Capstone Project #1 – Milestone Report

Link: Jupyter Notebook

The Problem

The project will attempt to build a model that can accurately predict the relevance of search results for an online retailer in order to create an improved customer shopping experience.

The Client

The client in this case is Home Depot, although the methods used should be applicable to other online retailers. Home Depot currently uses human raters to evaluate changes to their search algorithms, which is a slow process that can have bias. It is hoped that the results of this project can minimize or remove the amount of human input in the process and increase the number of iterations that Home Depot can perform on their search algorithms leading to a faster and more accurate customer experience.

The Data

The data set contains a number of products and real customer search terms from Home Depot's website. The challenge is to predict a relevance score for the provided combinations of search terms and products. To create the ground truth labels, Home Depot have crowdsourced the search/product pairs to multiple human raters.

The relevance is a number between 1 (not relevant) to 3 (highly relevant). Each pair was evaluated by at least three human raters. The provided relevance scores are the average value of the ratings.

There are four files given for the problem: attributes.csv, descriptions.csv. train.csv and test.csv.

attributes.csv

The attributes file consists of a product id, the name of an attribute and the values for that attribute.

There were 155 missing values for the id and name columns and 2090 missing values for the value column. The 155 missing values were entire

rows that were null, so these were deleted. The remaining missing values for the value column were filled with "None" since the lack of information for these attributes may prove to be valuable.

The product id column consists of a numerical id, so it was read into the data frame as a float. Since these are product id values, they are more appropriately classified as a string. After confirming that all the entries in the column consisted of six digits followed by a decimal point and a zero, the values were converted to a string and had the decimal point and zero trimmed.

The dataframe consists of text data, so there were no outliers.

descriptions.csv

The descriptions file consists of a product id and a text description of the product.

There were no missing values in the data.

Similar to the attributes file, the product id was read to the dataframe as an integer. After confirming that all values had the same format, the column was converted to strings.

The dataframe consists of text data, so there were no outliers.

train.csv & test.csv

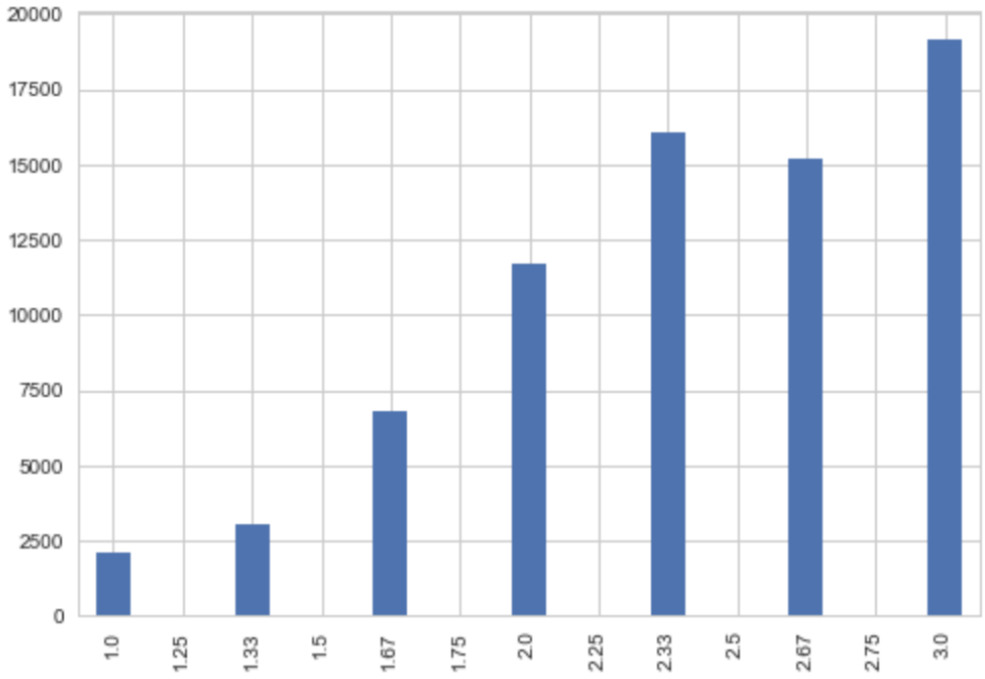
The train and test files consist of an id, a product id, the name of the product, a search term. The train file also includes the average relevance score assigned by the raters.

There were no missing values in the data.

Similar to the previous files, the product id was read to the dataframe as an integer. After confirming that all values had the same format, the column was converted to strings.

