

INFSCI 2725: Data Analytics
Final Project: Introductory report



Home Depot
Product Search
Relevance

A competition by



0. Overview

0.1. Group Member

	Name	Pitt email
1	Mohammed Alharbi	maa271@pitt.edu
2	Yixuan Edison Wang	yiw72@pitt.edu

0.2. Group Name

Our group name in Kaggle is MAYW, which is the initials of our first and last names.

0.3. Introduction

Home Depot become an important store that home owners rely on to find solutions to their home improvement needs, ranging from installing a new ceiling fan to remodeling an entire kitchen. Customers expect the correct results to their queries when they use Home Depot's system. Speed and accuracy are essential.

This project is a competitive project published by Kaggle, which is a platform for data prediction competitions. In this competition, Home Depot encourages Kagglers to improve their customers' shopping experience by developing a model that can accurately predict the relevance of search results. They provide three datasets about products, their attributes, and some search terms and their relevance.

Technically, Kagglers have to used data that were stored in file train.csv to predict the search relevance in another file, test.csv. They have to fit some model such as Linear Regression Model to learn from train set, and apply or predict the test data set.

To tackle the problem, we firstly bind the necessary datasets. Working in one dataset containing all needed data is more efficient. Then, we try to clean the data to eliminate missing values and unnecessary records. Next we fit some models such as Linear Regression Model and Generalized Boosted Model. Next, we predict the relevance of search terms using that model. Before generating the submission file, it is essential to test the accuracy of each model by computing Root Mean Squared Error (RMSE), highest accurate will be applied.

0.4. Programing Language

Before processing data, it is crucial deeply to think which programming language is going to help us to tackle the problem. We primarily use R programming for couple important reasons. R is essentially build to tackle problem involving exhausted statistical operations. R also has a great environment to adopted statistical graphs. It eases the way by which we draw, edit, and extract the graphs with fully control. Therefore, it gives us a way to understand the features of some main variables in the datasets.

1. Data

In this project, data is only the subject we are going to deal with. To take advantage of these data and learn their patterns, it requires a clear understanding of data features. This part will introduce the data features.

1.1. Data files

There are six files that produced by Kaggle. The following is their descriptions:

- **train.csv** - the training set, contains products, searches, and relevance scores.
- **test.csv** - the test set, contains products and searches. You must predict the relevance for these pairs.
- **product_descriptions.csv** - contains a text description of each product. You may join this table to the training or test set via the product_uid.
- **attributes.csv** - provides extended information about a subset of the products (typically representing detailed technical specifications). Not every product will have attributes.
- **sample_submission.csv** - a file showing the correct submission format.
- **relevance_instructions.docx** - the instructions provided to human raters.

1.2. Data fields

The following is the descriptions of all fields that the files contain:

- **id** - a unique Id field which represents a (search_term, product_uid) pair.
- **product_uid** - an id for the products.
- **product_title** - the product title.
- **product_description** - the text description of the product (may contain HTML content).
- **search_term** - the search query.
- **relevance** - the average of the relevance ratings for a given id.

- **name** - an attribute name.
- **value** - the attribute's value.

2. Data Exploration

2.1. Exploring Train and Test data

There are 74067 rows in the train.csv file and 166693 in the test.csv file. Both have the same fields: id, product_uid, product_title, search_term, and relevance. However, the relevance is not included in the test.csv file.

2.2. Exploring search terms

- Number of search terms in train: 11795.
- Number of search terms in test: 22427.
- Number of terms in train not in test: 2174.
- Number of terms in test not in train: 12806.
- Number of common terms in test and train: 9621.

2.3. Exploring product data

- Number of unique product ids in train: 54667
- Number of unique product ids in test: 97460
- Number of common product ids: 27699
- Number of product id in train not in test: 26968
- Number of product id in test not in train: 69761
- Number of product id in product descriptions: 124428

Examining the column product_id in both train.csv and test.csv files, there are 54667 unique values in the train.csv file and 97460 in the test.csv file. There are only 27699 common product_uid values among these. The total number of unique product ids across both the train and test data is 124428, which equals the total number of product ids and rows in the product_descriptions.csv file. This is displayed below in a Venn Diagram.

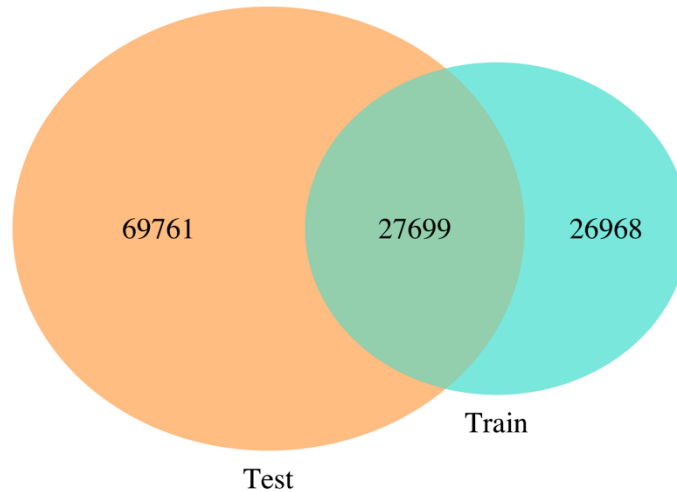


Figure 2-1: A Venn Diagram of product ids in both Test and Train

Examining the attributes.csv file, there are a total of 2044803 rows and a total of 86264 unique product ids. The intersection of product ids across the train, test, and attributes files is displayed in the Venn Diagram below. There is only one value in attributes.csv file that is neither in the Train nor Test files. On examining this, there are 155 rows in attributes.csv file that do not have a product_uid value. As it is unneeded, these rows can be **removed**.

