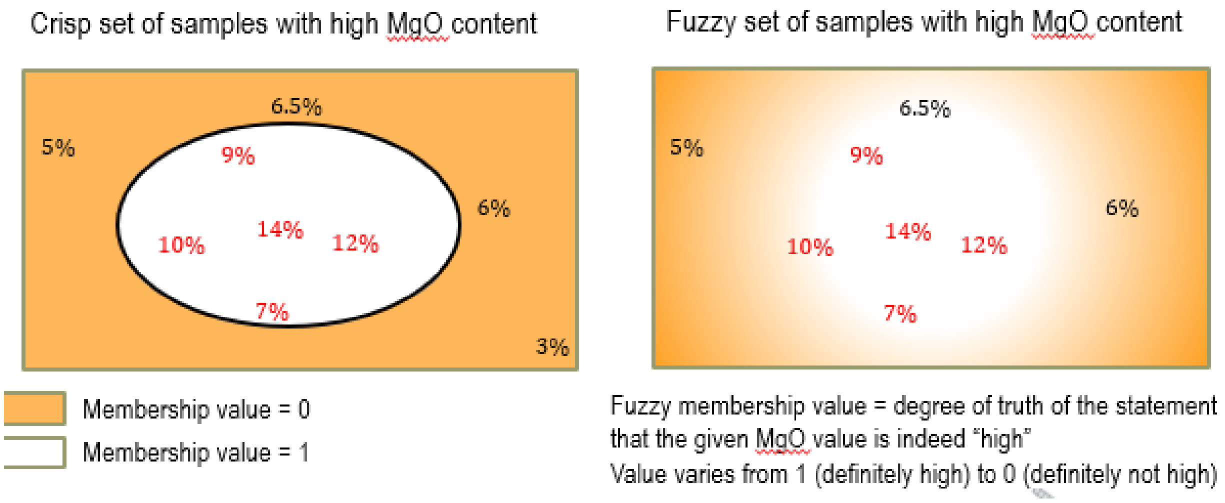
QGIS Plugin for Fuzzy Inference Systems

Prakriti Shetty, 200020095, BTech 3rd year, IIT Bombay

Prof Alok Porwal

Background:

Fuzzy Logic is a way to use natural language in logic and take into consideration imprecision and approximate reasoning based on knowledge.
Fuzzy sets are extensions of classical sets where the membership value isn't restricted to "in the set"=1 or "outside the set"=0. It represents a partial membership value of each datapoint varying from 1 (definitely in the set) to 0 (definitely not in the set).
In geology, fuzzy logic is used for prospectivity analysis to take into account the knowledge and experience of geologists, who can explain their inferences in linguistic terms rather than crisp numeric values. Prospectivity analysis is most often not a clear black/ white case and involves a lot of grey area, which can be modelled using fuzzy logic.


Crisp set of samples with high MgO content

Fuzzy set of samples with high MgO content

Fuzzy membership value = degree of truth of the statement that the given MgO value is indeed "high"
Value varies from 1 (definitely high) to 0 (definitely not high)