Raters were to assign a relevance score as an integer between 1 and 3, so the average relevance should be a float between those two numbers. The values were plotted and examined and confirmed that they were in the proper range.

The distribution of relevance scores is summarized in the following graph:



There were no outliers in the data.

Merging data

Since the train and test files contain a column for product id and product name, there seemed to be unnecessary duplication in these file. The product name column was extracted from these files, concatenated together and then merged with the descriptions file where it seemed more appropriate. These should remove duplication of data and save memory.

The attributes data was also merged into the training data to create one dataframe with all product information.

Processing text variables

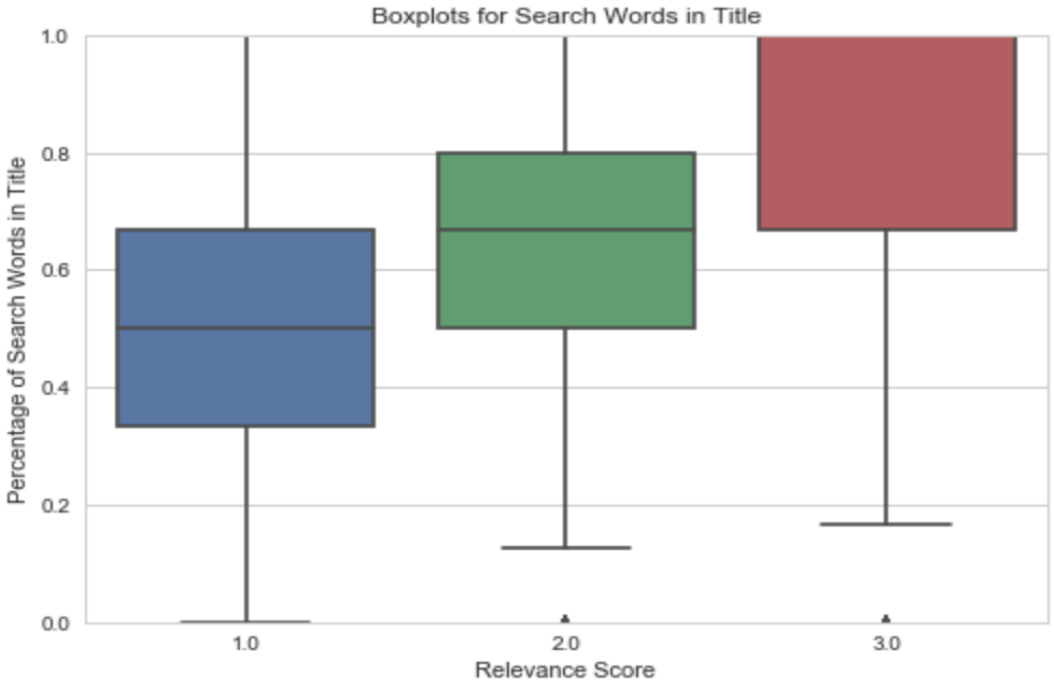
There are four primary text fields in the dataset: the Search Term and what will be called the long text fields, Product Title, Product Description and Product Attributes. Stop words were removed from these fields and stemming was performed to extract the roots of each word. Numerical variables were then extracted from these text fields in order to perform statistical analysis. These numerical fields considered such things as the

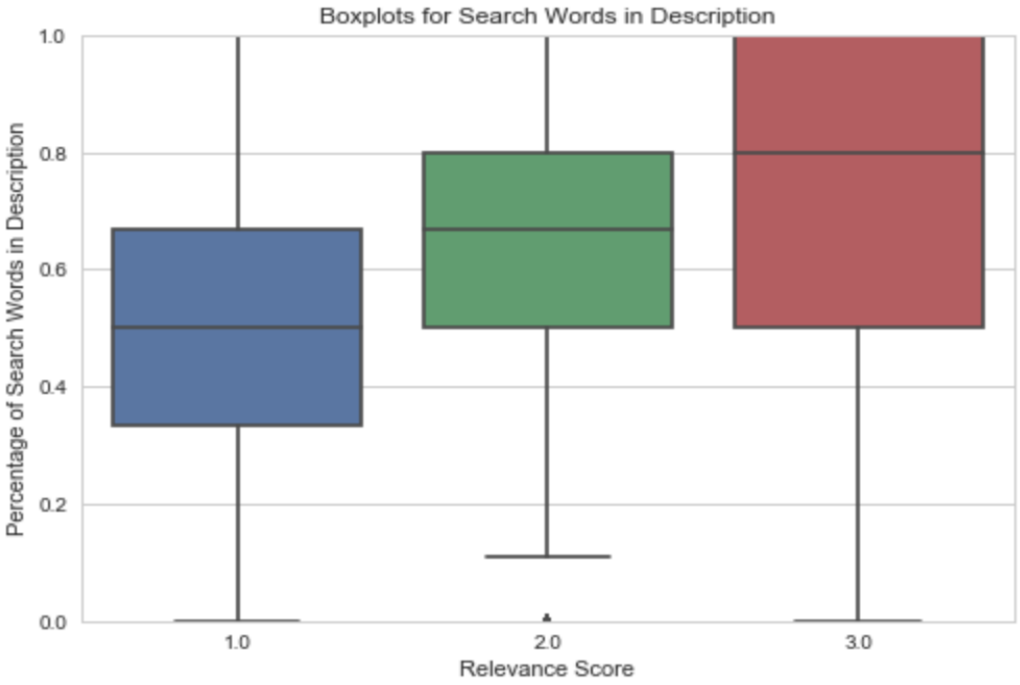
percentage of search words that are in the long text fields, the percentage of long text fields that contain a search term, the total, average and maximum number of search words that appear in a long text field.