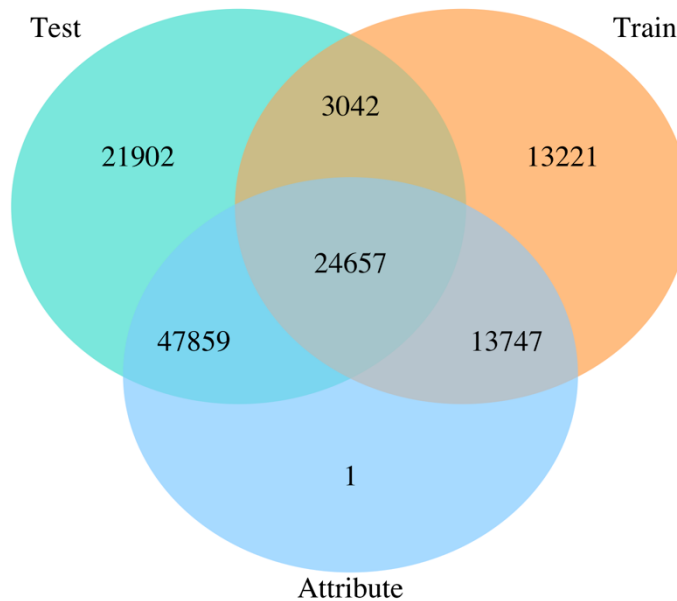
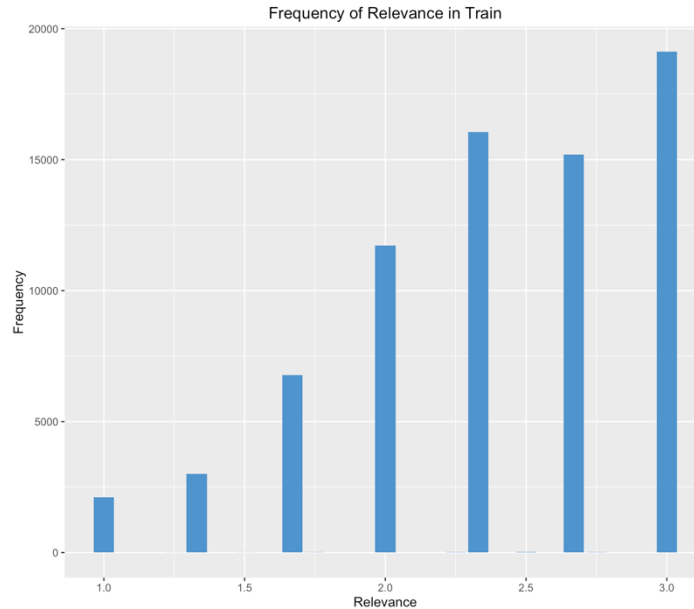


Figure 2-2: A Venn Diagram of product ids in the train, test, and attributes files

2.4. Train data

On investigating Train data, there are a total of 13 unique relevance values. A bar-chart of their frequency is displayed below. Note there are sets of relevance that occur with very low frequencies.



2.5. Attributes data

Examining Attributes data, there are 2044803 rows. Of these only 86264 are unique products. Moreover, there are 5344 unique categories (or name) of attributes. A sorted count of the top 10 for each category are displayed below.

Attribute's Name	Frequency
mfg brand name	86,250
bullet02	86,248
bullet03	86,226
bullet04	86,174
bullet01	85,940
product width (in.)	61,137
bullet05	60,529
product height (in.)	54,698
product depth (in.)	53,652
product weight (lb.)	45,175

Table 2-1: Count of top 10 Frequency of categories in attributes data

On investigating this data, 63 names refer to a color aspect. Similarly, 10 names refer to brand aspect. A sorted count of the top 10 for each are displayed below.

