I chose textdistance.jaccard from the textdistance library do calculate the similarity between two attributes. I tried to use the attributes from the "amazon_small.csv" file as references and loop all the attributes in the "google_small.csv" file to find the biggest similarity under the same idAmazon. Since I tried to compare the similarities of two products through their name/title, description, manufacturer and price, I gave each similarity a weight, which are 0.3, 0.2, 0.1, 0.4.

The point that I considered when giving each part a weight is that, if the product in both amazon and google are the same or similar, then this product's name or title in both stores will be the same or similar, therefore, the weight I gave on name is 0.3. And I have the same feeling when I tried to weight the price, since the same product in both stores will not vary too much, thus, the weight on price is 0.4.

The threshold I chose is 0.40. I tried 0.50 and 0.45 before I chose 0.40. I found that the result after set the threshold to 0.50 and 0.45 is a little bit less, therefore, I decide to use 0.40 as my threshold.

Since the Jaccard similarity will be affected by the length of the string, then if the same product's description in both datasets has a really huge difference in length, then the result of the similarity will be low. I think the use of hamming might improve the result.

I chose price in both dataset as the blocking key.

I realized that some of the prices' unit is "gbp", therefore, I wrote a function called "gdp_to_aud" so that I can transform the price from "gdp" to "aud". I did not drop all the prices that have a unit of "gdp" because I would like to get as many results as I can.

Then I transform all the type of prices to float(used to be object) and fill the missing values in the dataset with nan, so that it is for me to separate the prices and put the corresponding product into the corresponding bin later in this task.

I then created the bins that has a range from 0 to 60000 and their corresponding step size.

The last step is that, I treated the bins as keys and the products as values, therefore, I can put the products into the corresponding keys if the price is in that range.

I think the performance can be improved by rearranging the step sizes. If the range from 0 to 60000 has the same step sizes, then the result of the performance will vary.