

## 1.introduction

In this paper, we tried to detect whether a person has COVID-19 or not based on a simple blood test with the usage of a fuzzy rule based system. We use dataset from IRCCS Ospedale San Raffaele<sup>3</sup> for this study and it consisted of 81 cases. Each case included the patient's age, gender, and values from routine blood tests.

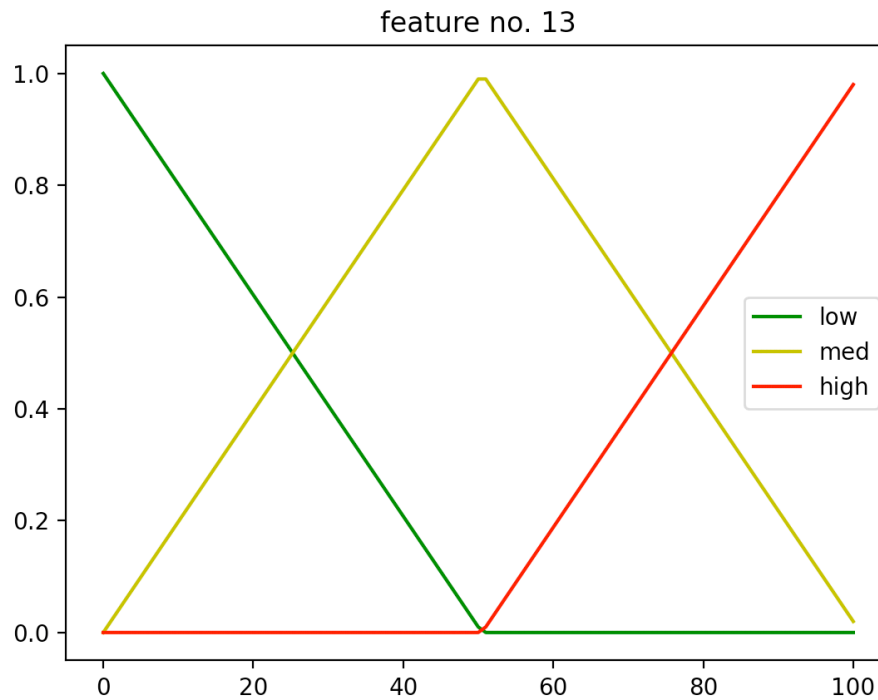
## 2. implementation:

implemented with python. below is some code explained.

- **Feature\_fuzzification:** first we should fuzzify each feature value with some fuzzy sets. to do that i normalized all features to be between 0 and 100. then define 3 fuzzy set for each feature domain(low, medium and high).

```
x_domain = np.arange(minFeatures[i], maxFeatures[i]+1, 1)
```

```
fiSets.append(fuzz.trimf(x_domain, [minFeatures[i],  
minFeatures[i], minFeatures[i]+step])) #low  
fiSets.append(fuzz.trimf(x_domain, [minFeatures[i], minFeatures[i]  
+step, minFeatures[i]+2*step])) #medium  
fiSets.append(fuzz.trimf(x_domain, [minFeatures[i]+step,  
minFeatures[i]+2*step, minFeatures[i]+2*step])) #high  
ri.append(membership)
```



**Extract\_rule\_from\_patients:** to extract rules, we use each train patient data to extract one rule. the way i do that is i compute the membership value of each patient's feature in fuzzy sets of that feature (low, medium, high), and choose the fuzzy set with largest membership value for that feature. i do this for all of the features of that patient to make our anticipates. and the label of that patient is the q. for example: if feature1 is high AND feature2 is medium AND feature3 is low AND ... Then patient is infected. after i extract rules from all patients, i eliminate similar rules.