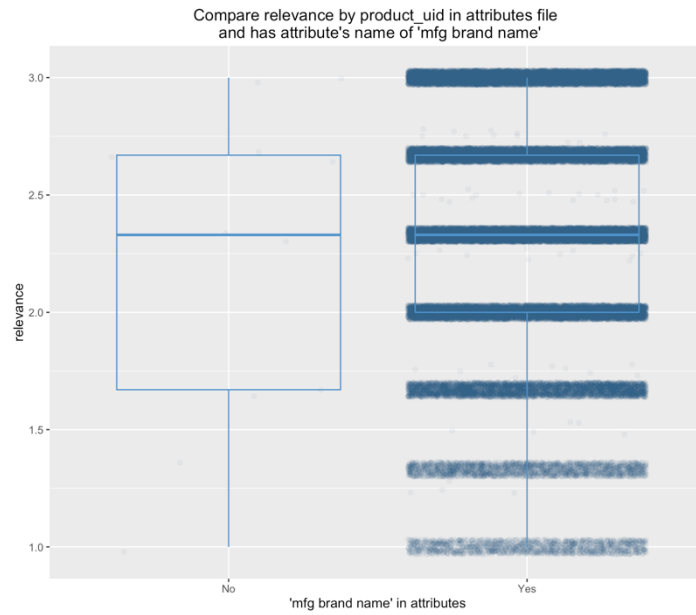
Attribute's Name	Frequency
color family	41,508
color/finish	28,564
color	6,222
color/finish family	4,630
fixture color/finish	4,119
fixture color/finish family	2,256
shade color family	2,006
actual color temperature (k)	1,421
color rendering index	1,118
top color family	996

Table 2-2: Count of top 10 frequency of color categories

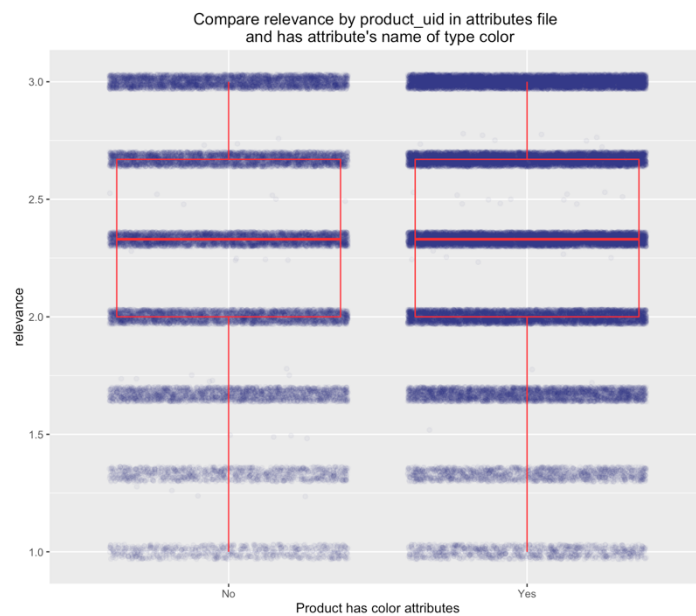
Attribute's Name	Frequency
mfg brand name	86,250
brand/model compatibility	681
brand compatibility	609
fits faucet brand	550
fits brands	407
fits brand/models	168
fits brands/models	119
pump brand	87
fits brand/model	6
brand/model/year compatibility	4

Table 2-3: Count of top 10 frequency of brand categories

The **mrg brand name** looks like the standard for brand attributes. In terms of color, the information is spread across multiple fields. A comparing relevance by product_uid in Attributes data and having attribute's name of 'mrg brand name' is displayed below.

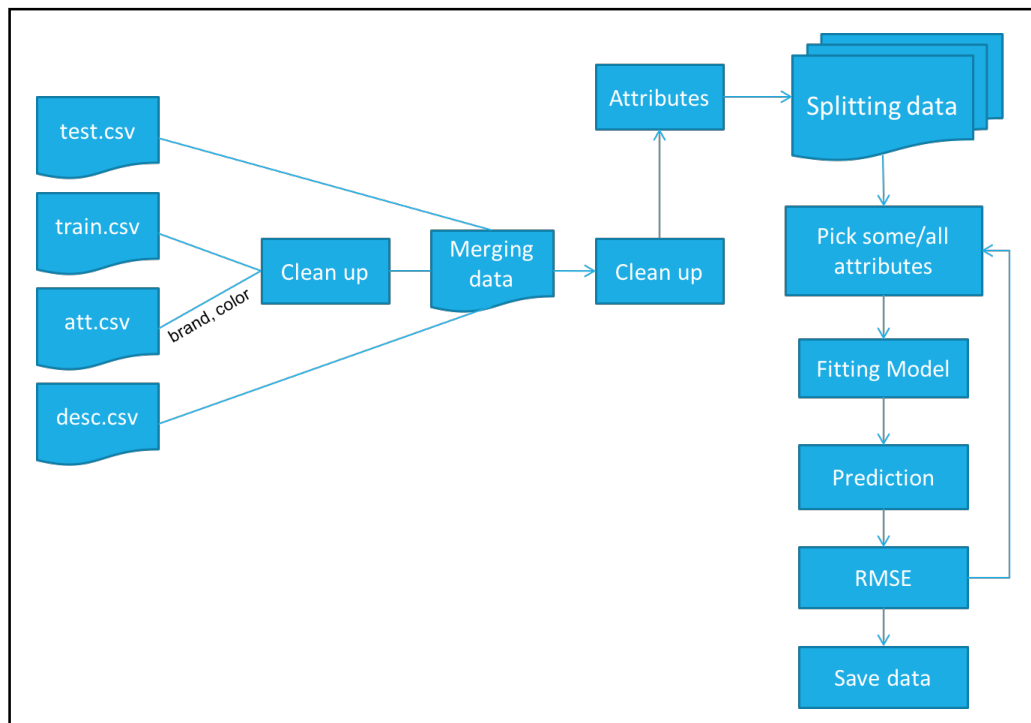


The following is a comparing relevance by product_uid in Attributes data and having attribute's name of type color.