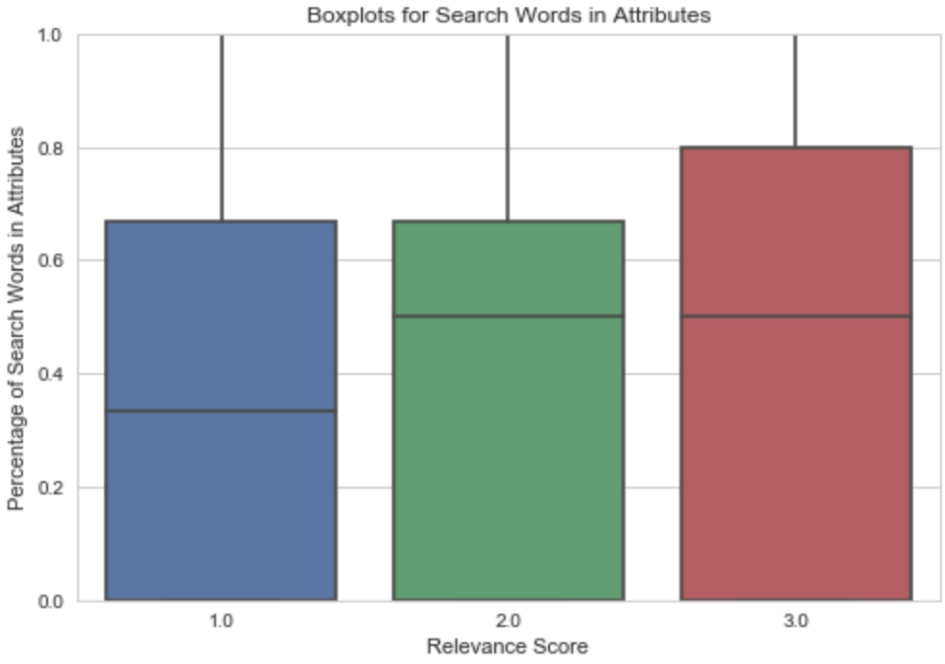
Other numerical variables that were extracted from our text fields were Boolean variables that captured whether or not the entire search phrase was found in the long text fields.

Visualization & Inference

For these numerical variables, boxplots were created to examine the differences between different relevance score levels. The boxplots for the percentage of search words that appear in the Title, Description and Attributes were as follows:







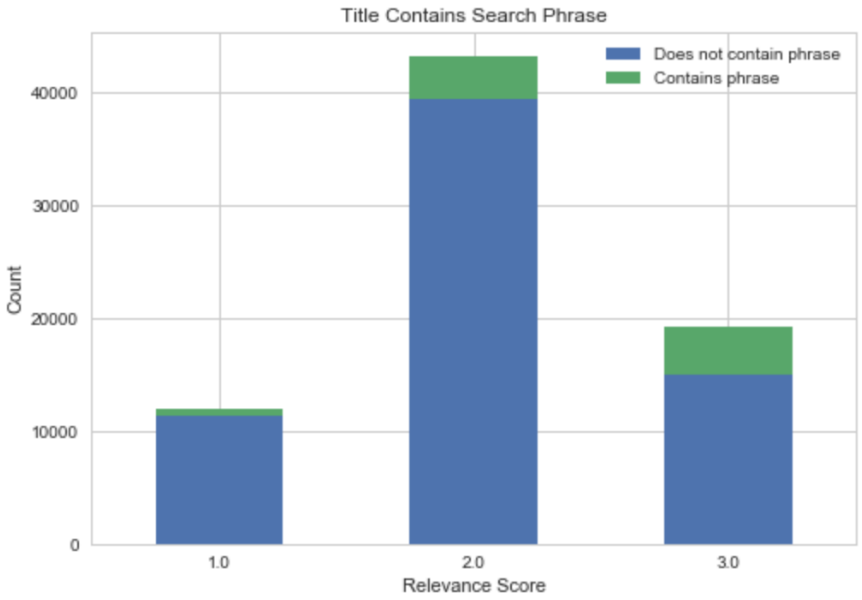
Many of the variables were similar to the one above where there is a positive correlation evident with Title and Description and either a weak or no correlation with Attributes. Other variables did not display any trend across any of the three text fields.

