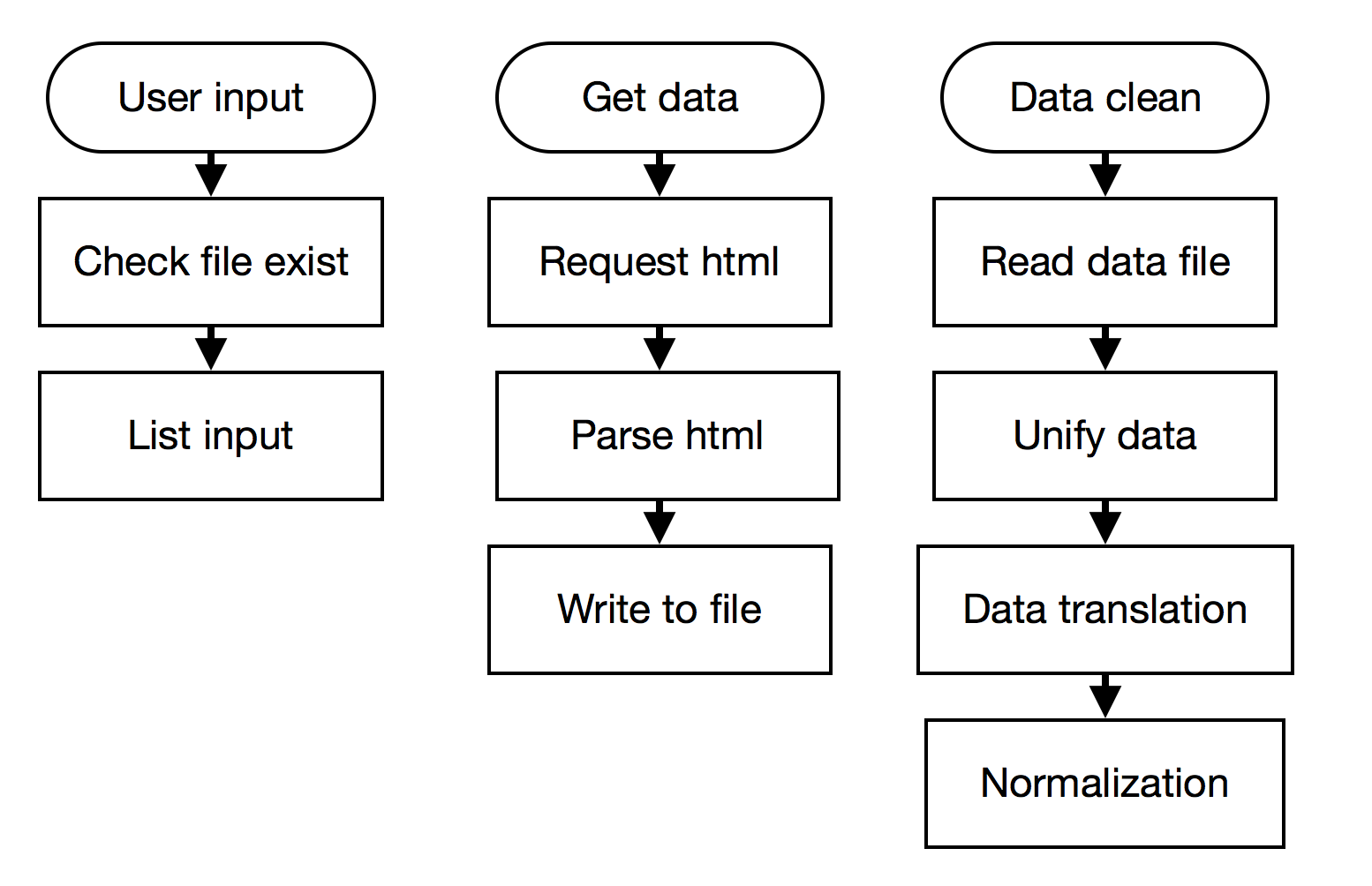
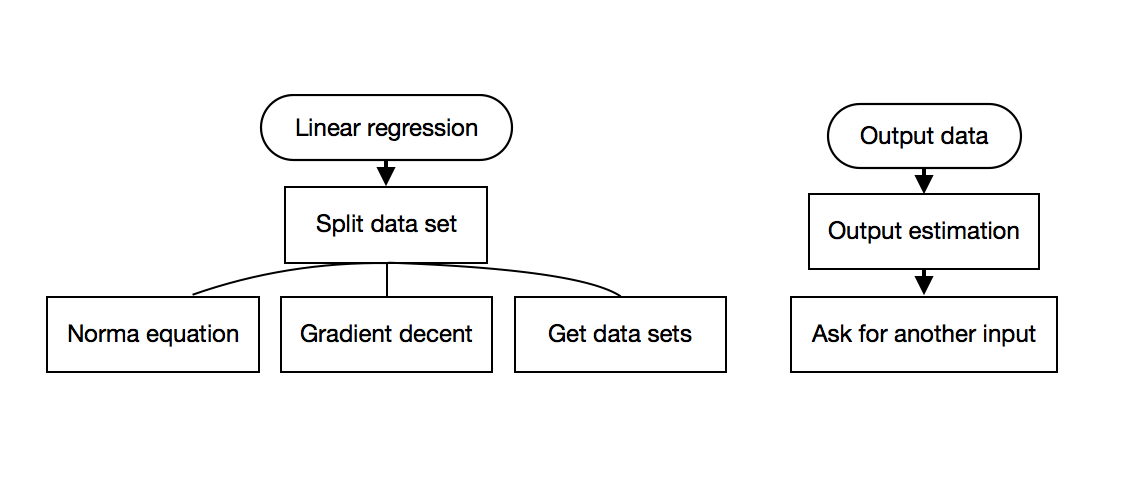
Criterion B: Design

