First I put all data MCU into an csv file name “data.csv”

After that I use python to load csv file into pandas data frame

**Analytics – 1**

First I Create a numpy array with no element and loop a data frame row by row. Check it if exist a number, I append to numpy array, if not so do nothing, that's against having many non-existent values included in the calculation.

Average critics rating=

The results are displayed in a new column named “Average Critics Rating” in file “output.csv”

**Analytics – 2**

I will divide it into two main groups, the group with the rating close to the audience score and the group with the rating far from the audience score.

I think I will choose one of three algorithms:

1. Classification: Based on audience rating i will create a confidence interval. For example, audience score is 7 and i will create an interval of +2 and -2 so Critic with rating between 5 and 9 will satisfy group 1 and outside the range from 5 to 9 will satisfy group 2
2. K-Nearest Neighbors(KNN): I will take out the audience rating and compare the rating difference between each Critic and the audience, then count if the 2 Critic with the most rating closest to the audience will be in group 1, the other will be in group 2
3. Random forest: Select random data and then evaluate whether it is in group 1 or 2 . However I don't like this algorithm very much because it's hard to manage what I'm actually doing

I decided to choose the first algorithm because it will not be forced to get how many Critic in a group like the second option and it is not difficult to manage the input data like the third option.

**Modeling – 1.1**

I go through all the Critic ratings and get the maximum and minimum values. I then normalize to the range from 0 to 1.

Normalize[0:1]=

Then I calculate population mean (μ) and standard deviation (σ)

μ=

σ =

Use a for loop through all values in the array to calculate z score

z score=

**Modeling – 1.2**

First calculate mean

μ=

then calculate covariance matrix by following step:

First remove the rows with null values, take only the rows that do not have null values because null values are not possible when calculating the covariance matrix. We have matrix A

Second calculate deviation matrix

D =

Finally calc covariance matrix:

C =

**Modeling – 1.3**

Since it's too difficult to solve with 5 variables, I tried to write the equation in a lot of pages but still haven't solved so I decided to use 2 variables to generalize. The first variable remains the value(x1, μ1, σ1) of the column to be searched, the other variable will average the remaining columns

We have two variables, X1 and X2 and that these are bivariately normally distributed with mean vector components μ1 and μ2 and variance-covariance matrix shown below:

The determinant of the variance-covariance matrix is simply equal to the product of the variances times 1 minus the squared correlation.

The inverse of the variance-covariance matrix takes the form below:

Joint Probability Density Function for Bivariate Normal Distribution

Substituting in the expressions for the determinant and the inverse of the variance-covariance matrix we obtain, after some simplification, the joint probability density function of (X1, X2) for the bivariate normal distribution as shown below:

Now apply maximizing log likelihood:

First I want to calculate the loganepe of *f*(x1,x2)

I want to examine the graph of this function follow by p because only p is currently the only value that has not been determined remember p is the squared correlation so p belong to [0 🡪 1].

Because now x1, μ1, σ1, x2, μ2, σ2 can change based on how we choose 2 out of 5 variables Critic.

Base on my calculation from ex2 code we have

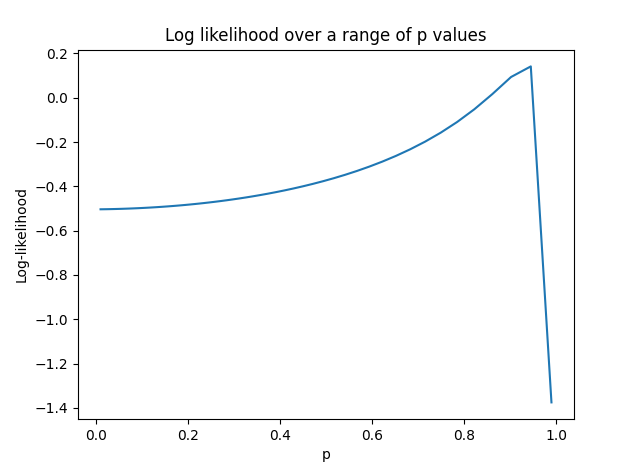
[μ1 , μ2 , μ3 , μ4 , μ5] =[-0.7911122022087103, -0.043950677900483424, 0.8692467406984606, 0.039067269244875136, 0.37113905782630924]

[σ1, σ2, σ3, σ4, σ5] = [56.23653685037043, 4.36478399072784, 61.761849866249165, 3.535291072462517, 26.520324468039814]

Case 1: μx1= μ1==-0.79111 and μx2= μ2345 = AVG(μ2, μ3, μ4, μ5) = 0.30885

σx1= σ1=56.2365 and σx2 = AVG(σ2, σ3, σ4, σ5) = 24.0455

the equation become:



Now we evaluate function by code in file “eva\_func.py”

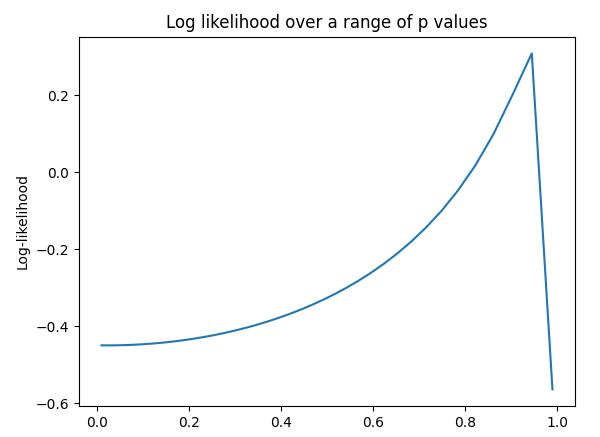
We can see max Ln(f(x1,x2))=0.138 when p = 0.943

