**598 Project 4**

**Weichuan Deng**

**112516608**

**I use Jupyter Notebook to do the project.**

**Introduction**

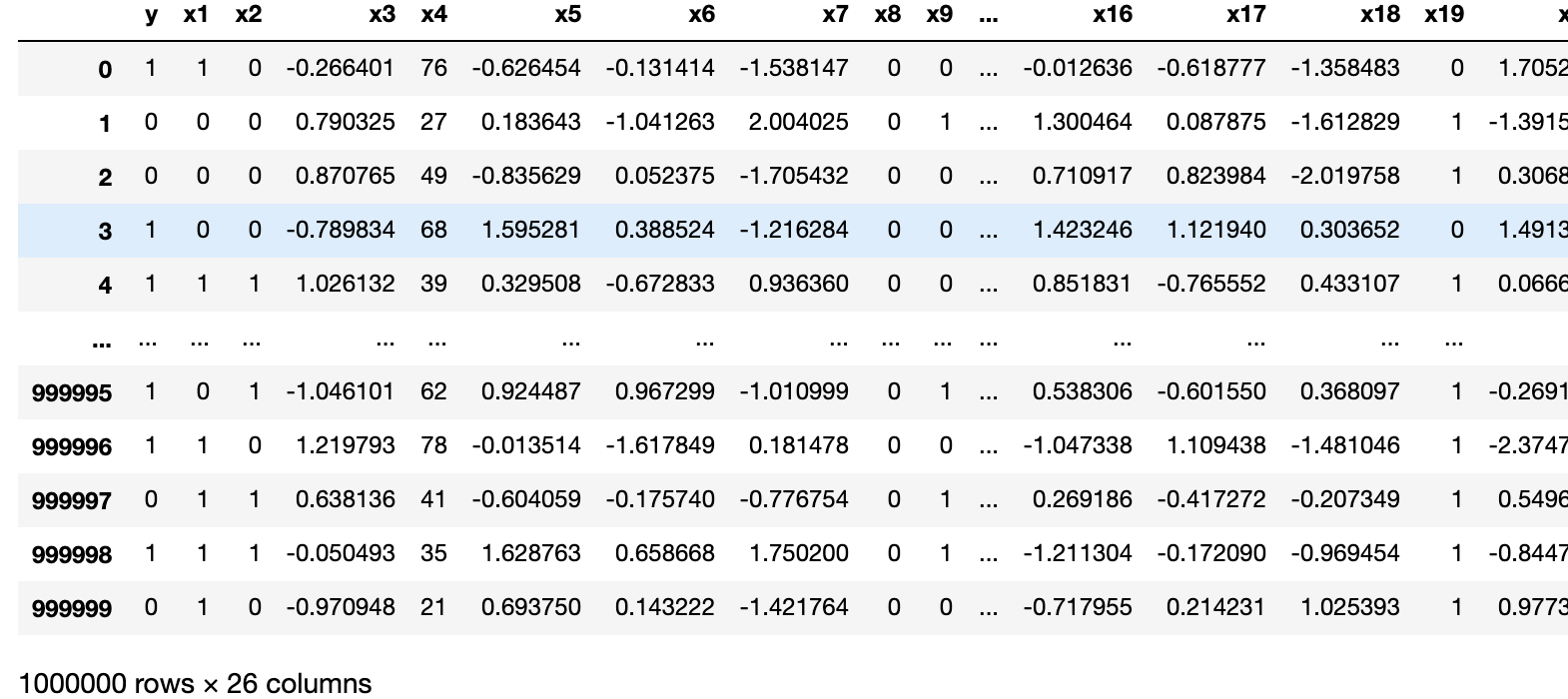
One big dataset has been partitioned into 10 chunks and are stored under the shared class space on SeaWulf: /gpfs/projects/AMS598/projects/project4

Each file contains one column called ‘y’, which is a binary 0/1 response variable, and 25 other columns named x1 – x25, representing explanatory variables.

Use the ADMM algorithm to run a logistic regression on all data, and obtain one set of consensus estimates for the coefficients of the explanatory variables.

Write a report about how you did the analysis. Submit your scripts with the report.