General process:

Ask for information input -> Get data from the internet -> Data cleaning -> Linear regression algorithm -> Output estimated price

Break down of each:





Explanation of each part:

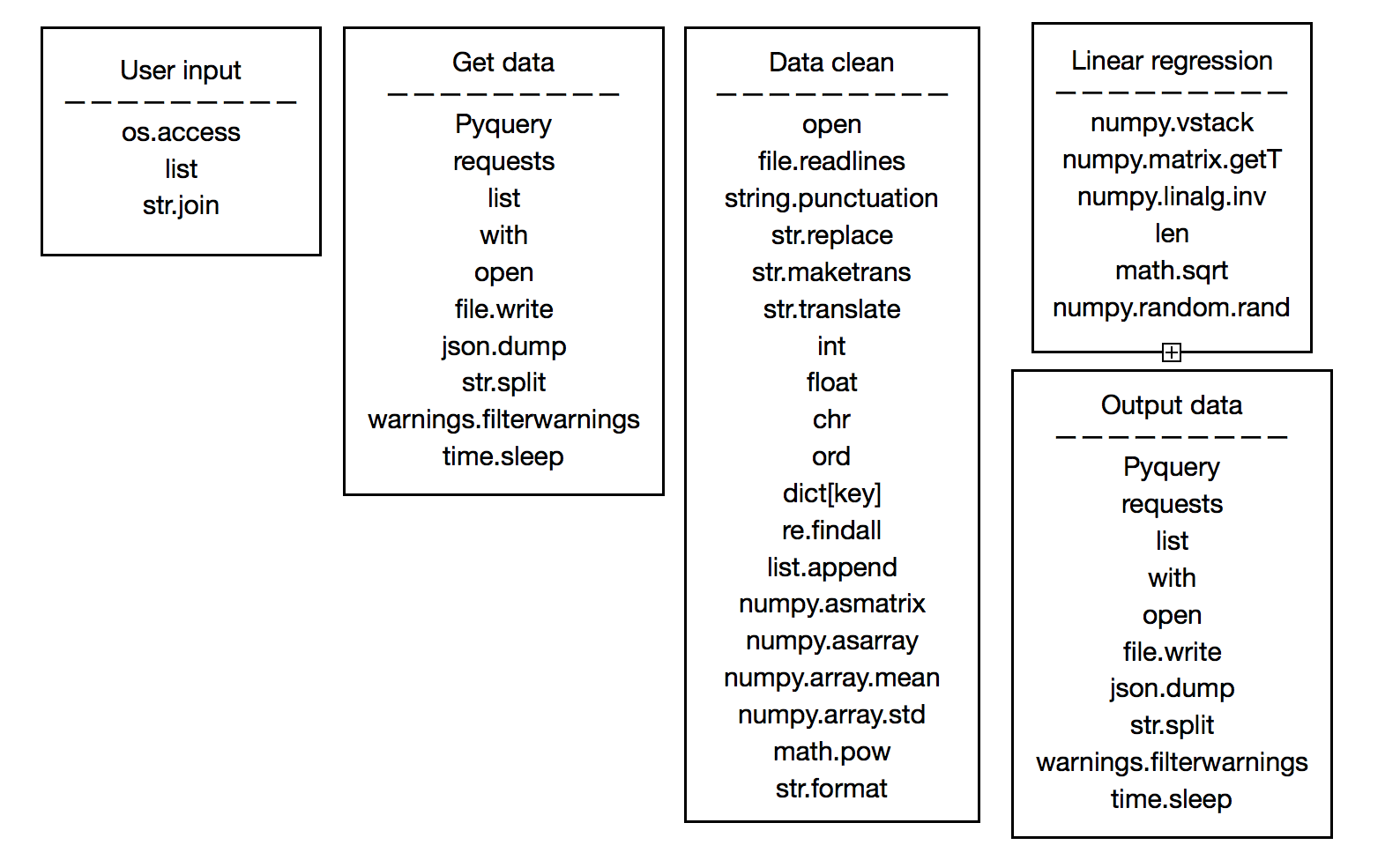
User input: Check if the data file exists, and ask for user’s input of features of the house which is formalized into specific data type.

Get data: Fetching the house data from the internet and save them to a .txt file.

Data clean: Read the result file data and eliminate all undesired punctuations within it, and turn qualitative features into numbers and normalize all data.

Linear regression: A class object that receives x matrix and y matrix and has the method of splitting the x and y into training set, testing set, and developing set, normal equation, gradient decent, and test function.

Output estimation: Output the estimation by multiplying the theta vector and the user input array and ask for whether to perform another prediction.

