Elements of Data Processing, 2020
Assignment 2 Task 1C: Report on Data Linkage Project
Name: Aoi Fujii

Scoring method: how the product comparison works

Preprocessing: Firstly, the name and the price of the product is preprocessed to remove unnecessary characters. Regular expression is utilised for this process. Any duplicate decimal points in the price and alphabetical characters from the price is removed. However, the character gdp was converted to the price in AUD by multiplying them by 1.86 which is the current. AUD/GDP rate (here I assumed the price in the dataset is in AUD).

Comparison function: Then, the similarity of each item name in google dataset is compared with the item names in the amazon dataset using token_set_ratio() from fuzzywuzzy. The token_set_ratio() separates each word in the item name and put them into a set. This prevents matching titles with many repeated words such as "v v v v v cap" and "book v v v v v" which are obviously different product from inspection. In addition, the price difference of each item is calculated by subtracting price in google dataset from the price in amazon dataset and taking its absolute value. All the scores are stored in a data frame with corresponding idAmazon and idGoogleBase.

Final scoring function: The price difference in the data frame are normalised by using MinMaxScalar(). The similarity score is converted to a distance measure by subtracting 100, taking its absolute value, and dividing it by 100 as the similarity score is in between 0 to 100. The normalised price difference and the distance score are weighted and added together. The total score is normalised and stored in the data frame in the ascending order.

Threshold: For each matching pair in the sorted data frame, all the matching pair below 0.1 distance score is selected. As I stored the pairs with more than 66% similarity of their names for the efficiency purpose (use less memory, shorter execution time), the recall and precision is consistent for the threshold over 0.013. The threshold was found iteratively using for loop for this method.

Evaluation of the product comparison:

Overall, this algorithm gives 0.938 recall and 0.968 precision with 122 TP, 22066 TN, 4 FP, 8 FP for this sample. Thus the performance of the product comparison is considered high with higher proportion of true matches in the pairs classified as matches and higher proportion of true matching pairs that are classified as matches. For this task I removed manufacturer on purpose as most data in google dataset does not contain information about manufacturer which makes is less appropriate as a similarity measure. The product name and price are used as a measure since they are clear and precise. In particular, names are shorter than descriptions thus it contains more similar words for the same product. However, this algorithm could be further improved by taking into consideration about description. Even if the description varies for the same product, it can be used as a measure by a proper preprocessing method such as tokenisation and lemmatisation, as well as by weighting the score of each measures. Furthermore, the price preprocessing could be also improved by confirming the unit of each prices as it is not clear whether the data is in AUD or GBP.

Blocking implementation:

Firstly, the name for each item was preprocessed to remove any unnecessary characters. Then, each name in google datasets and amazon dataset are vectorised using count vectorizer() from sklearn.featre_extraction.text so that similar words can be expressed in vector values. I also set the word boundaries aware variant, char_wb which creates n-grams only from characters inside word boundaries. This improves the accuracy while retaining the internal structure of the text and retaining the robustness to misspelling. Then the similarity of the two vector sets is calculated using cosine_similarity from sklearn.metrics.pairwise. Then the pairs from google vector set and amazon vector set (in index form) that have more than the threshold (0.45) for their similarity score is found. (Here, higher score means having more similar words in common). Lastly, idAmazon and idGoodleBase is found from the pair of index and each item in the selected google datasets is assigned to the same block if they have the same amazon item as a pair.

Evaluation of the blocking method:

This method uses product name as a measure as this is more likely to show the precise similarity. (From the truth.csv the price difference differs from 0 to 141342 which is not an accurate measure of similarity). The vectorising method used also improves the performance in terms of accuracy as it allows us to compare the similarity of all the names from different dataset to categorise them into blocks. This vectorisation method also makes the time complexity linear (the execution time is 0.00347s for this dataset). The performance is 0.884 pair completeness and 0.998 reduction ratio. Thus, this method reduced the data into smaller dataset and allocate them into blocks with matching pairs. There was a small number of FN but the FP can be further decreased by adjusting the number of blocks and number of feature to take into consideration. In this method, all the selected amazon items are allocated to each individual block and the google items are assigned to the block for those that have the same amazon item as its pair. This means all the selected pair of items are put in the same block sorely by the similarity of its product name. Thus, further analysis of the description and manufacturer could reduce the number of items in each block that increase the FP. Price could be also analysed but must have been weighted carefully as the measure is not likely to be accurate.