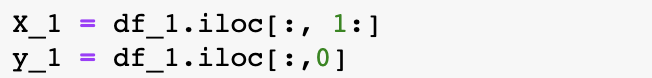
**Dataset**

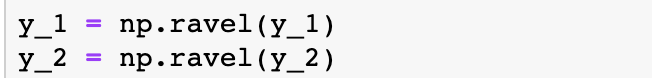


The first column is the dependent variable and rest 25 columns are features. For each data part, we have 1000,000 rows and 26 columns. And totally, we have 10 data part in the whole dataset.

**Data Processing**

At first, I separate the data set to 2 parts, the features matrix and the dependent variable vector.

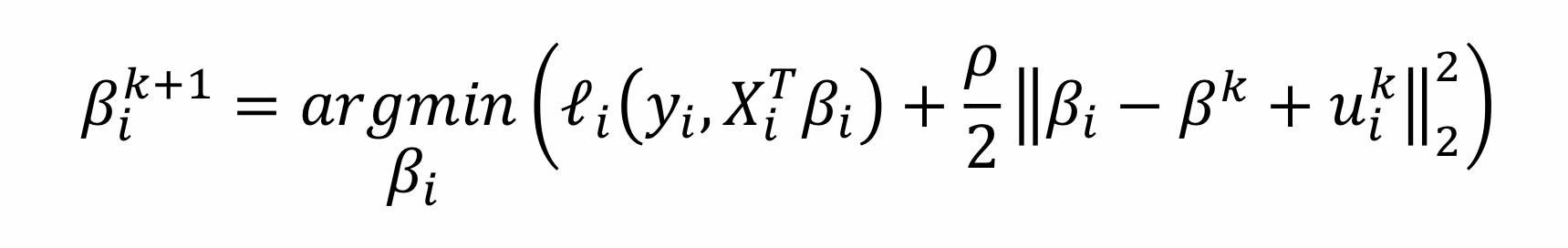




Then, add a “1” column to our features to get the intercept of the regression.

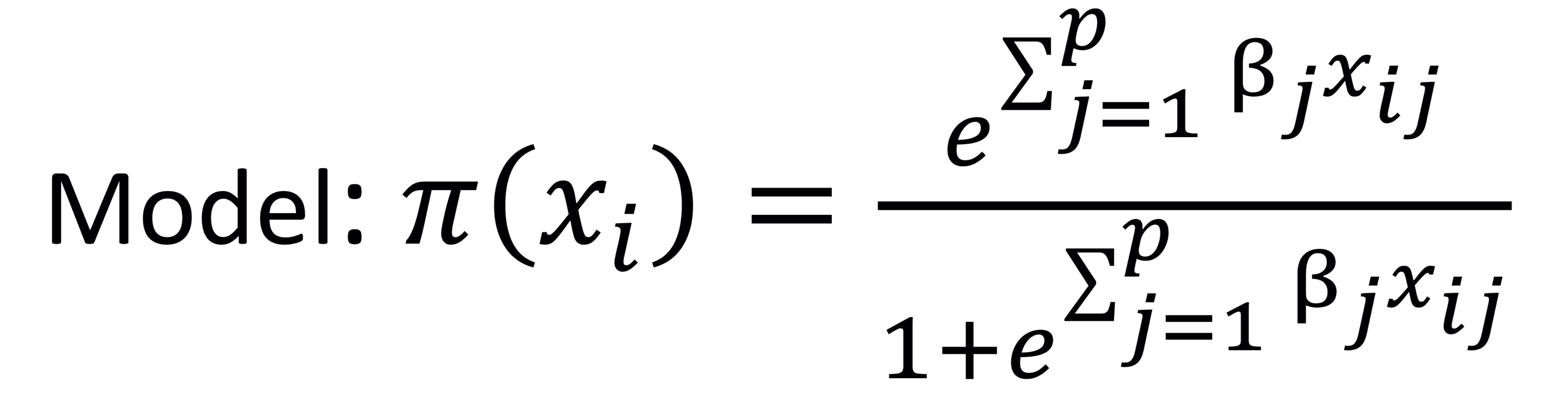
/var/folders/ww/x5b4hg0s1j7d8v5hzsjz2k280000gn/T/TemporaryItems/（screencaptureui正在存储文稿，已完成27）/截屏2020-11-11 下午3.38.50.png

Initialize our beta\_0 and u\_1, u\_2, …, u\_10 for the first iteration of the ADMM algorithm. And just let the .