Functions and methods diagram:

Detailed breakdown:

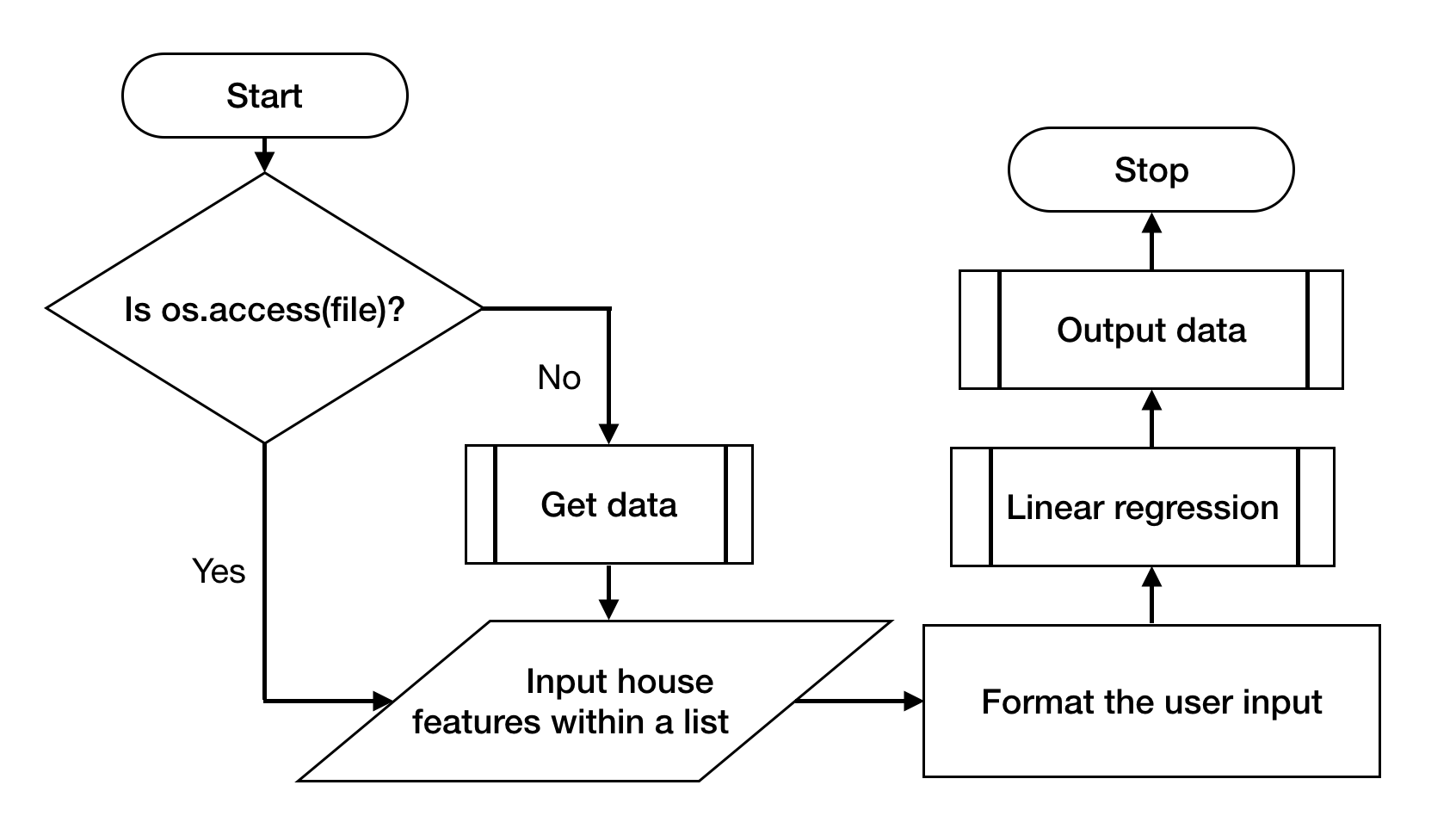


Figure 1 – The process within the User input

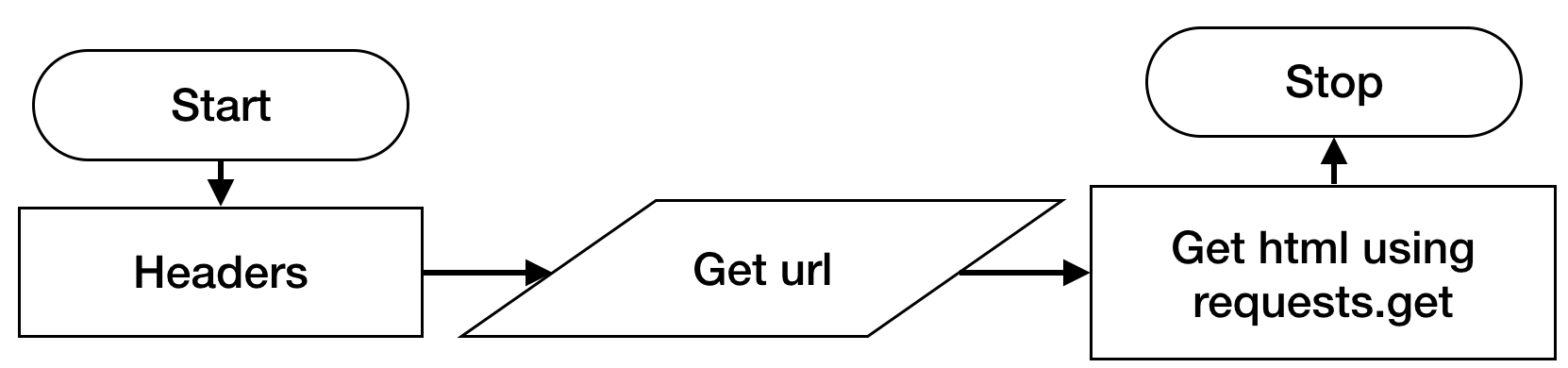


Figure 2 – Get html of the target url using requests module

