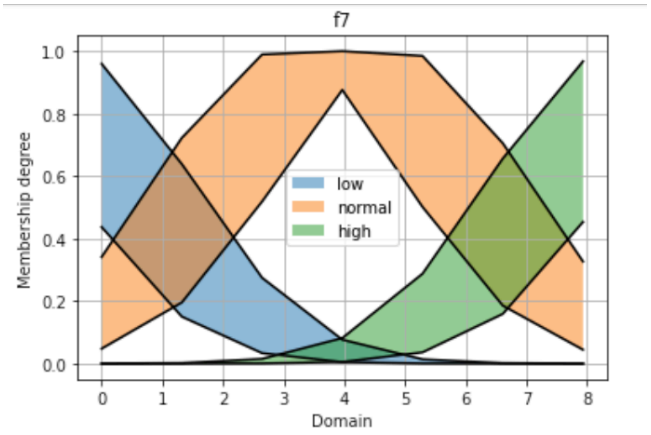

f2
low
normal
high
Membership degree
Domain

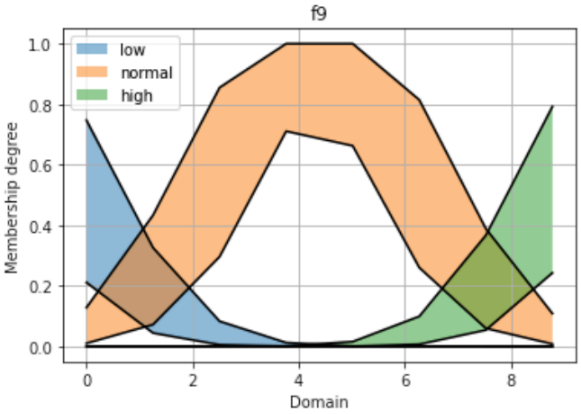
Membership degree
Domain

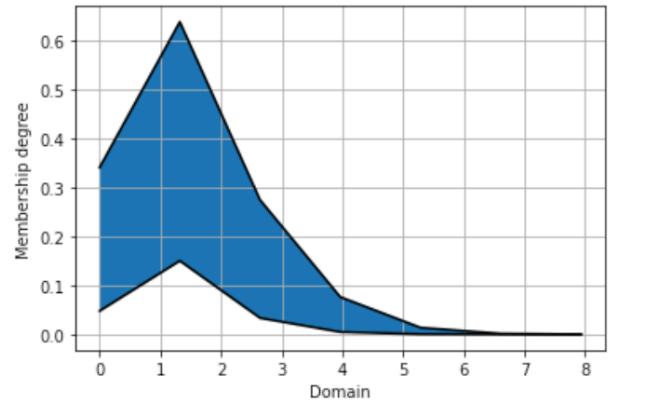

Membership degree
Domain

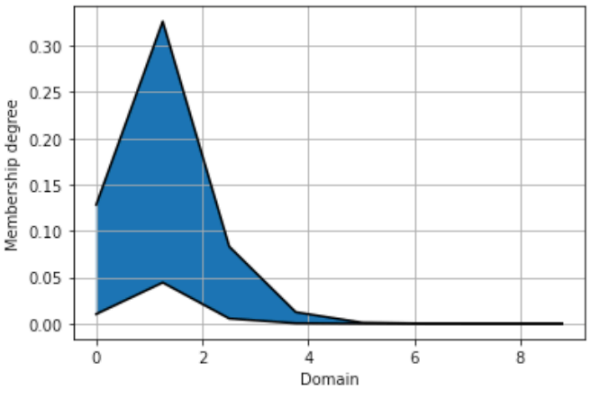
Membership degree
Domain

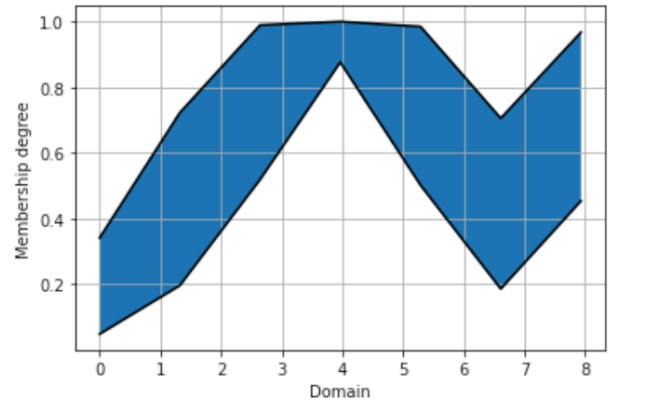

Membership degree
Domain

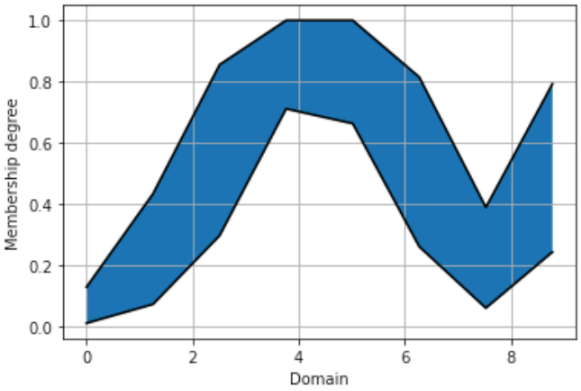

f7
low
normal
high
Membership degree
Domain

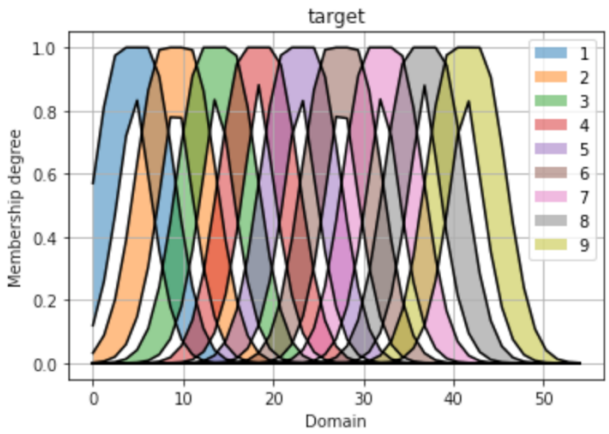

f9
low
normal
high
Membership degree
Domain


Membership degree
Domain


Membership degree
Domain


Membership degree
Domain


Membership degree
Domain



Type2 labels fuzzy sets

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:

```
[([('f1',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880
594596735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880594
596735, 1.0] parameters.),
   ('f2',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [3.126059702701632, 6.073940297298368, 2.9478805945
96735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [3.126059702701632, 6.073940297298368, 2.947880594596735,
1.0] parameters.)],
  [('1',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [10.026059702701632, 12.973940297298368, 2.94788059
4596735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [10.026059702701632, 12.973940297298368, 2.9478805945967
35, 1.0] parameters.),
   ('2',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [10.026059702701632, 12.973940297298368, 2.94788059
4596735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [10.026059702701632, 12.973940297298368, 2.9478805945967
35, 1.0] parameters.)]),
 ([('f1',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880
594596735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880594
596735, 1.0] parameters.),
   ('f2',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [3.126059702701632, 6.073940297298368, 2.9478805945
96735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [3.126059702701632, 6.073940297298368, 2.947880594596735,
1.0] parameters.)],
  [('1',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [3.126059702701632, 6.073940297298368, 2.9478805945
96735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [3.126059702701632, 6.073940297298368, 2.947880594596735,
1.0] parameters.),
   ('2',
    Interval type 2 fuzzy set with gauss_uncert_mean_umf UMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880
594596735, 1.0] parameters, and gauss_uncert_mean_lmf LMF function with [-3.7739402972983673, -0.8260597027016323, 2.947880594
596735, 1.0] parameters.)]),
 ([('x1',
```

Example of rules in type2 systems