3. Tackling the problem

Tackling the problem requires 11 steps including loading data, cleaning up the data (phase 1), merging data, cleaning up the data (phase 2), computing attributes, splitting data, picking attributes, fitting model, predicting train/test data, computing Root Mean Squared Error (RMSE), and saving submission data. We tried to apply those steps in optimal way in order to predict as much as possible from the test data. The graph below shows the sequence of the solution steps.



3.1. Loading data

We firstly read and load the datasets that we need in this project to our workspace in R programming. These datasets include:

1. Test data (test.csv).
2. Train data (train.csv).
3. Product descriptions data (product_descriptions.csv).
4. Attributes (attributes.csv).

3.2. Cleaning up the data (phase 1)

As there are many records in the datasets are missing some information of does not belong to training/testing data, those data do not affect the prediction process. Therefore, if we eliminate them, we will get a better performance to our code. In this step, some records have been found do not belong to train or test data, thus we get rid of them. In addition, we kept only important attributes such as brand and color; other attributes have been excluded.

3.3. Binding/merging data

We bind the datasets that we think will help us to fit models. It is a critical phase in our analysis. Working in one dataset including all needed data is more efficient. It gives us an opportunity to load data in memory with the minimum code lines. In this step, we merged `product_descriptions.csv` with both `train.csv` and `test.csv` files. Then, we bind `train.csv` and `test.csv` files in one file, called `all.csv`. Finally, we joined color and brand attributes form attributes datasets with the resulted data (`all.csv`).

3.4. Cleaning up the data (phase 2)

It's necessary to make training dataset more complete and uniformed in the beginning, which could make the later learning more accurate. We start, in this step, with working on correcting misspelling and making data uniform (normalization). For instance, toilet word may misspell as 'toliet'. Also, numbers may exist in search terms as letter, but it was inserted in `product_descriptions.csv` as a number. We have started to uniform the numbers in all fields and change them to numbers. In addition, in cleaning data aspect, we include the 'SnowballC' package to our work as it helps us to gain the word stem. We think that swapping words by its stem will increase the accuracy. We did a great effort in this step since our data is text those kind of data have much chance to have mistakes and correcting it is very expensive.

3.5. Computing attributes

Text data is hard to be used in prediction. Converting them to fact number is much easier and very helpful in the prediction. Eighteen new attributes (explanatory variables) were computed as they were found very beneficial. Then, these variables are appended to the major dataset, all.csv. The explanatory variables are computed as the following:

1. len_of_query: length of search term.
2. len_of_title: length of product's title.
3. len_of_description: number of words in product description.
4. len_of_brand: number of words in brand.
5. len_of_color: number of words in color.
6. query_in_title: if the whole search term is in the title, this attribute will be 1.
7. query_in_description: if the whole search term is in the product description, this attribute will be 1.
8. query_last_word_in_title: number of occurrence of last word (of search term) in title.
9. query_last_word_in_description: number of occurrence of last word (of search term) in product description.
10. word_in_title: number of occurrence of each word of search term in title.
11. word_in_description: number of occurrence of each word of search term in product description.
12. word_in_colors: number of colors that present search term.
13. ratio_title: $\text{word_in_title} / \text{len_of_query}$
14. ratio_description: $\text{word_in_description} / \text{len_of_query}$
15. word_in_brand: number of brands that present search term.
16. ratio_brand: $\text{word_in_brand} / \text{len_of_brand}$
17. brand_feature: ranking the brand, each brand has a specific number.
18. search_term_feature: number of characters in search terms.

3.6. Splitting data

The basic idea here in this step is splitting the training data from testing data, preparing to fit the model. It results in two different datasets, train and test datasets. Training datasets contains the original relevance values, but test data does not.

3.7. Picking attributes

Preparing to fit model we need to pick the important attributes (explanatory variables) that we are going to inject in the desired model. An easy way is pick all computed attributes, but this is not suitable all way. We will see in the result part how this affect the result.

3.8. Fitting model

There are various models that may used a machine learning technique for regression and classification problems. We have applied three of them in our project. The first model it has been applied was **Linear Regression**, which is used famously as a type of regression analysis and it has been covered early in our class. The second model is **Generalized Boosted Model (GBM)**, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. Indeed, we get the best result using this model. Finally, we applied **Random Forests**, which is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. It was able to fit the training data in this model, but unfortunately, it did predict the test data as it was expected; it seems like there are over-fitting.

3.9. Data Prediction

In this step, we predicted a relevance not only for each id in the test set, but also for each id in the train set for testing purpose.

3.10. Computing RMSE

Submissions are evaluated on the root mean squared error (RMSE), which is the square root of the mean/average of the square of all of the error.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

If the result of RMSE does not satisfy, we can perform the steps from 7 to 10 again.