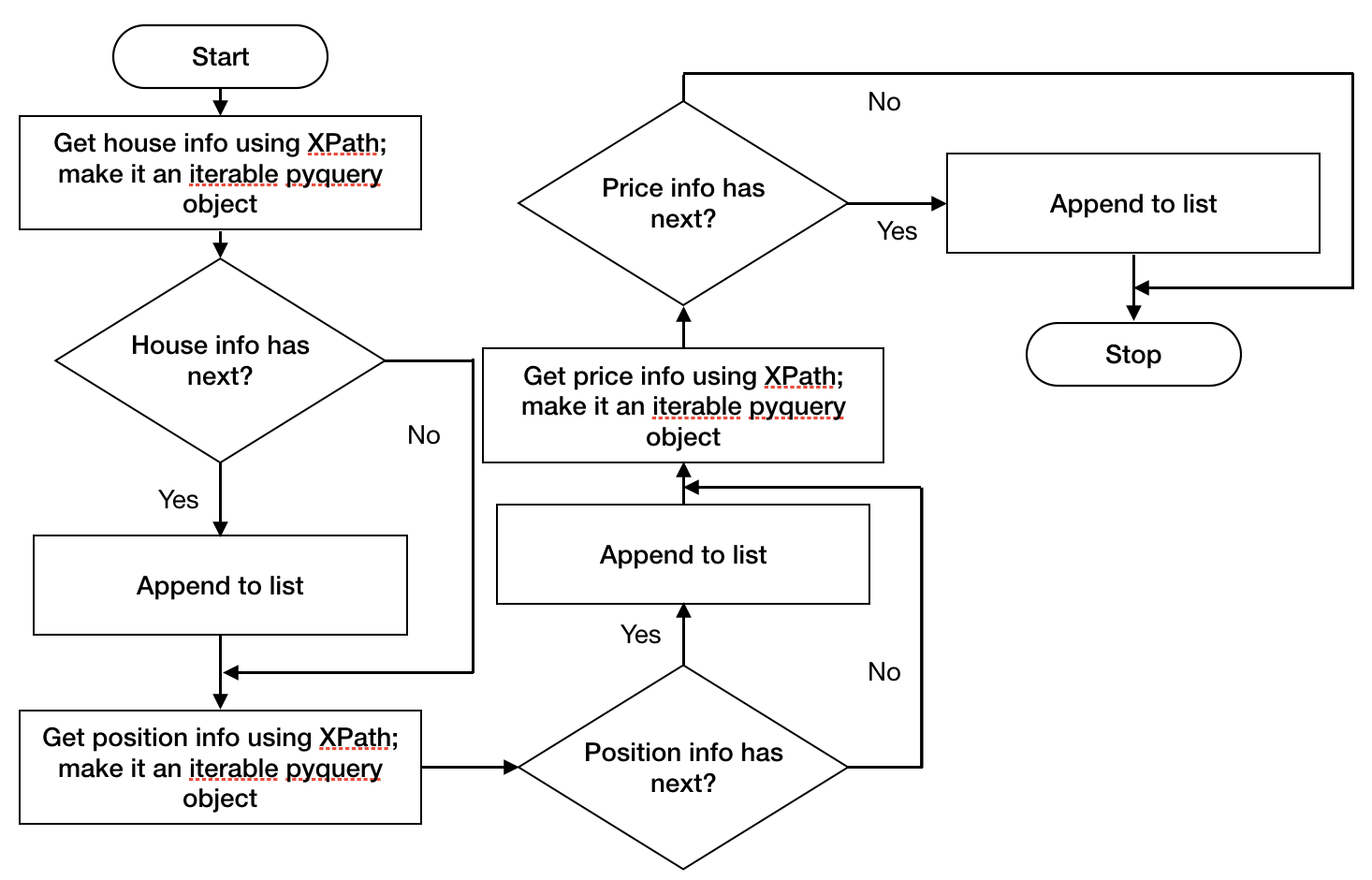


Figure 3 – Parse html received from Get html using pyquery and XPath selection

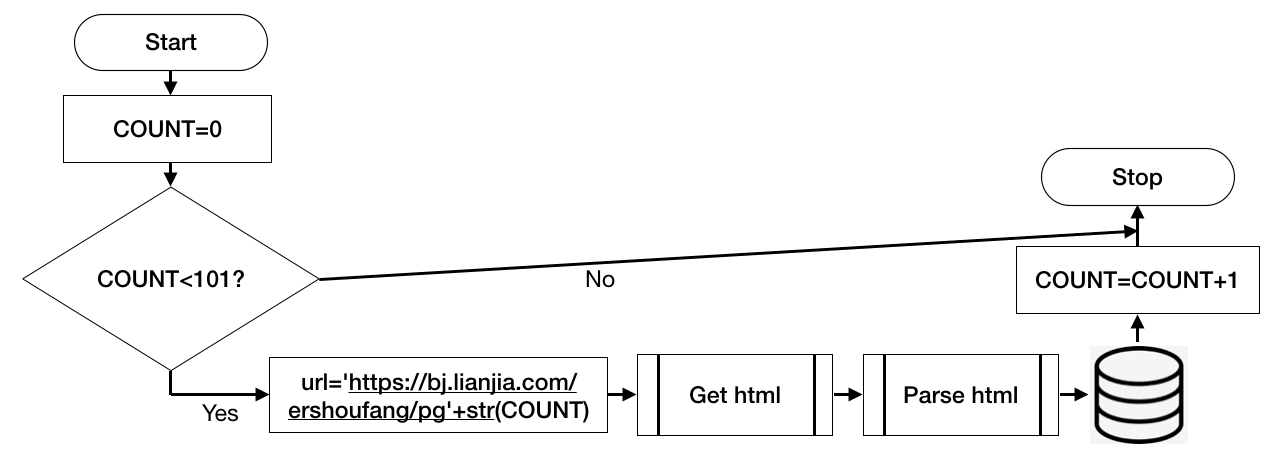


Figure 4 – The main lop for Get data section; write information to .txt file in desired format

