Name: Mayank Baheti

Topic: Fuzzy Software Product Name match with G2 Product Names.

Fuzzy matching is a method that provides an improved ability to process word-based matching queries to find matching phrases or sentences from a database. When an exact match is not found for a sentence or phrase, fuzzy matching can be applied. Fuzzy matching attempts to find a match which, although not a 100 percent match, is above the threshold matching percentage set by the application.

Data Description:

Data is provided in the following data sheets:

1. **Software Catalog**: List of Client's software name and respective websites
2. **G2 Catalog**: List of G2 software name and respective websites

Output: Matched_Product.csv, which has the matched products.

The data that we have has some software and websites, which has been taken over by other companies, hence there are some outdated records as well.

Problem Statement:

G2 often needs to match its software catalog with software catalogs of customers and data vendors. This can be problematic when the software products are listed slightly differently in the two sets of data. Also while comparing the websites like Facebook, G2 Catalog has different pages for each of its vertical, and Client's data had only the domain name associated, so while doing the partial match as well, we are getting 100% match of each of these vertical's web urls.

Approach:

1. So when we need to match a list to products with other list of products, the first thing which comes to my mind is Fuzzy match of these two list. Python has a good implementation of fuzzy match in module called 'fuzzywuzzy', basically it uses Levenshtein Distance to calculate the differences between sequences.

2. Later we read the two data sheets, one for the Client Data catalog and other for the G2 Product Catalog, which will used further for finding similarity amongst these names.

3. The first step in terms of preprocessing was to bring the websites in a standard format, as some had HTTP and some had HTTPS, also there were some instances where these websites had www, and some didn't had it. So, I made a function '**remove_prefix_website'** which removes all these unwanted prefixes and get only the domain name and pages.

4. Now after this we had to compare the names of software, for this we have made a function **'match_columns'** which takes four parameters as input, firstly the type of column we are trying to compare i.e. Product or Website, second and third parameters are the Client's Catalog dataframe and G2's Catalog dataframe respectively. And the last parameter is the threshold we are ready to accept in fuzzy match. This function returns a dataframe with three columns- Software Catalog Product/Website, G2 Product/Website and the fuzzy-match score.

5. Then we merge this with existing G2 and Client's Catalog data, so that we can get the associated Websites along with them. And we can now calculate the fuzzy score for these websites well.

6. At the end, select the highest fuzzy-match score value for each of client's software name, also subsetting the relevant columns as asked in the sample output file.

Results: I was able to match 176 Software Product names with the 384 distinct G2 Products.

Future Scope:

1. Given more time, these old websites can be replaced by scrapping the webpages of these websites and get the information from the json if these are acquired by other companies or not.

2. Also, we can use some sort of Similar words if not getting a proper match for the websites URL for example, The Facebook URL in Client catalog had career page and G2 had Workplace, so which is somewhat similar but not exactly.