**Model Build**

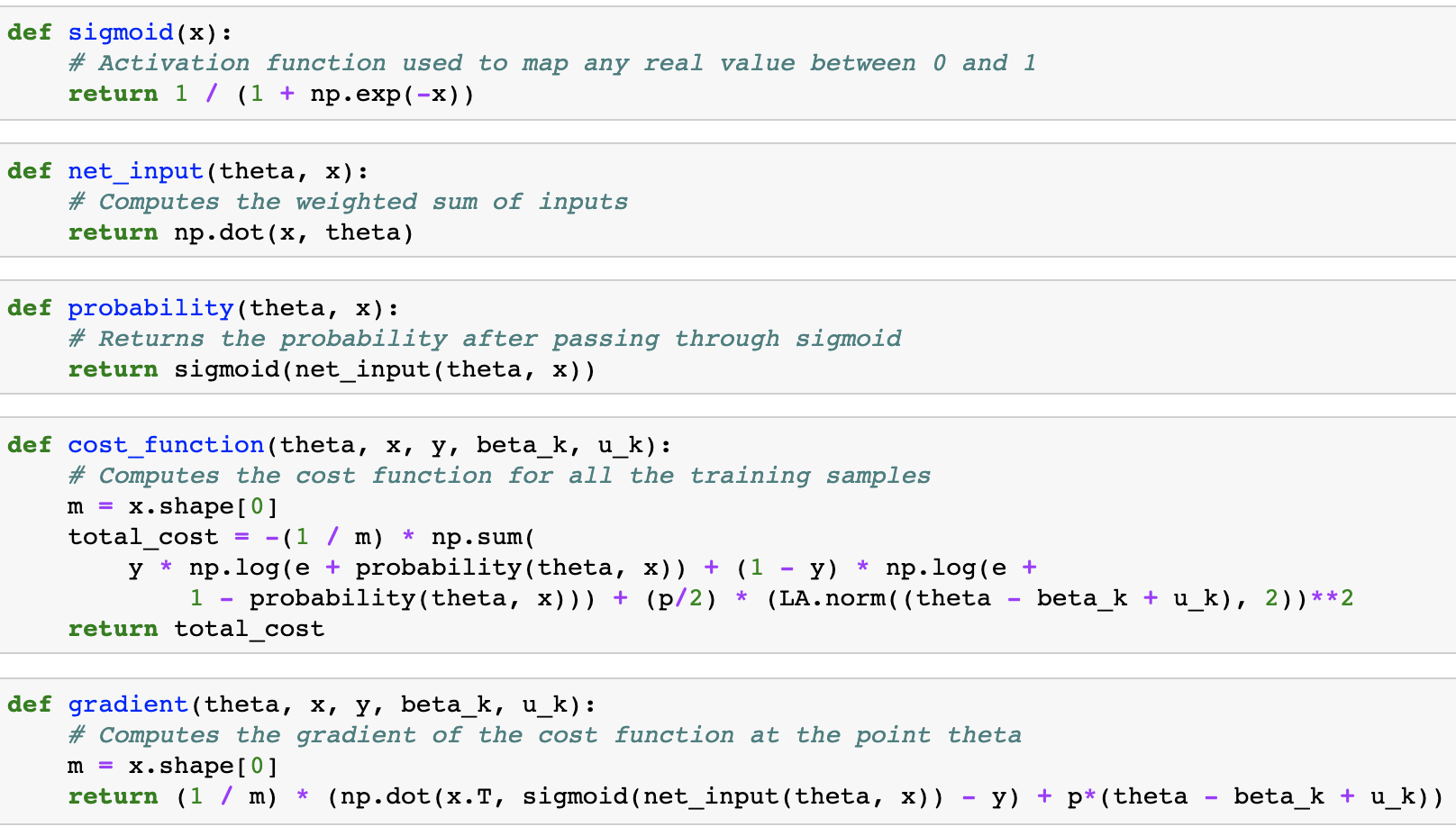
For logistic regression model.