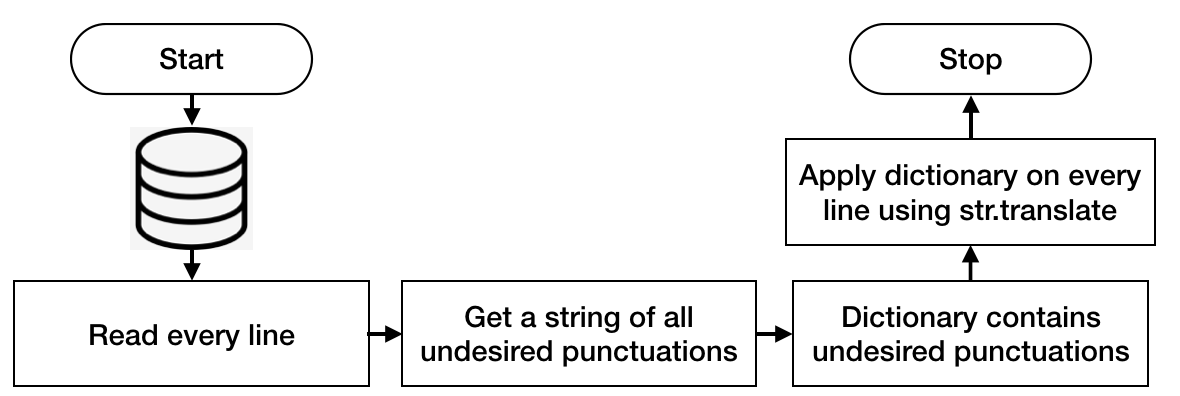


Figure 5 – Eliminating undesired punctuations and letter in data using dictionary and string operations