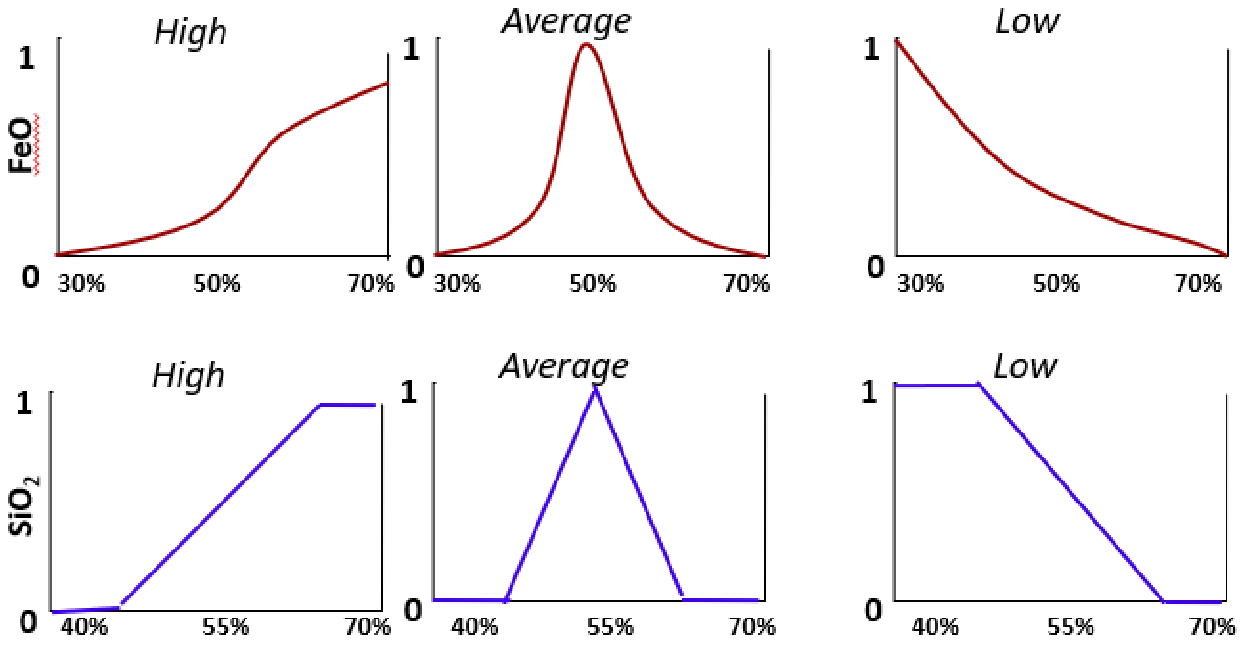
QGIS is a free and open source geographic information system that allows the creation, editing, visualisation, analysis and publishing of geospatial information.

Scope of the Project:

To create a QGIS Plugin to implement the fuzzy inference system directly within QGIS instead of the current method of exporting to MATLAB, performing computation, and then re-importing into GIS systems.

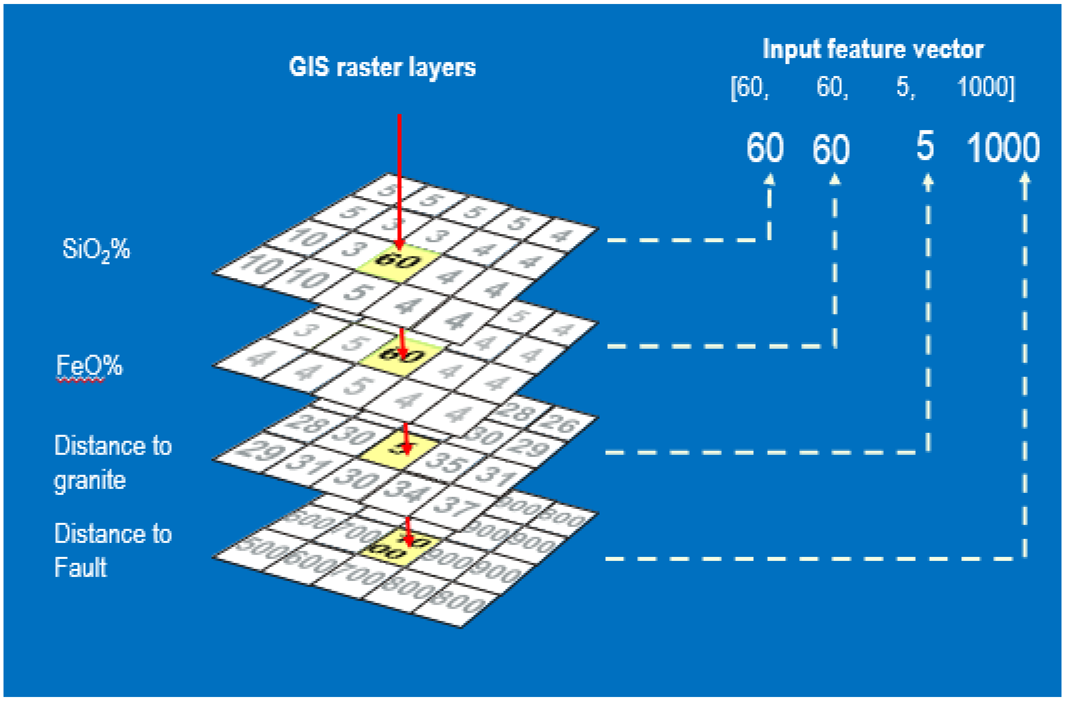
Steps to implement Fuzzy Logic:

1. Step 1: Identify the factors that control a system:
    a. *For example:*
    b. *Formation of a deposit (or mineral potential of an area) depends on the following factors:*
        i. *FeO content of the rocks*
        ii. *SiO2 content of the rocks*
        iii. *Closeness (or proximity) to granite*
        iv. *Closeness (or proximity) to faults*
2. Step 2 : Identify the variables for each of the factor:
    a. *FeO content: {high, average, low}*
    b. *SiO2content: {high, average, low}*
    c. *Proximity to granite :{proximal, intermediate, distal}*
    d. *Proximity to faults: {proximal, intermediate, distal}*
3. Step 3 : Identify the output variable of the system
    a. *Mineral potential: {high, average, low}*
4. Step 4 : Decide a fuzzy membership function for each variable

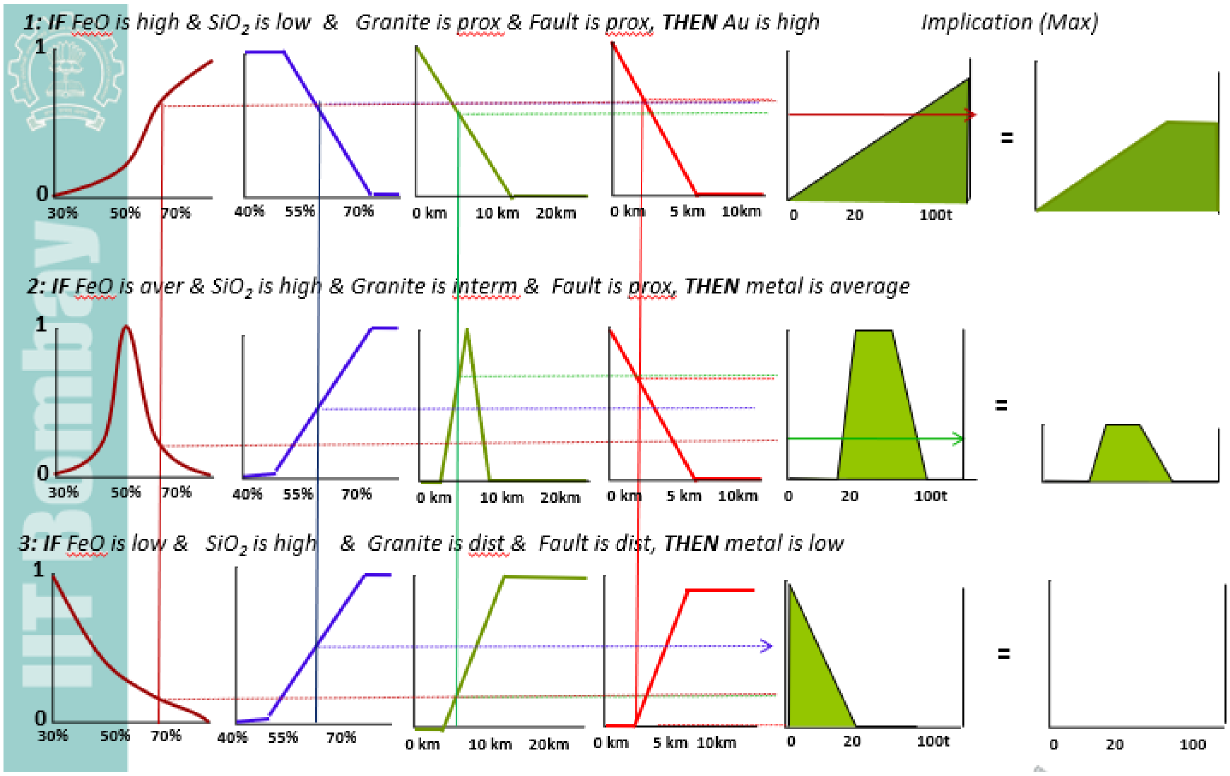


5. Step 5: Develop a set of fuzzy if-then rules to explain the behavior of the system
   a. *Rule1: IF FeO is high AND SiO2 is low AND Granite is proximal AND Fault is proximal, THEN gold potential is high.*