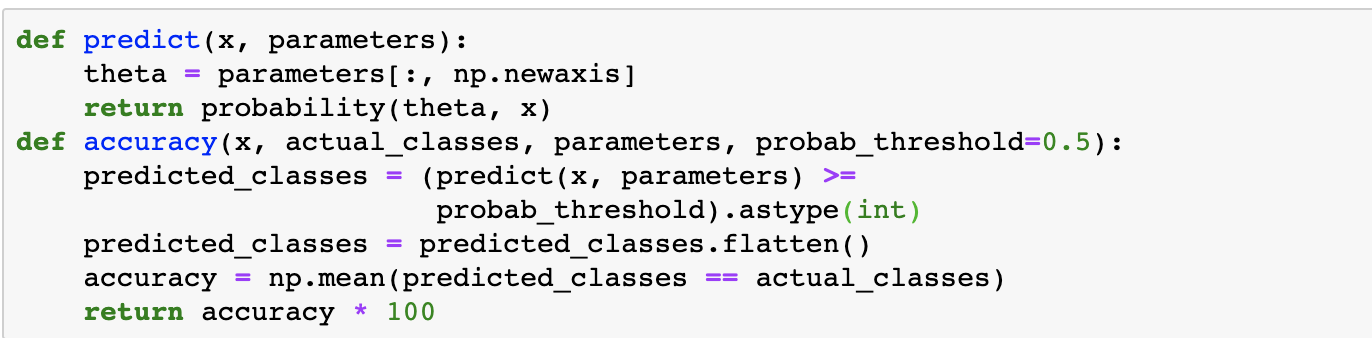
Then the negative log-likelihood cost function is:

Then the Lagrangian function is

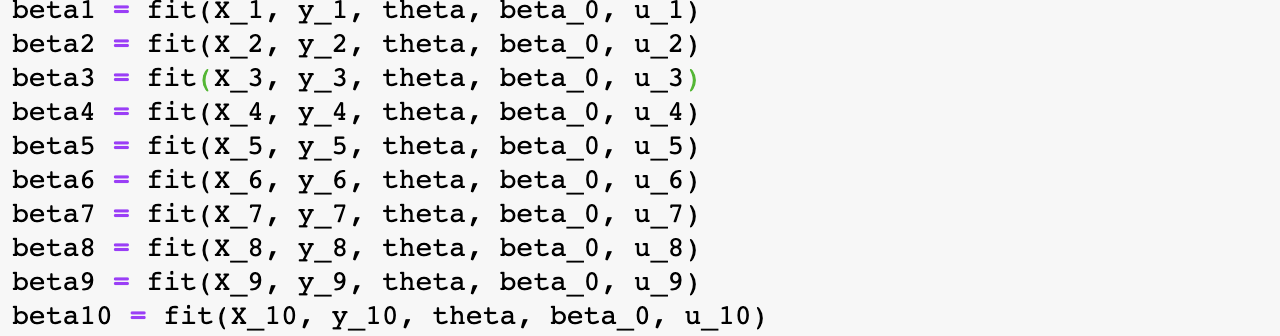
We will use gradient descent to minimize the Lagrangian function.



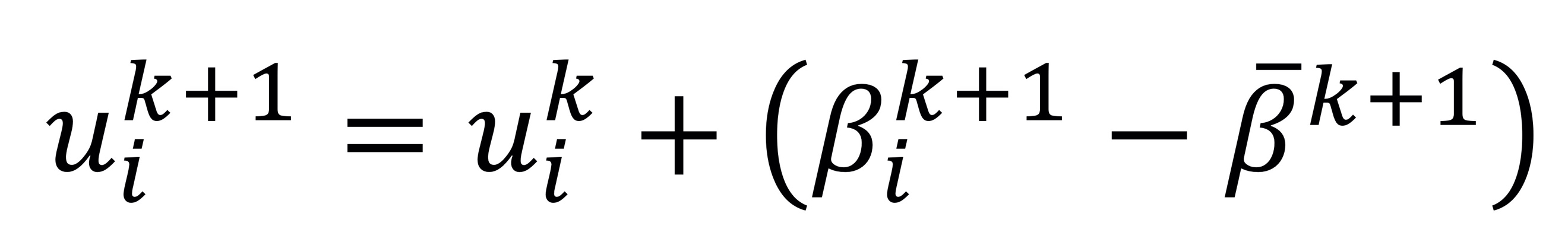
And define the accuracy function to test our model goodness of fit.