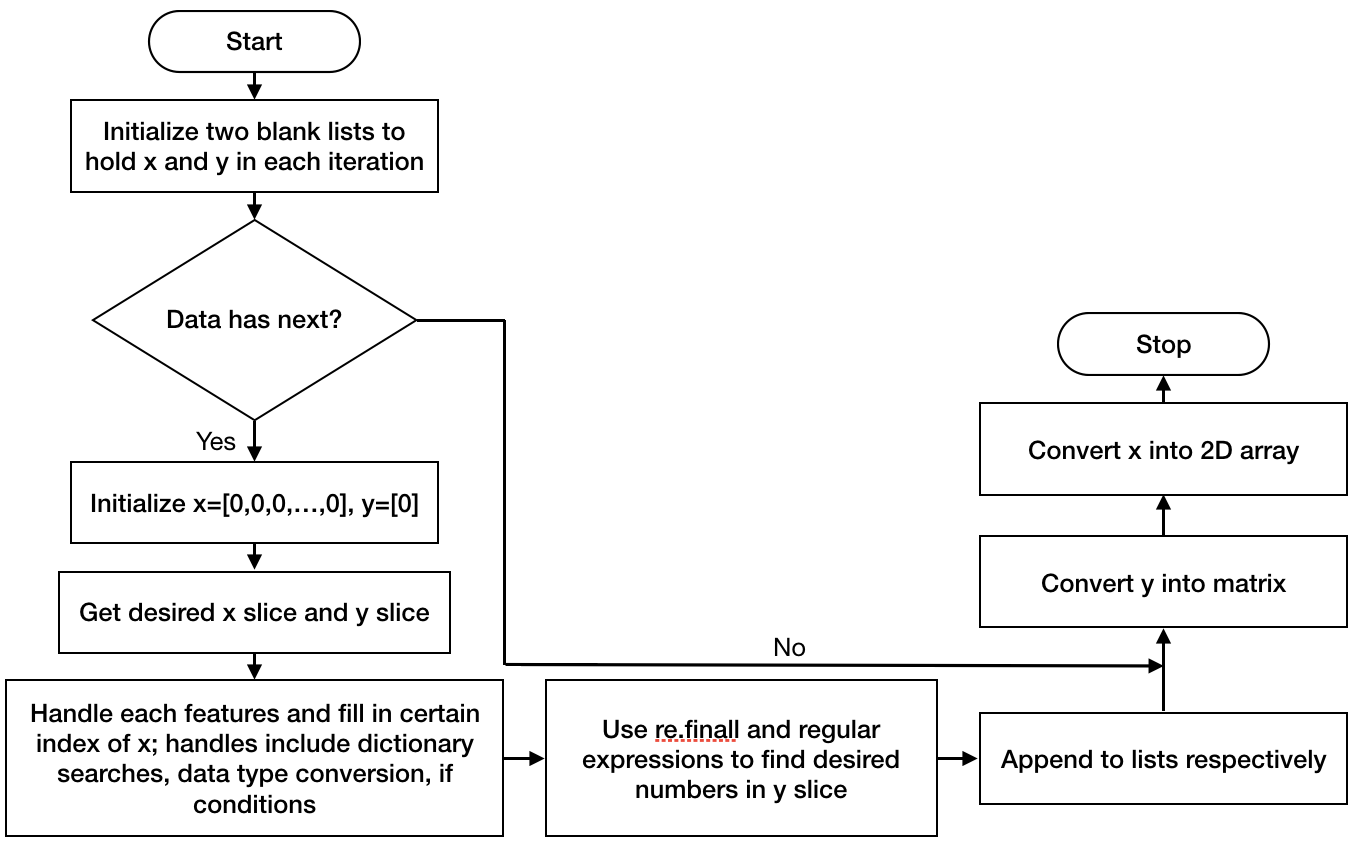


Figure 6 – Get wanted features and translate qualitative Chinese characters in x into numbers using string translation and re module

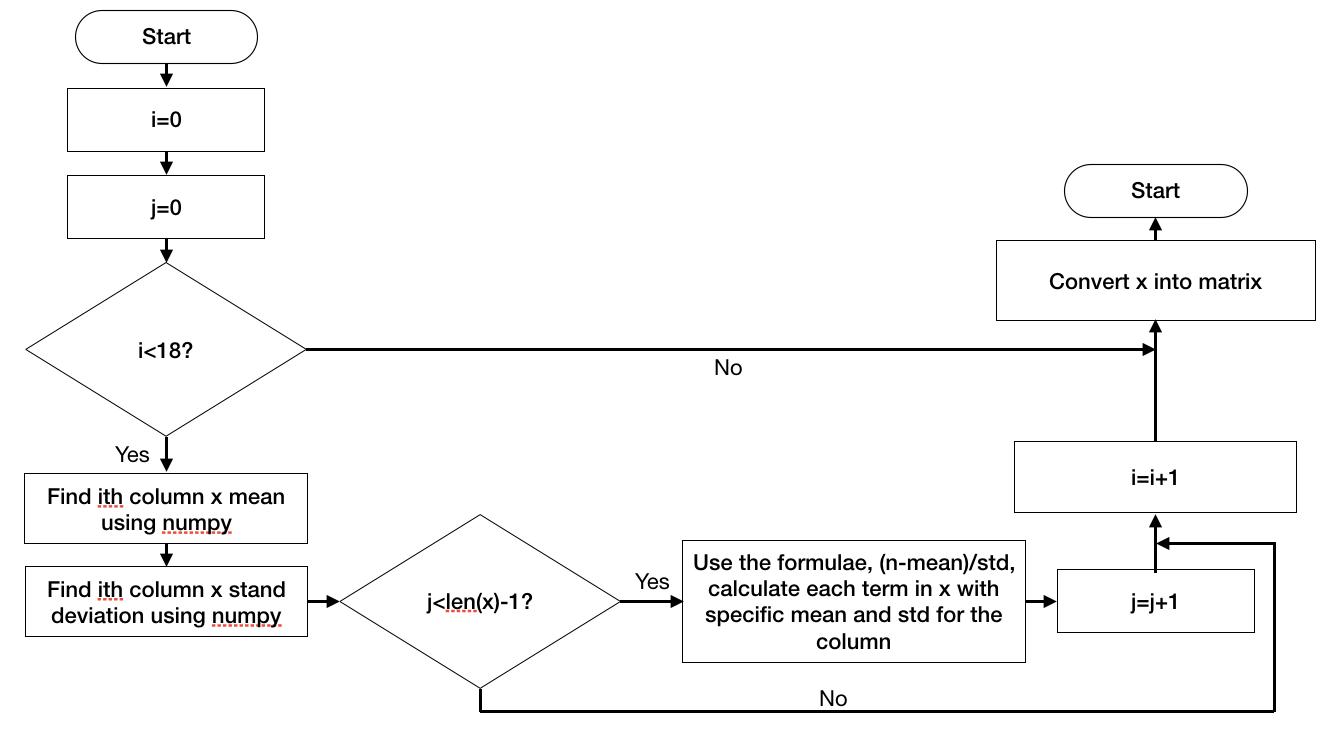


Figure 7 – Normalization of numbers in x, output the cleaned x matrix, using numpy

