INTERNATIONAL HELLENIC UNIVERSITY

Kothri Marilena AM 3308220013
Iakovidis Filippos AM 3308220008

Natural Language Processing and Text Mining

At the beginning of the code, we import the libraries we need. NumPy, Pandas, NLTK, re, random, and the Google Colab and after we mount the data from our google drive. The code is written in Google Colab.
Starting with the basics we define the paths for our csvs and we transform the data into our Pandas dataframes.

The dataframe consists of many unnecessary data, and we are only interested in brand names. So df_all is the dataframe that consists of “MFG Brand Name” from df_attr merged with df_train and df_pro_desc based on “product_uid”.

Then we begin with the text pre-processing
With str_stem we try clean our data.
1) re_sub
The re.sub() function belongs to the Regular Expressions (re) module in Python. It returns a string where all matching occurrences of the specified pattern are replaced by the replace string. (https://www.educative.io/answers/what-is-the-resub-function-in-python)
2) s.replace
The replace() method replaces a specified phrase with another specified phrase. (https://www.w3schools.com/python/ref_string_replace.asp)
With this function we replace special characters at first and then we proceed with the removal of blank space between parenthesis and brackets. The last replace aims to replace the length of the of the new string.

After this process we are ready to stem the rest of our data. So we apply str_stem to product_title", "search_term", "product_description", and "brand" columns in df_all.

Then we have a first look of our data.

simpson strong tie 12 gaug angl

angl bracket

not onli do angl make joint stronger they also provid more consistent straight corners. simpson strong tie offer a wide varieti of angl in variou size and thick to handl light duti job or project where a structur connect is needed. some can be bent skewed to match the project. for outdoor project or those where moistur is present

use our zmax zinc coat connectors which provid extra resist against corros look for a z at the end of the model number .versatil connector for variou 90 connect and home repair projectsstrong than angl nail or screw fasten alonehelp ensur joint are consist straight and strongdimensions 3in. x 3in. x 1 1/2in. made from 12 gaug steelgalvan for extra corros resistanceinstal with 10d common nail or 9 x 1 1/2in. strong drive sd screw

simpson strong tie

3.0

simpson strong tie 12 gaug angl

l bracket

not onli do angl make joint stronger they also provid more consistent straight corners. simpson strong tie offer a wide varieti of angl in variou size and thick to handl light duti job or project where a structur connect is needed. some can be bent skewed to match the project. for outdoor project or those where moistur is present use our zmax zinc coat connectors which provid extra resist against corros look for a z at the end of the model number .versatil connector for variou 90 connect and home repair projectsstrong than angl nail or screw fasten alonehelp ensur joint are consist straight and strongdimensions 3in. x 3in. x 1 1/2in. made from 12 gaug steelgalvan for extra corros resistanceinstal with 10d common nail or 9 x 1 1/2in. strong drive sd screw

simpson strong tie

2.5

Then we proceed with the creation of str_common_word and str_whole_word. The aim of those functions are to calculate common grams count and common entire term count.

Then we move on to create new columns to our first df.

We create a new column that combine "search_term", "product_title" and "product_description", another one that shows the number of times each term appears in product title., in product description, another column that counts the words that appear in search term that also appear in product title, and in product description, and different columns that calculate the ratio between columns.

Kothri Marilena AM 3308220013
Iakovidis Filippos AM 3308220008

After the creation of the columns we drop the columns that we don't need. So the df that remains has these columns.

Index(['relevance', 'word_len_of_search_term', 'word_len_of_title', 'word_len_of_description', 'word_len_of_brand', 'query_in_title', 'query_in_description', 'word_in_title', 'word_in_description', 'query_title_len_prop', 'query_desc_len_prop', 'ratio_title', 'ratio_description', 'word_in_brand', 'ratio_brand'], dtype='object')

Finally we move on to training and testing.

We use random forest regressor, ridge regressor, and XGBoost Regressor on the training data and evaluate their performance using MSE and RMSE as requested.

```
RandomForest RMSE: 0.4790
```

```
Ridge RMSE: 0.4852
```

```
Xgboost RMSE: 0.4773
```