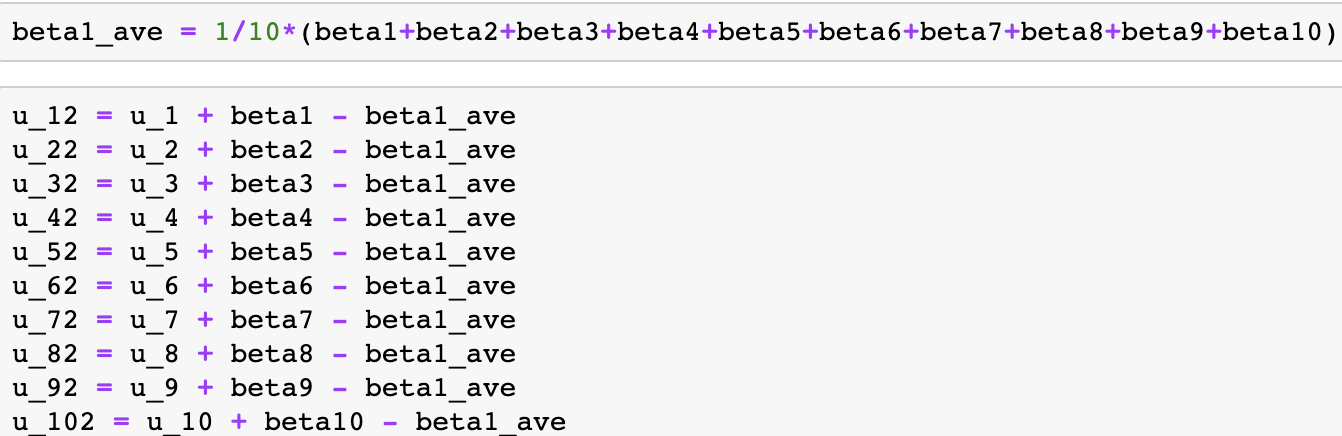


**Model training**

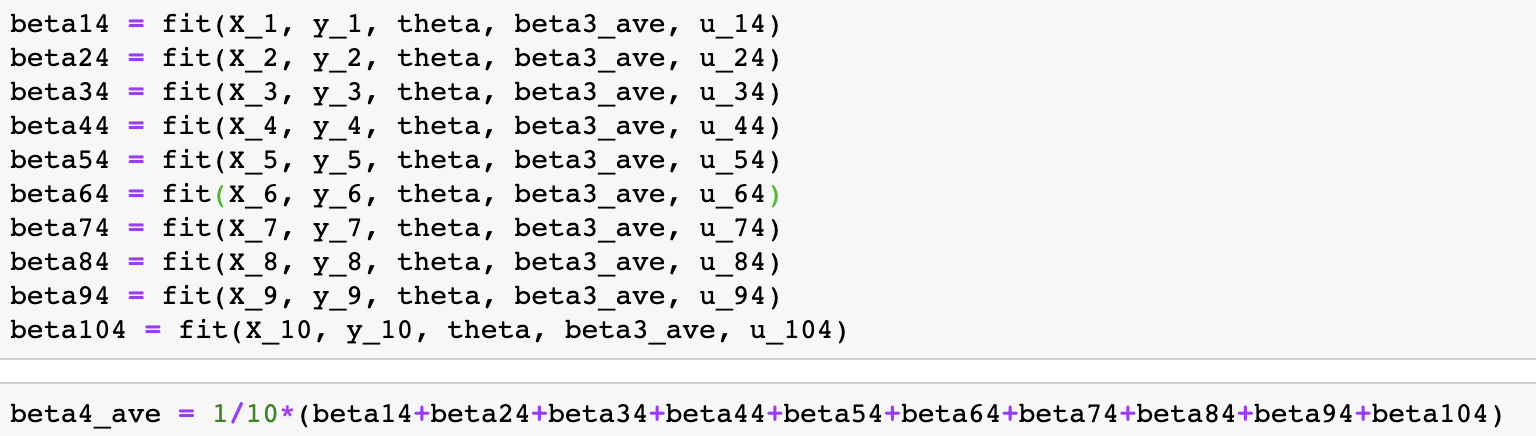
****

**Update consensus**

****

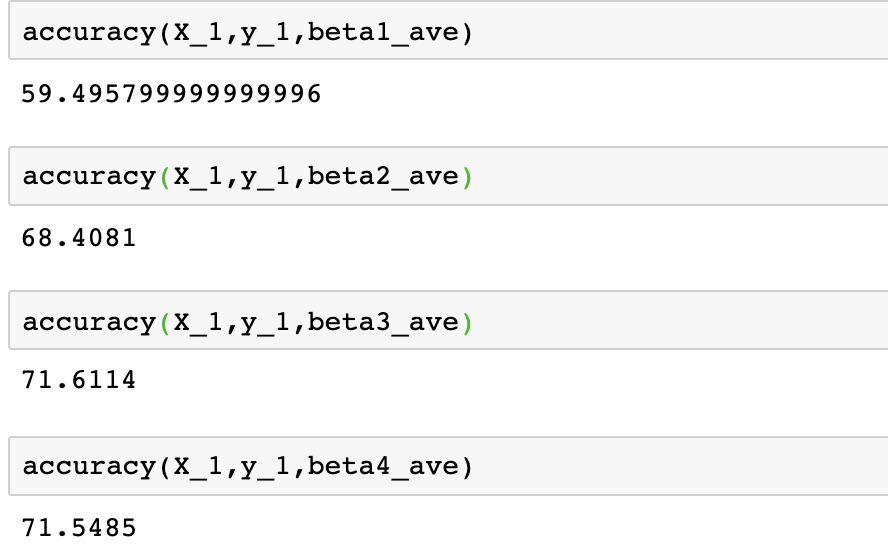
****

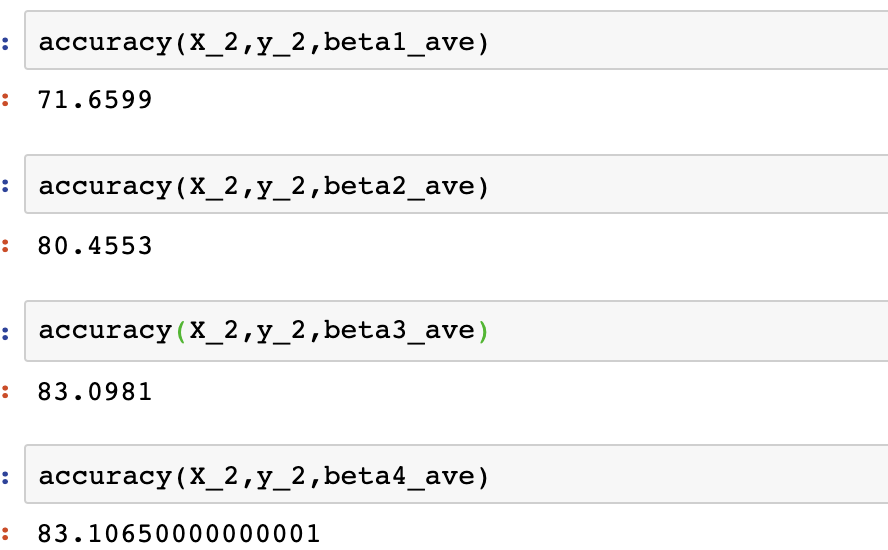