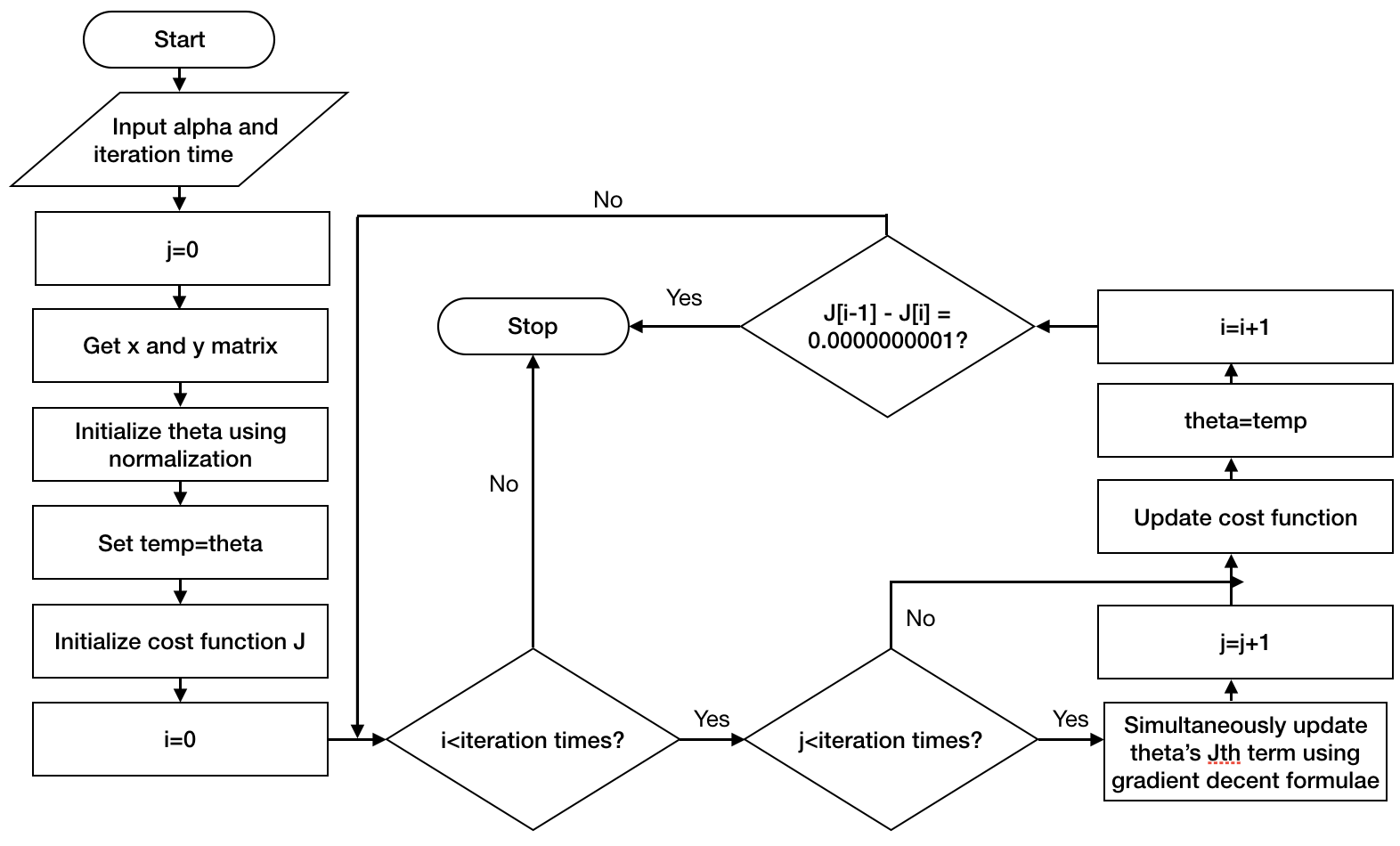


Figure 8 – Gradient decent within linear regression class, output the result vector, using the gradient decent formulae and the cost function formulae