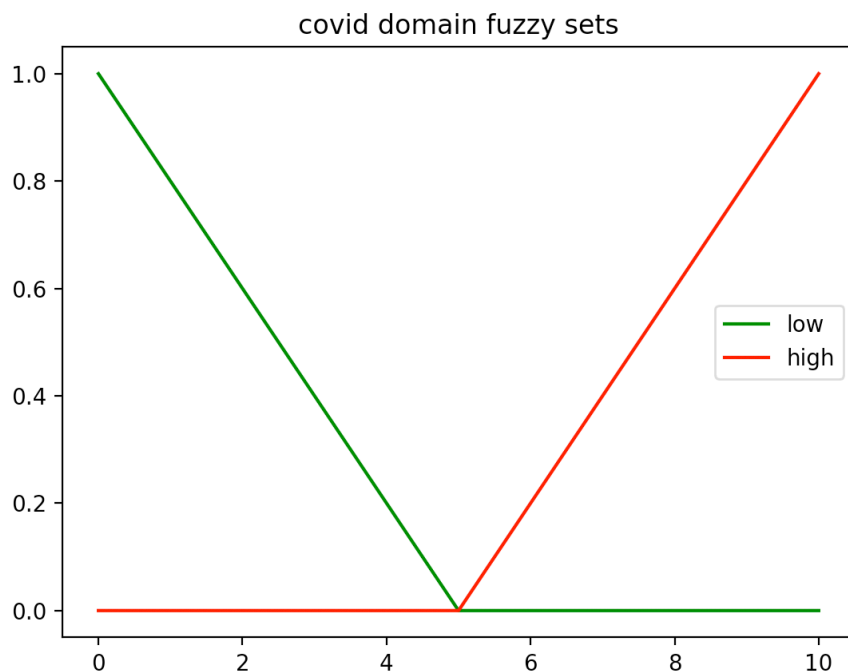
```
x_domain = np.arange(minFeatures[i], maxFeatures[i]+1, 1)
lo_member = fuzz.interp_membership(x_domain,
self.featuresFuzzySets[i][0], p[i])
md_member = fuzz.interp_membership(x_domain,
self.featuresFuzzySets[i][1], p[i])
hi_member = fuzz.interp_membership(x_domain,
self.featuresFuzzySets[i][2], p[i])
memList = [lo_member, md_member, hi_member]
membership = memList.index(max(memList))
```

here is the example of two rules:

```
if feature 0 is medium and if feature 1 is low and if feature 2 is
low and if feature 3 is low and if feature 4 is low and if feature
5 is low and if feature 6 is low and if feature 7 is low and if fea
ture 8 is low and if feature 9 is low and if feature 10 is low and
if feature 11 is low and if feature 12 is low and if feature 13 is
low then patient has covid
```

```
if feature 0 is high and if feature 1 is medium and if feature 2 is
medium and if feature 3 is medium and if feature 4 is medium and i
f feature 5 is low and if feature 6 is low and if feature 7 is low
and if feature 8 is medium and if feature 9 is medium and if featur
e 10 is low and if feature 11 is low and if feature 12 is low and i
f feature 13 is medium then patient doesn't have covid
```

**Predict:** to predict whether a patient is infected or not. first we define two fuzzy set showing high and low covid in domain of 0 to 10. then, we use our rules. for each feature, we find membership value of that feature value in that rule. then we find minimum of all features membership value in that rule fuzzy sets. and then find alpha cut of that min membership in corresponding q fuzzy set (if q of that rule is not having covid we find alpha cut in low covid fuzzy set and if q of that rule is having covid we find alpha cut in high covid fuzzy set).



then we union all alpha cuts from each rule and defuzzify the union with centroid method. and compare result of low covid and high covid defuzzified values, which of them was larger is predicted.

```
memberships = []
for i in range(len(rule)-1):
    x_domain = np.arange(minFeatures[i], maxFeatures[i]+1, 1)
    fuzzySetToFire = self.featuresFuzzySets[i][rule[i]]
    m = fuzz.interp_membership(x_domain, fuzzySetToFire, p[i])
    memberships.append(m)
minMembership = min(memberships)
alphaCutsCovid.append(fuzz.lambda_cut(qFuzzyHighCovid,
minMembership))
defuz = self.unionAlphacuts(alphaCutsCovid)
covidHigh = self.defuzzy_alpha_cut(defuz)
```

## Results:

```
results:
accuracy: 93.75
precision: 0.9375
recall: 1.0
fmeasure: 0.967741935483871
```

## Comparison:

in faze 1, we just use mean of each feature and with the help of min max method it gave us 76 percent accuracy. in faze 2, we use mean, min, max of each feature. define one fuzzy set for each feature based on mean, min and max, and then with the help of min max of fuzzy sets method it gave us 88 percent accuracy. this faze has more accurate results than faze1 and faze2. because in this faze, with the help of fuzzy rule base system we use

5 of 5

all data of all patients and not just the mean of each feature. and it gave us 93 percent accuracy which is an acceptable result.