**Task 1a - Linkage Method** The product comparison works by first filtering out definite matches and then selecting matches from possible matches based on thresholds. Names of each dataset are preprocessed via punctuation removal, tokenization, stopword removal and lemmatization, but columns containing original names are also retained for pattern searching. The columns are alphabetically sorted based on their names for easier handling.

Manual inspection shows that two true matches have identical models (e.g. ‘IM600USB’) in their names (Assumption 1). However, the positions of model codes in two datasets are different. Hence, regular expression with the pattern r'(?<= - )[A-Z0-9]{3,20}' was used to search for model codes in unprepossessed idABT names, while four patterns <r'[A-Z0-9]{4,20} -'>, <r'[A-Z0-9]{4,20}$'>, <r'(?<= )[A-Z0-9]{4,20}'> and <r'(?<= )[A-Z0-9]+-[A-Z0-9]+'> are used for idBUY names. The string (model codes) obtained for each record in Abt is compared with strings of each record from buy, and matched records with identical model codes were selected out as definite matches.

Pairwise comparison was conducted for the remaining records and the cosine similarity score between their names was obtained. For each record in ABT, the record in BUY that has a cosine similarity > 0.5 were added to a list. This list was reverse sorted and the top 3 most similar BUY records were retained as possible matches for the ABT record, with the rest discarded. The cosine similarity was used because it measures similarity irrespective of different lengths of product names, and the threshold 0.5 was chosen since value < 0.5 has low similarity and can be considered as definite non-matches.

Finally, possible matches for each ABT record are inspected using the regular expression pattern <r'[A-Z0-9]+[0-9]+'> to search for strings in product names that contain both capital letters and numbers. To be a match, record names should have at least one such common substring, requiring the ‘intersection’ score calculated using Equation 1 to be >= 1. The final ‘match’ list is a collection of definite matches and matches found in possible matches.

**Task 1a - Linkage Evaluation** A low recall value, 0.725, indicates missing of true pairs potentially due to 1) high threshold 2) wrong assumption about true pairs. A relatively high precision ‘0.872’ indicates good selection of definite matches. Due to manual inspection, this method is prone to misunderstanding and subjectivity.

To reduce the false negative pairs and increase recall, we can select an optimal threshold that also maintains accuracy. This can be done by fitting a regression model that computes the F score for each threshold within a reasonable range, then the threshold with the greatest F score can then be selected and used.

Matching can also be done by machine learning to increase efficiency and accuracy. After preprocessing texts to numerical bag-of-words representations, decision trees can be used to classify records by splitting records based on different attributes. Splitting criteria like Information gain can be used for selecting the best attribute. Two records within the same leaf node can be matched if the purity exceeds a threshold.

**Task 1b - Blocking Method**  This method is a two-pass blocking. Firstly, based on Assumption 1 in Task 1a, regular expressions were used to extract model codes for each dataset using patterns described in Task 1a. The common model codes found across the two datasets were used as blocking keys. For the remaining records, manual inspection shows that the first word in the product name attribute in two datasets consistently indicates its manufacturer. Hence, the first word of product names was firstly preprocessed (via lemmanization, stopword removal, case lowering and punctuation removal), and then used as blocking keys. Pairwise comparison between blocking keys in Abt and Buy was conducted, and identical blocking keys were identified and used as final blocking keys. Records in two datasets were separately allocated into blocks based on the identical blocking key that they share.

**Task 1b - Blocking Evaluation** The blocking method worked relatively well (Table 2). a high reduction ratio of 0.986 means the method significantly reduced the execution time when compared to non-blocking linkage, which has a time complexity of O(n \* m), where n=n. of records in Abt, m= n. of records in Buy. In this method, the time complexity is O(m\*n/b), b = n. of blocks, since the average block size is 7.33 that indicates quite even distribution of records.. Yet, the dominant block size, 72 x 73, still requires 5256 comparisons. To improve, a multi-pass approach can be adopted. The records within the big blocks can be divided into smaller blocks using other blocking keys (e.g. price range, product types). This way, small, equal-sized blocks can be obtained to further increase time efficiency.

A PC of 0.933 indicates relatively high accuracy. To reduce the chance of matched pairs not allocated to a common block (false negative counts), the records can be assigned into multiple blocks. This especially suits the given two datasets due to the large difference between their ‘description’ column. This way, when matched pairs do not have the same parts of the records in common and are missed by a blocking key from the ‘description’ column, another blocking key (e.g. from ‘price’) can be used to capture them.