3.11. Saving submission data

After predict a relevance for each id in the test set. Then, save the result in CSV format. The file should contain a header [id, relevance].

4. Outcome

The result is arranged according to the models that were used. The table in the next page shows the results that are gained. The minimal RMSE, which was 0.47626, was gained by **Generalized Boosted Model (GBM)**. **Random Forests** was the best in fitting the training data, but not in the test set.

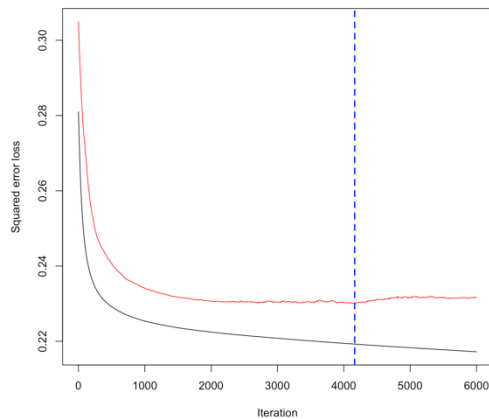
A table introduces our outcome

	Model	num.tree	product_uid	len_of_query	len_of_title	len_of_description	len_of_brand	len_of_color	query_in_title	query_in_description	query_last_word_in_title	query_last_word_in_description	word_in_title	word_in_description	ratio_title	ratio_description	word_in_brand	ratio_brand	brand_feature	search_term_feature	importance (permutation)	distribution	shrinkage	interaction.depth	n.minobsinnode	RMSE(train)	RMSE(test)
1	LR	NA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	NA	NA	NA	NA	0.4838606	0.48604
1	GBM	6000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	0.01	3	10	0.4720986	0.47862
2	GBM	6000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	0.05	3	10	0.4727213	
3	GBM	500	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	0.008	5	10	0.4701253	0.47818
4	GBM	3000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	0.006	7	8	0.468994	0.47798
5	GBM	3000	1	1	1	1		1	1	1	1	1	1	1	1	1		1	1	1	NA	gaussian	0.006	7	8	0.4703201	
6	GBM	3000	1	1	1	1		1	1	1	1	1	1		1	1		1	1	1	NA	gaussian	0.006	7	8	0.4702524	
7	GBM	3000	1	1	1	1		1			1				1	1			1	1	NA	gaussian	0.006	7	8	0.4705538	
8	GBM	3000	1	1	1	1		1			1				1	1			1	1	NA	gaussian	0.006	7	5	0.4702829	0.47843
9	GBM	6000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	0.01	7	10	0.4687573	0.47802
10	GBM	6000	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	gaussian	?	?	?	?	0.47626
1	RF	500	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	NA	NA	NA	0.5292541	
2	RF	2500	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	NA	NA	NA	0.2749385	0.48069
3	RF	2500							1	1				1	1		1				1	NA	NA	NA	NA	0.5069339	
4	RF	2500	1						1	1				1	1		1				1	NA	NA	NA	NA	0.4870805	
5	RF	4000	1	1	1	1		1			1				1	1			1	1	1	NA	NA	NA	NA	0.253911	0.48265

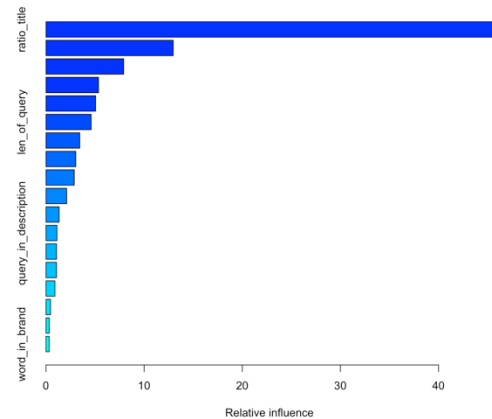
4.1. Generalized Boosted Model (GBM)

As it has been focused on this model by which we gained the optimal result, it is worth to mention and share some of its outcomes. For each trial, we are going to present a plot of GBM performance (left), a plot of the relative influence of each variable (right), and a table shows the relative influence of each variable at that trial (as it came from the model).