   b. *Rule 2: IF FeO is average AND SiO2 is high AND Granite is intermediate AND Fault is proximal, THEN gold potential is average.*
   c. *Rule 3: IF FeO is low AND SiO2 is high AND Granite is distal AND Fault is distal, THEN gold potential is low*

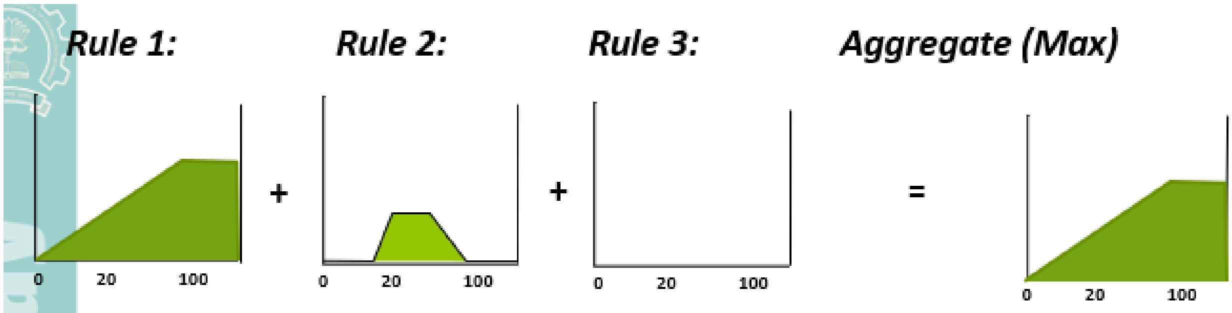
6. Step 6: Rasterize the input predictor maps, combine them, and generate feature vectors



7. Step 7: Represent fuzzy if-then rules in terms of membership functions


1: IF FeO is high & SiO2 is low & Granite is prox & Fault is prox, THEN Au is high
Implication (Max)
1
0
30% 50% 70%
40% 55% 70%
0 km 10 km 20km
0 km 5 km 10km
0 20 100t
=
2: IF FeO is aver & SiO2 is high & Granite is interm & Fault is prox, THEN metal is average
1
0
30% 50% 70%
40% 55% 70%
0 km 10 km 20km
0 km 5 km 10km
0 20 100t
=
3: IF FeO is low & SiO2 is high & Granite is dist & Fault is dist, THEN metal is low
1
0
30% 50% 70%
40% 55% 70%
0 km 10 km 20km
0 km 5 km 10km
0 20 100t
=
0 20 100

8. Step 8 : Combine outputs of each rule



Defuzzify (Find centroid)