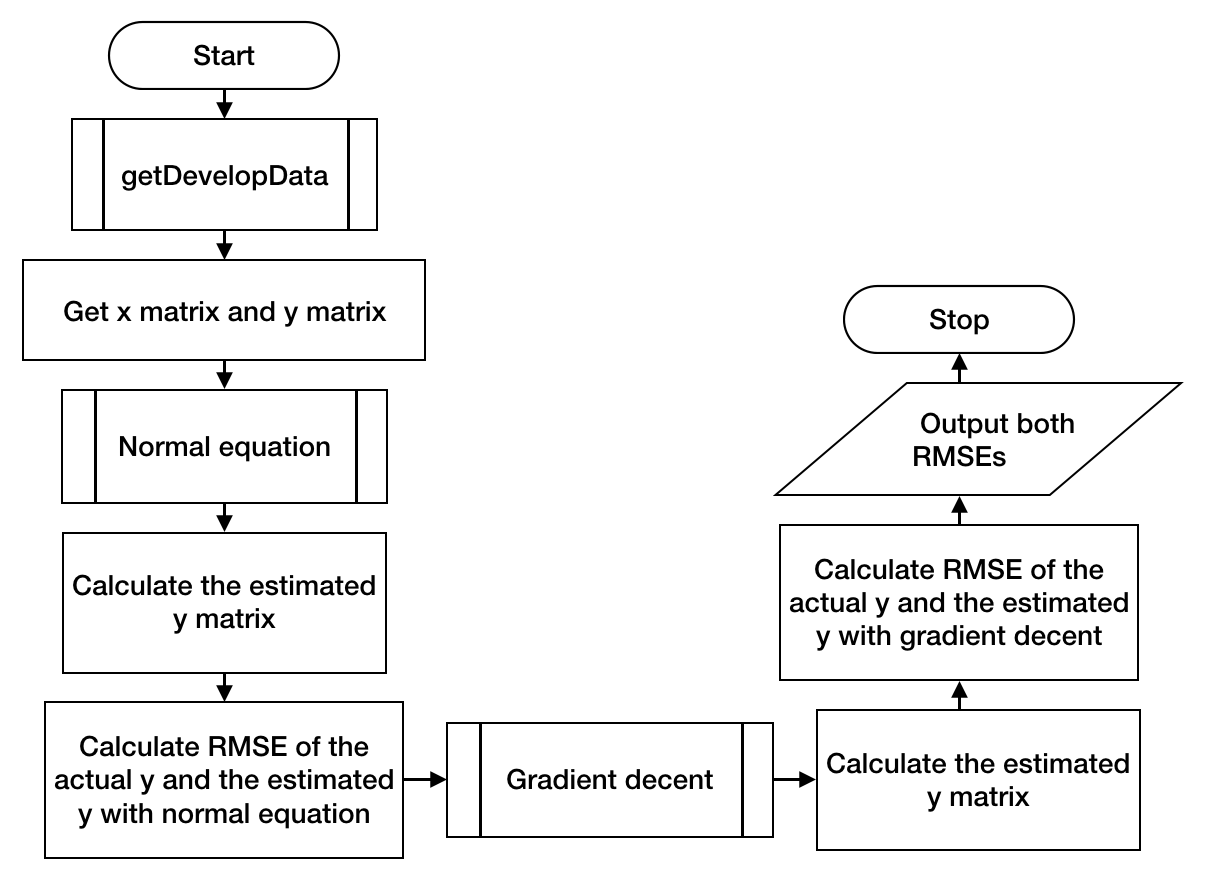


Figure 9 – test function within the class linear regression, using develop data and RMSE for both linear regression methods using numpy

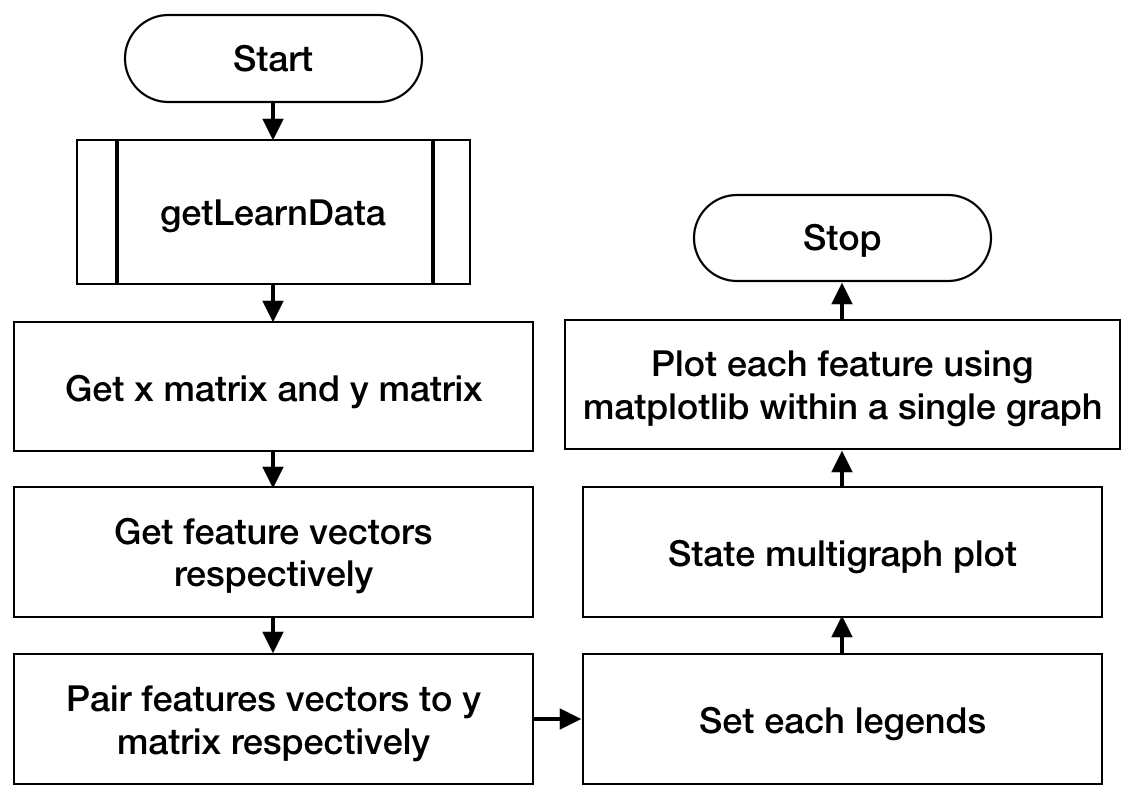


Figure 10 – Plot each features separately with y using matplotlib

Test Plan:

|  |  |
| --- | --- |
| Action to be test | Test method |
| Input can be saved in the correct format | Input different numbers and different features and check the correctness of each. |
| The html getter can report errors without halting the program while getting the html | Try getting html with poor internet connection and see how the error handler report the errors and whether it retries or not |
| The html parser gets correct information and the information is saved properly in the correct data format. | Run the parsing code, wait for result, and see whether the result file is correctly saved in the correct path with correct encodings. |
| The result file can be loaded and readline, and the punctuations are correctly removed | First tried the open statements and readline function, then use string operations and write additional output expressions so to see the correctness of the modified x. |
| The data cleaning process is successful with correct translation of all Chinese characters and numbers in wanted arrangement. | Add an temporary print expression so to visualize the result of data cleaning. Deal with the mistake accordingly to the float chart. Succeed if all data is correctly indexed in the x list and the output is in wanted data type and wanted size (matrix dimensions.) |
| The normalization is successful with all scattered data converted to closely normalized numbers | Run the normalization function and use print statements to visualize the normalized data. Succeed if all numbers are seemingly closer than raw data. |
| The linear regression class can be successfully initialized and the methods are all valid | Run through each method within class linear regression, succeed if all methods are perfectly working. |
| The gradient decent is working precisely. | Run gradient decent method in linear regression class, use test function to calculate the RMSE of the predictions. Succeed if the RMSE is low enough. |