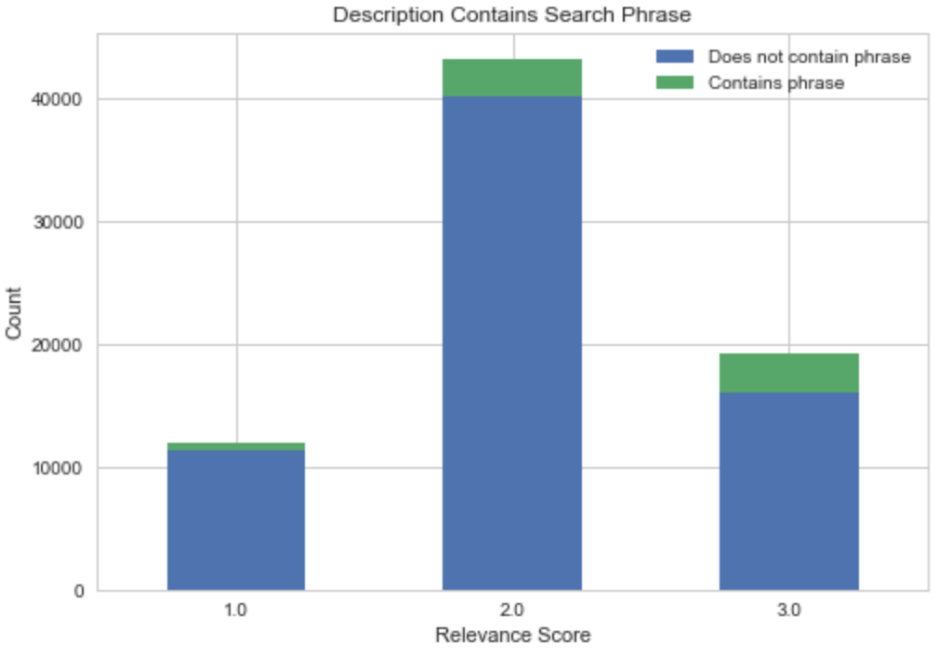
To test these findings, the nonparametric Kruskal-Wallis test was performed on the data, as histograms did not show evidence of normality. The Kruskal-Wallis test has the null hypothesis that the distributions of all populations tested (in our case three relevance levels) are the same. At a significance level of $\alpha = 0.05$, we were able to reject the null hypothesis and conclude that the distribution differed across relevance levels.

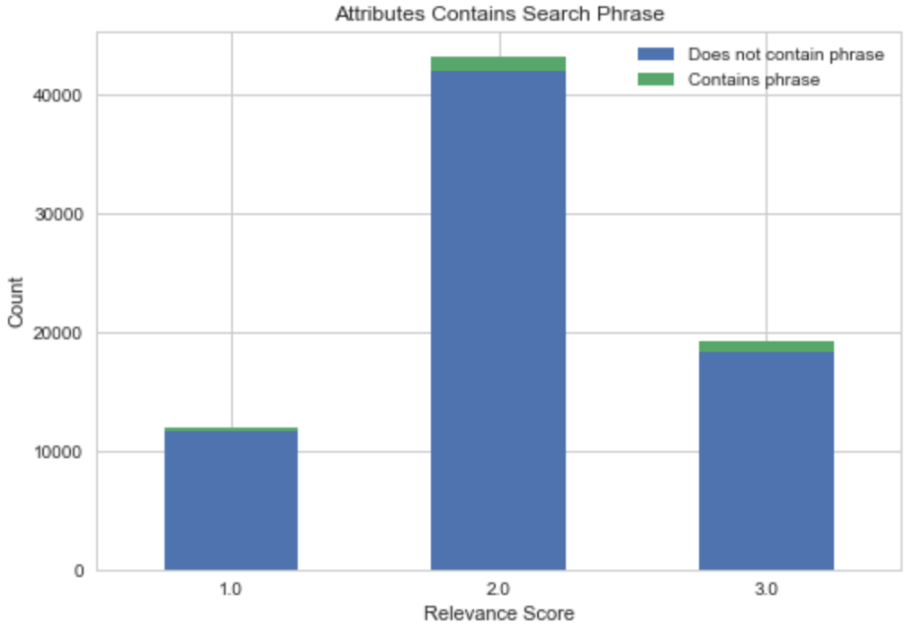
Based on some of the boxplots, this was not surprising, however many of the boxplots showed similarity in distributions. To examine further, the medians and means for all the variables were looked at. In the cases of many of the variables, the medians were similar across all relevance levels, which is consistent with the boxplots. However, in almost every case the means increased with relevance score confirming the results of the Kruskal-Wallis tests.