Formula for centroid

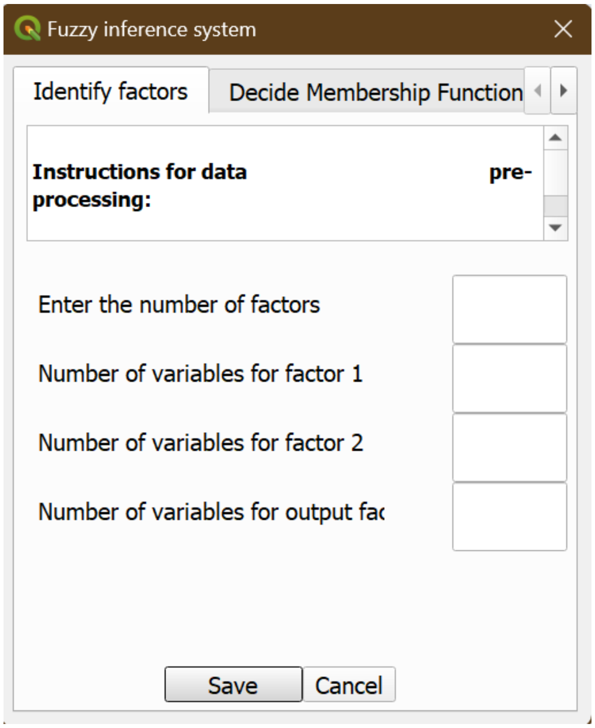
$$= \frac{\sum_{i=0}^{n} x_i \times \mu(x_i)}{\sum_{i=0}^{n} \mu(x_i)}$$

Achieved Deliverables for the project:

I've developed a simplistic version of the plugin which implements the logic correctly but takes some assumptions.

Features of the plugin realized -

1. **Identify Factors** : Identifying the factors that affect the system (inputs as number of factors and corresponding number of variables for each of them)


Fuzzy inference system
Identify factors
Decide Membership Function
Instructions for data pre-processing:
Enter the number of factors
Number of variables for factor 1
Number of variables for factor 2
Number of variables for output fac
Save
Cancel

a. NOTE: Assumption here -

i. Hardcoded 2 input factors and 3 variables for each of these 2 factors as well as the output factor.

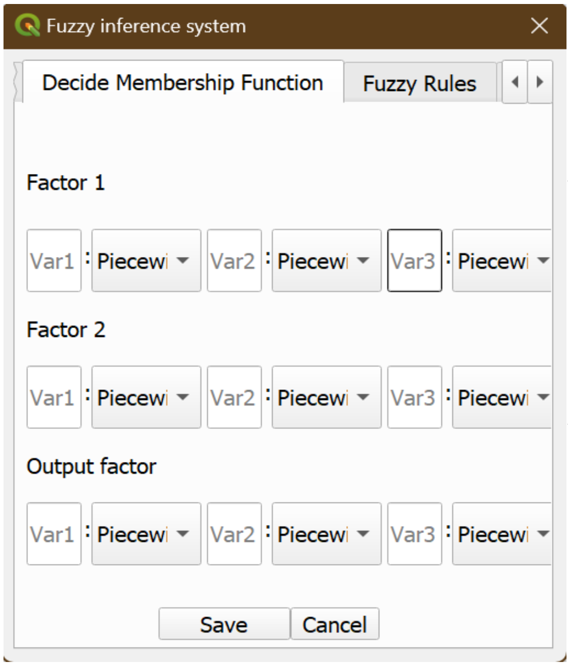
b. Reason and feasibility of the assumption to be overcome -

i. Reason - Researched a lot (over the web as well as a mail to Bijal Ma'am which I didn't get a reply for because she might've been busy) but couldn't figure out how to make a dynamic GUI such that QLabels and QLineEdits appear on the plugin interface. The nearest I could get to a dynamic GUI was pasting in QLineEdits but that wouldn't help here.

ii. Feasibility to overcome - When we can figure out how to make the GUI dynamic.