4.1.1. 1st Trial



A plot of GBM performance

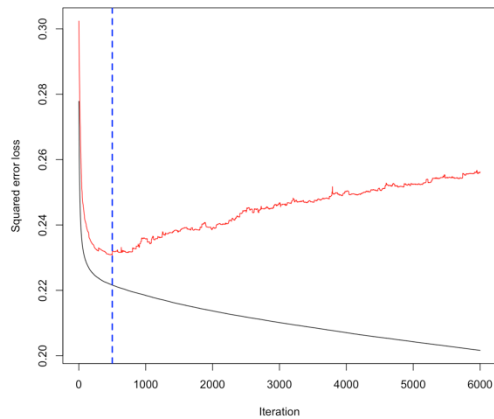


A plot of the relative influence of each variable

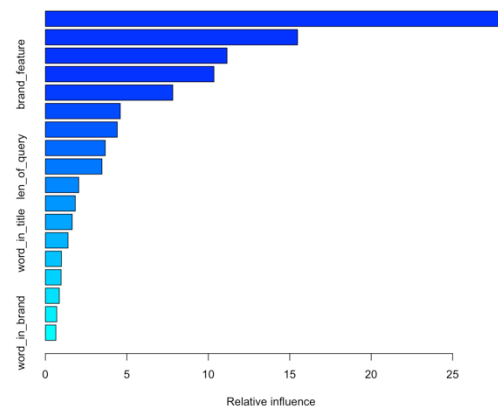
	var	rel.inf
ratio_title	ratio_title	46.0035038
search_term_feature	search_term_feature	12.9616674
product_uid	product_uid	7.9131976
ratio_description	ratio_description	5.3557942
brand_feature	brand_feature	5.0551997
len_of_query	len_of_query	4.6019530
query_last_word_in_title	query_last_word_in_title	3.4411718
len_of_title	len_of_title	3.0365670
len_of_description	len_of_description	2.8811351
len_of_color	len_of_color	2.1071329
ratio_brand	ratio_brand	1.3394996
query_in_description	query_in_description	1.1199486
query_last_word_in_description	query_last_word_in_description	1.0732244
word_in_title	word_in_title	1.0731572
query_in_title	query_in_title	0.9102424
word_in_description	word_in_description	0.4619976
len_of_brand	len_of_brand	0.3358273
word_in_brand	word_in_brand	0.3287805

A table shows the relative influence of each variable at that trial

4.1.2. 2nd Trial



A plot of GBM performance

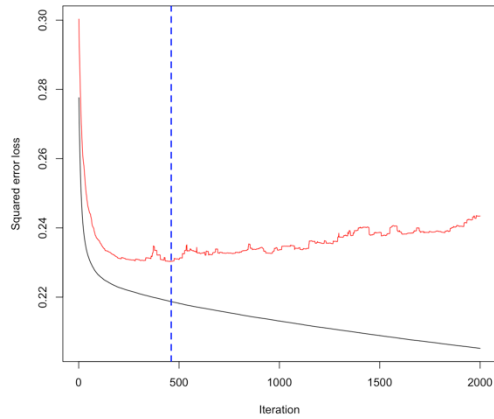


A plot of the relative influence of each variable

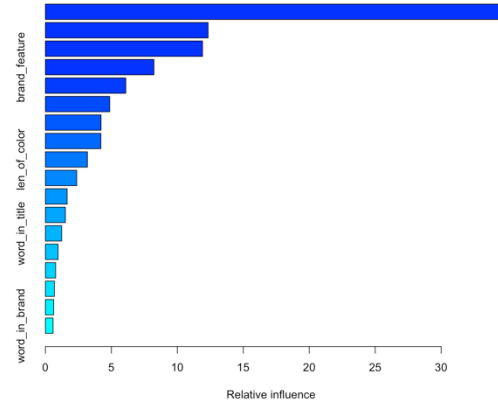
	var	rel.inf
ratio_title	ratio_title	27.9858273
product_uid	product_uid	15.4814140
search_term_feature	search_term_feature	11.1518609
brand_feature	brand_feature	10.3487984
len_of_description	len_of_description	7.8213802
ratio_description	ratio_description	4.5954036
len_of_title	len_of_title	4.4133017
len_of_color	len_of_color	3.6732268
len_of_query	len_of_query	3.4714031
query_last_word_in_title	query_last_word_in_title	2.0479372
query_in_description	query_in_description	1.8344418
ratio_brand	ratio_brand	1.6448350
word_in_title	word_in_title	1.3893930
query_last_word_in_description	query_last_word_in_description	0.9848640
word_in_description	word_in_description	0.9637272
query_in_title	query_in_title	0.8523451
len_of_brand	len_of_brand	0.6982565
word_in_brand	word_in_brand	0.6415842