Examining the correlations between each variable and relevance score, it was seen that with only one exception, all variables had a mild to moderate positive correlation with relevance score. The one exception had a very weak negative correlation. Throughout every step of this process it was evident that Title had the strongest relationship with Relevance Score while Attributes had the weakest.

In order to examine the Boolean variables, stacked box plots were created for each of the three long text fields.




Description Contains Search Phrase
Does not contain phrase
Contains phrase
40000
30000
20000
10000
0
Count
1.0
2.0
3.0
Relevance Score


Attributes Contains Search Phrase
Does not contain phrase
Contains phrase
40000
30000
20000
10000
0
Count
1.0
2.0
3.0
Relevance Score

The stacked bar charts seem to show that the proportion of fields that contain the search phrase do increase with relevance score. These proportions were tested for equality using pairwise z-tests. Again, with a significance level of $\alpha = 0.05$, all null hypotheses that the proportions were equal were rejected. Once again, Title shows the highest degree of positive correlation, while Attributes shows the least.

All of the numerical variables extracted so far were tested for differences across the three relevance levels, and in almost all cases, evidence of a positive correlation was found. Across the study, the Product Title field has the greatest positive correlation with Product Description also trending positively. Product Attributes consistently had the weakest relationship with Relevance Score. This result may not be surprising as not all products have attribute data.

Machine Learning

With exploratory analysis complete, some final cleanup steps were necessary to prepare the training set to fit machine learning models.

Since only the numeric variables would be used in the models, all text variables were dropped from the dataframe. In addition, the id column was dropped, as it would not be needed in the analysis, along with the previously created relevance_reduced variable that had been created for visualization purposes.

With this complete, the dataframe was partitioned by separating the dependent variable (relevance) from the other numeric variables. The data set was then split into training and test sets (70/30) in order to fit the models. The overall test set provided from Kaggle is still set aside to produce a final score.

Since the Kaggle competition uses the root mean squared error (RSME) to score the given test set, various regression models will be tried and evaluated using that metric

The regression models that were attempted were Linear Regression, Lasso Regression, Ridge Regression, Random Forest Regression and SVM Regression. A similar process for each model: a pipeline was created that scaled the data and performed a cross-validation grid search to tune the parameters for each model. The models were then fit and an RSME score

calculated on each set of predictions. The results are summarized in the graph below:

del with an RSME score of 0.48011.

In an attempt to improve the model a Bagging Regressor was fit to the model. The RSME score of this model was 0.48015, which was not an improvement.

Finally, the model was used to make predictions on the provided test set. This was then submitted to Kaggle and received a final RSME score of 0.48614. This is consistent with the results seen previously and a sign that the model was not overfitting and was translating well to new data.

Conclusion and Next Steps

After fitting multiple regression models, the Random Forest Regression model was found to do the best job of predicting relevance score based on a given search term. This model could be recommended to the client in order to reduce the amount of human work in the process.

In order to improve the level of accuracy, I would suggest that more advanced and thorough Natural Language Processing could be used in the data

processing stage to produce a more robust set of features to feed into the model.

Another suggestion would be to revisit the attributes data file as more careful munging of this data may extract more useful information. The attributes related values consistently led to the least visible relationship with relevance score and I believe more value could be uncovered.

Finally, one suggestion I would make to the client is that rating the relevance on a three-point scale might be overly restrictive. Collecting data on a five, or even a ten-point scale, may give the human raters more flexibility in differentiating between the subtleties between products and produce a model that better reflects what the user is looking for.