2. **Membership Functions :** Provision to label each of the variables for the inputs as well as output factor

3. **Membership Functions (contd.) :** Choosing a membership function for each of them from the following options - Piecewise Linear, Sigmoid, Gaussian, Quadratic and Cubic polynomial



Factor 1

Var1 : Piecewi Var2 : Piecewi Var3 : Piecewi

Factor 2

Var1 : Piecewi Var2 : Piecewi Var3 : Piecewi

Output factor

Var1 : Piecewi Var2 : Piecewi Var3 : Piecewi

a. NOTE - Assumption here -

i. Hardcoded Piecewise linear function=> y=0.1x (Domain: [0,10] and Range: [0,1]

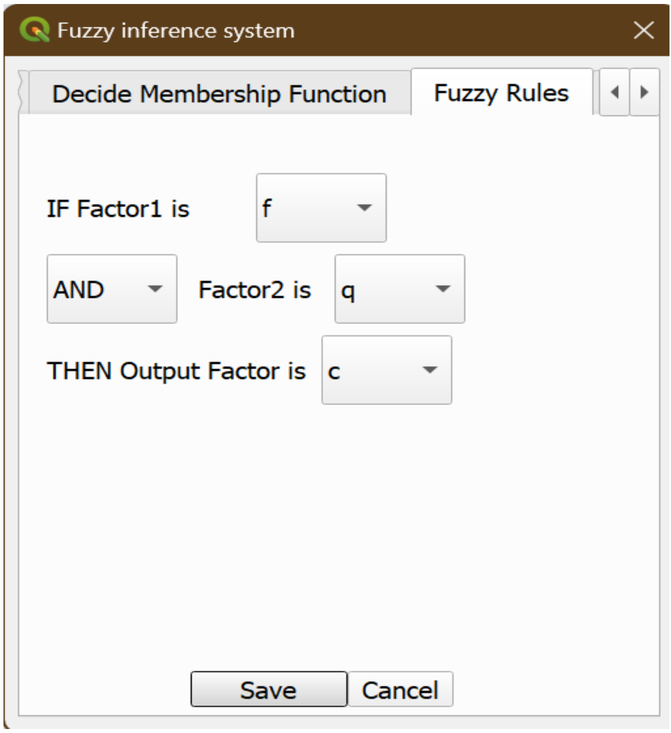
ii. Hardcoded parameters of the gaussian=> mean = 0 and standard deviation = 1

b. Reason and feasibility of the assumption to be overcome -

i. Reason - Just the time constraint

ii. Feasibility to overcome - Can easily modify a few lines of code to allow users to define all parameters of the functions they want.

4. **Fuzzy Rules** : Provision to add fuzzy rules governing the system.



IF Factor1 is f

AND Factor2 is q

THEN Output Factor is c

a. NOTE - Assumption here -

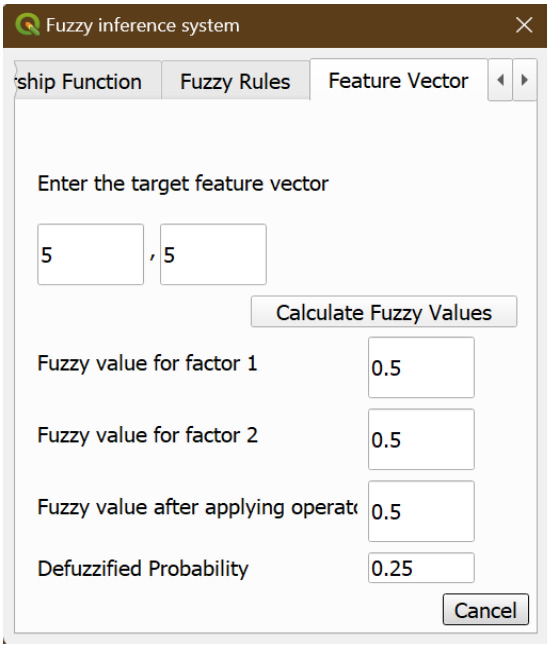
i. Hardcoded provision for only 1 fuzzy rule for now.

b. Reason and feasibility of the assumption to be overcome -

i. Reason - Just the time constraint

ii. Feasibility to overcome - Very simple. All I have to do is add to the GUI and put a condition before calculation of the final fuzzy output to add up all of these answers from each fuzzy rule.