A table shows the relative influence of each variable at that trial

4.1.3. 3rd Trial



A plot of GBM performance

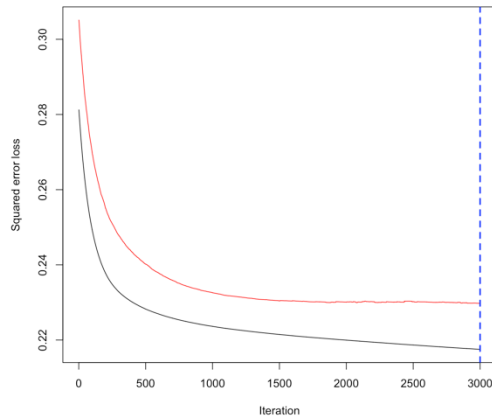


A plot of the relative influence of each variable

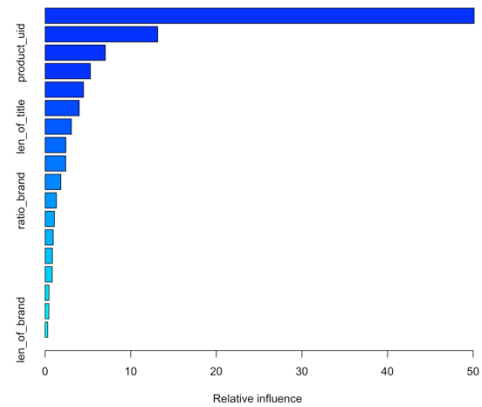
	var	rel.inf
ratio_title	ratio_title	34.5561323
product_uid	product_uid	12.3329043
search_term_feature	search_term_feature	11.9127532
brand_feature	brand_feature	8.2277649
len_of_description	len_of_description	6.0937960
ratio_description	ratio_description	4.8747679
len_of_query	len_of_query	4.2138045
len_of_title	len_of_title	4.2023677
len_of_color	len_of_color	3.1867105
query_last_word_in_title	query_last_word_in_title	2.3752428
ratio_brand	ratio_brand	1.6482304
query_in_description	query_in_description	1.5155415
word_in_title	word_in_title	1.2340180
query_last_word_in_description	query_last_word_in_description	0.9602818
query_in_title	query_in_title	0.7754319
word_in_description	word_in_description	0.6830407
len_of_brand	len_of_brand	0.6273728
word_in_brand	word_in_brand	0.5798387

A table shows the relative influence of each variable at that trial

4.1.4. 4th Trial



A plot of GBM performance

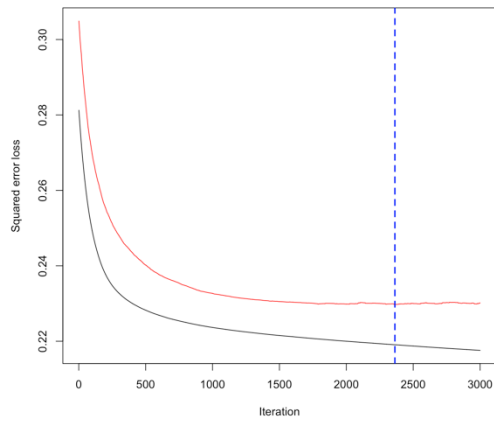


A plot of the relative influence of each variable

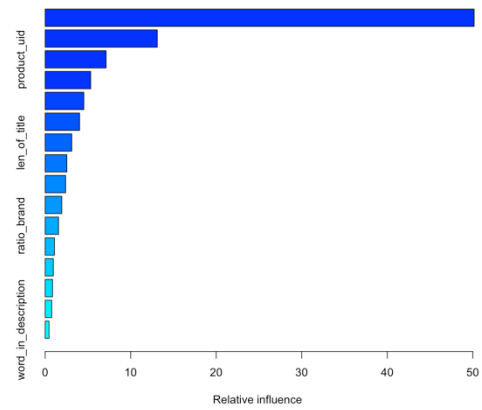
	var	rel.inf
ratio_title	ratio_title	50.1199976
search_term_feature	search_term_feature	13.1468200
product_uid	product_uid	7.0302800
ratio_description	ratio_description	5.2863581
len_of_query	len_of_query	4.4847730
brand_feature	brand_feature	3.9801691
len_of_title	len_of_title	3.0682878
query_last_word_in_title	query_last_word_in_title	2.4191648
len_of_description	len_of_description	2.4084447
len_of_color	len_of_color	1.8341298
ratio_brand	ratio_brand	1.3221049
query_last_word_in_description	query_last_word_in_description	1.0860626
query_in_description	query_in_description	0.9429245
query_in_title	query_in_title	0.8490493
word_in_title	word_in_title	0.8267396
word_in_brand	word_in_brand	0.4541414
word_in_description	word_in_description	0.4411732
len_of_brand	len_of_brand	0.2993794

A table shows the relative influence of each variable at that trial

4.1.5. 5th Trial



A plot of GBM performance

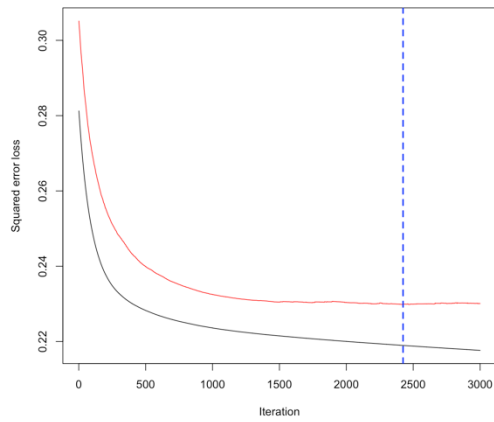


A plot of the relative influence of each variable

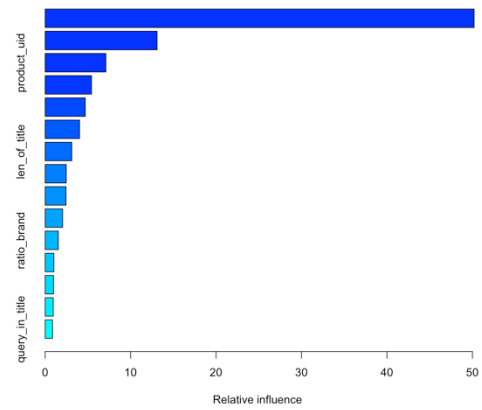
	var	rel.inf
ratio_title	ratio_title	50.1566632
search_term_feature	search_term_feature	13.1139861
product_uid	product_uid	7.1243150
ratio_description	ratio_description	5.3285620
len_of_query	len_of_query	4.5270282
brand_feature	brand_feature	4.0325609
len_of_title	len_of_title	3.1164095
len_of_description	len_of_description	2.5347362
query_last_word_in_title	query_last_word_in_title	2.3999387
len_of_color	len_of_color	1.9449161
ratio_brand	ratio_brand	1.5711999
query_last_word_in_description	query_last_word_in_description	1.0896323
query_in_description	query_in_description	0.9552364
query_in_title	query_in_title	0.8673857
word_in_title	word_in_title	0.7769229
word_in_description	word_in_description	0.4605071