Case 2: μx1= μ2= -0.04395 and μx2= μ1345 = AVG(μ1, μ3, μ4, μ5) = 0.1220

σx1= σ2= 4.3647 and σx2 = AVG(σ1, σ3, σ4, σ5) = 37.0134

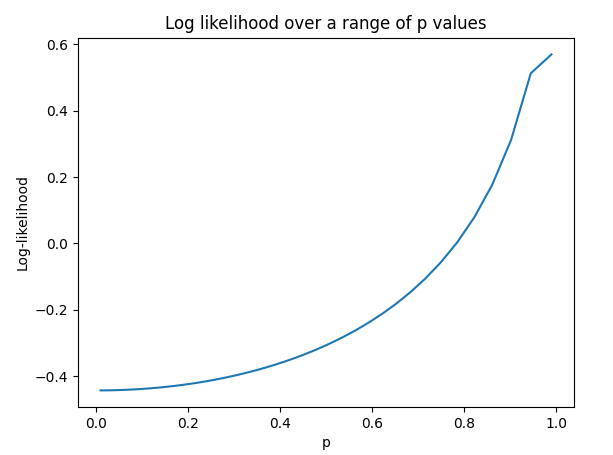
Now we evaluate function by code in file “eva\_func.py”

We can see max Ln(f(x1,x2))=0.307 when p = 0.943



Case 3: μx1= μ3= 0.8692 and μx2= μ2345 = AVG(μ1, μ2, μ4, μ5) = -0.1062

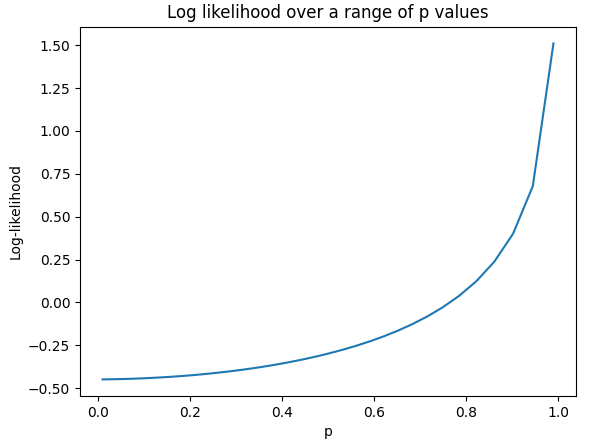
σx1= σ3=61.7618 and σx2 = AVG(σ1, σ2, σ4, σ5) = 22.6641



We can see max Ln(f(x1,x2))=0.571 when p = 0.999

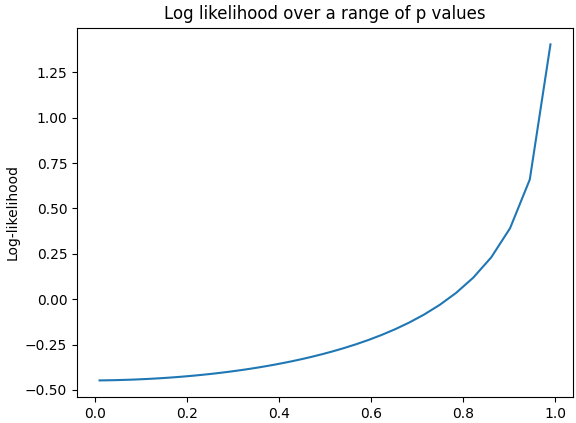
Case 4: μx1= μ4=0.03906 and μx2= μ2345 = AVG(μ1, μ2, μ3, μ5) = 0.1013

σx1= σ4=3.5352 and σx2 = AVG(σ1, σ2, σ3, σ5) = 36.9703



Case 5: μx1= μ5=0.37113 and μx2= μ2345 = AVG(μ1, μ2, μ3, μ4) = 0.0183

σx1= σ5=26.5203 and σx2 = AVG(σ1, σ2, σ3, σ4) = 31.47



Now we will do a for loop through each line of the data frame

We now compute the lost x1 where x2 is the average of the non-lost variables

Using equation with ln(fx1,x2), μ1, σ1, x2, μ2, σ2 already know to calculate x1

We do the same with all five critic columns

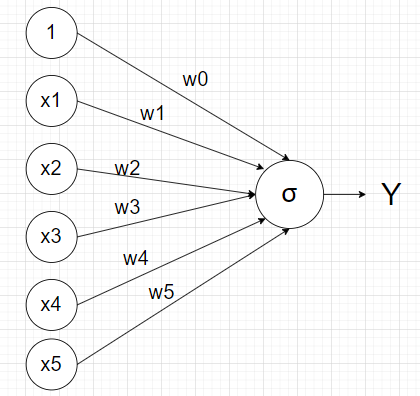
Because the time is limit so very sorry that I can’t done it

**Modeling – 2.1**

We can choose either **Linear regression** and **Polynomial regression**

Code for linear and poly is 2 colab (ipynb) file in the same directory

I choose linear regression to prevent the model from overfitting due to limited data.



The equation become Y= w0+w1\*x1+w2\*x2+w3\*x3+w4\*x4+w5\*x5

you can see that the model is quite simple because it only goes through 1 node but if you add more node or layer it will add activation => nonlinear

I had try to build model by hand using tensorflow keras but until now the model is not working. Maybe my code format is wrong, so sorry about that.

**Modeling – 3.1**

There are two conditions for me to think of deleting a column from this dataset, first is to calculate correlation and second is to check which column has more missing value

Open “Modeling3.xlsx” you can see this

Ảnh có chứa bàn

Mô tả được tạo tự động

Found that columns 2 and 5 have the highest R2 ~0.7, so it can be said that these are the 2 columns with the strongest correlation

So we can delete 1 of 2 columns 2 and 5

You can count that column 5 has more null values than column 2 so column 5 should be the deleted column