5. **Feature Vector :** Series of steps – Input data —-> calculate the fuzzy values for each factor —-> apply the operation defined in the fuzzy rule —-> get the fuzzy value of the output factor —-> Defuzzify by the centroid location method.



Enter the target feature vector

5 , 5

Fuzzy value for factor 1 0.5

Fuzzy value for factor 2 0.5

Fuzzy value after applying operat 0.5

Defuzzified Probability 0.25

a. NOTE - Assumption here -

i. Defuzzification by the centroid method is implemented only for the piecewise linear - piecewise linear - piecewise linear combination of the 2 input factors and 1 output factor.

b. Reason and feasibility of the assumption to be overcome -

i. Reason - Couldn’t figure out how to implement the centroid method without extensive geometrical considerations and mathematical calculations.

ii. Feasibility to overcome - A little unclear to me at the moment, but I believe I can use integration and make the problem simpler in future iterations.

Code is hosted at:

https://github.com/PrakritiShetty/GNR633Project_QGISPlugin

References and Bibliography:

1. **Professor Alok Porwal** - who very graciously helped me incorporate my love for coding into a very domain-heavy field such as geology.
2. **Bijal Chudasama Ma'am** - her Weights of Evidence Model has been a huge help throughout the development of my code. Also, the support she provided at the beginning.
3. **Debosmita Paul Ma'am** - her cheerfulness and cooperation has always kept my spirits up throughout the course and project.
4. **Sandeepan Sir** - He directed me to really useful resources like the QGIS mailing list and source codes for other plugins

https://www.csre.iitb.ac.in/~alok/GNR633_2023/
https://towardsdatascience.com/fuzzy-inference-system-implementation-in-python-8af88d1f0a6e
https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html
https://www.researchgate.net/publication/286510357_Mineral_prospectivity_mapping_with_weights_of_evidence_and_fuzzy_logic_methods/figures?lo=1
https://www.researchgate.net/publication/227894017_Using_fuzzy_logic_within_a_GIS-environment_to_enhance_conceptually_based_prospectivity_analysis_of_MVT_mineralisation
https://www.youtube.com/watch?v=-Y7kaubL_4k
https://download.osgeo.org/qgis/doc/manual/qgis-1.4.0_coding-compilation_guide_en.pdf
https://www.e-education.psu.edu/geog489/node/2358
https://www.semanticscholar.org/paper/PREPARING-MINERAL-POTENTIAL-MAP-USING-FUZZY-LOGIC-Karimi-Menhaj/d39c6634a543ac5f84c9ae03ea07b5754da30d0b
https://realpython.com/python-pyqt-gui-calculator/
https://realpython.com/qt-designer-python/
https://forum.qt.io/topic/93226/make-a-conditional-rule-between-combobox-value-and-button/2
https://www.youtube.com/watch?v=CvWl-Rhy2wI
https://doc.qt.io/qtdesignstudio/quick-states.html
https://www.e-education.psu.edu/geog489/node/2358
https://docs.qgis.org/2.18/en/docs/user_manual/working_with_vector/attribute_table.html
https://subscription.packtpub.com/book/programming/9781786460370/10/ch10lvl1sec94/a-simple-plugin-example
https://docs.qgis.org/3.28/en/docs/pyqgis_developer_cookbook/plugins/index.html
https://subscription.packtpub.com/book/application-development/9781787129450/1/ch01lvl1sec18/adding-several-widgets-in-a-loop
https://gisgeography.com/image-compression-encoding/
https://gisgeography.com/rasterization-vectorization/
https://docs.scipy.org/doc/scipy/tutorial/stats.html
https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.measurements.center_of_mass.html