After 4 time iterations, we get our beta\_average 1-4.

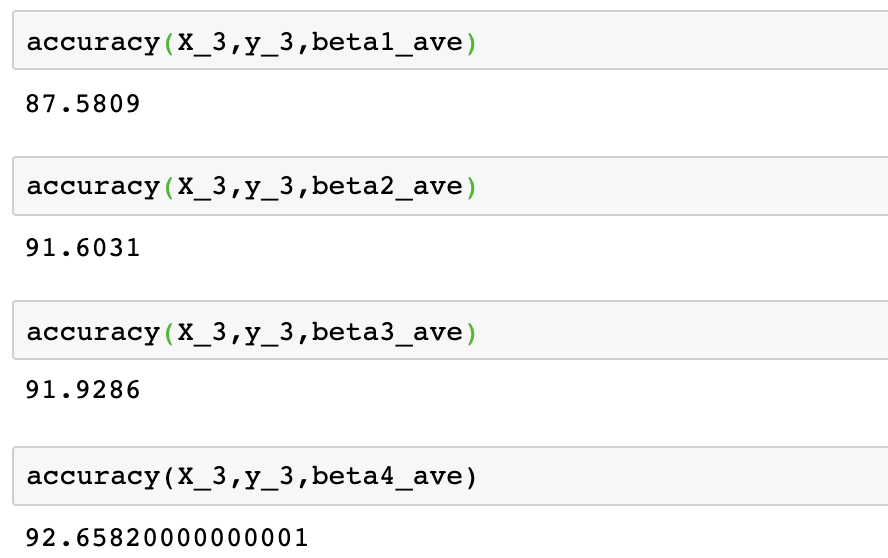


**Result**

From the accuracy of each iteration, we can see the accuracy rate is convergence.







Then, I get the result after 4 time iterations.

