I. Code:

The code for my solution is at: https://github.com/susiehuynh/CS4400

II. Solution Outline:

The solution includes five steps:

1. Data reading and EDA
   - In the first step, we read three tables: left, right, and training. We read the data set as pandas in order to help understand all the tables and design the solution. In total, we have 56376996 pairs (2554 rows for left table and 22074 rows for right tables). In order to efficiently examining every pair, we need blocking step to remove unnecessary pairs and reduce number of pairs that needed to be handle.
2. Blocking (by Branch and Modelno)
   - In this step, we perform blocking on two attributes: “brand” and “modelno.” Both blocking attributes share same process where two ids in each pair share the same brand and modelno. By doing so, we could likely to get two products with the same brand. This will secure that they are in the same entity. Similarity with model number. Next, intersect pairs between two blocking methods will be appended into a new list called “common.” This combination reduces the number of pairs from 56376996 to 3287.
3. Feature engineering
   - For each pair in the common set, we would apply the Jaccard similarity and Levenshtein distance of pair on five attributes in left and right table to generate a feature vector of 10 dimensions. By doing so, we obtain a feature matrix X_c for the common set. Then, we repeat the same process to those pairs in training set to obtain a feature matrix X_t. The label for the training set is denoted as y_t.
4. Model training
   - With this project, we use a random forest classifier method to train the model on (X_t, y_t). Because the number of non-matches and matches are imbalance in the given training set, we must set class_weight="balanced" in random forest to handle this training data problem. Next, we perform prediction on X_c to get predicted labels y_c for the common set.
5. Generating output
   - According to the indication, pairs with $y_c = 1$ are those predicted matching pairs in our M set. Since we want to exclusive any matching pairs in training tables, we will obtain a new predicted matching pair M^- set where all matching pairs in training tables get filtered out from M. Once the process complete, M^- will be exported to CSV file called “my_output.csv.”