A table shows the relative influence of each variable at that trial

4.1.6. 6th Trial



A plot of GBM performance

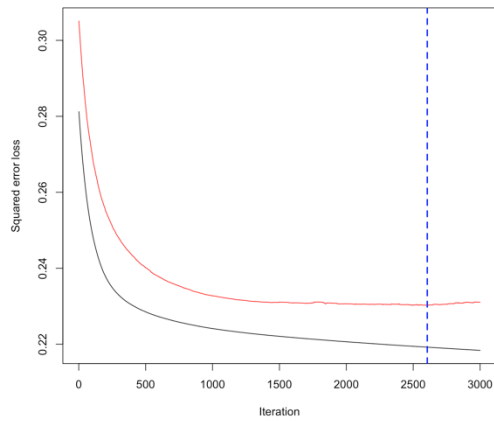


A plot of the relative influence of each variable

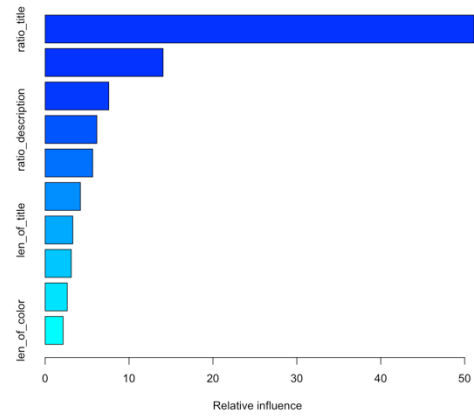
	var	rel.inf
ratio_title	ratio_title	50.2005423
search_term_feature	search_term_feature	13.1003365
product_uid	product_uid	7.1106142
ratio_description	ratio_description	5.4289312
len_of_query	len_of_query	4.6958973
brand_feature	brand_feature	4.0291789
len_of_title	len_of_title	3.1201628
len_of_description	len_of_description	2.4703155
query_last_word_in_title	query_last_word_in_title	2.4424120
len_of_color	len_of_color	2.0453154
ratio_brand	ratio_brand	1.5461957
query_last_word_in_description	query_last_word_in_description	1.0239730
word_in_title	word_in_title	0.9830085
query_in_description	query_in_description	0.9413998
query_in_title	query_in_title	0.8617168

A table shows the relative influence of each variable at that trial

4.1.7. 7th Trial



A plot of GBM performance

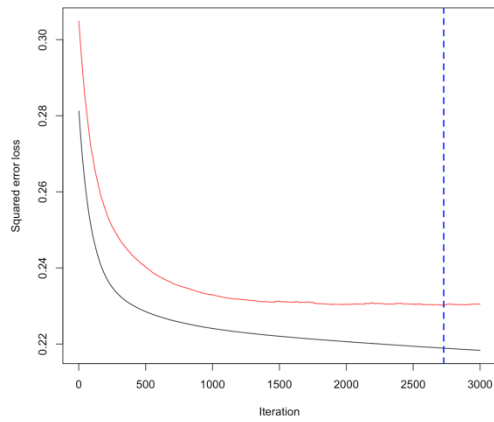


A plot of the relative influence of each variable

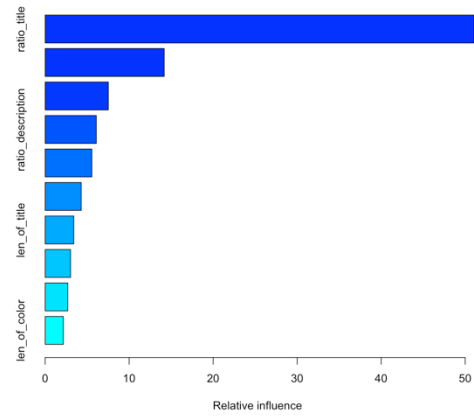
	var	rel.inf
ratio_title	ratio_title	51.130457
search_term_feature	search_term_feature	14.051950
product_uid	product_uid	7.581271
ratio_description	ratio_description	6.167532
len_of_query	len_of_query	5.671552
brand_feature	brand_feature	4.203273
len_of_title	len_of_title	3.293601
query_last_word_in_title	query_last_word_in_title	3.114714
len_of_description	len_of_description	2.639675
len_of_color	len_of_color	2.145975

A table shows the relative influence of each variable at that trial

4.1.8. 8th Trial



A plot of GBM performance

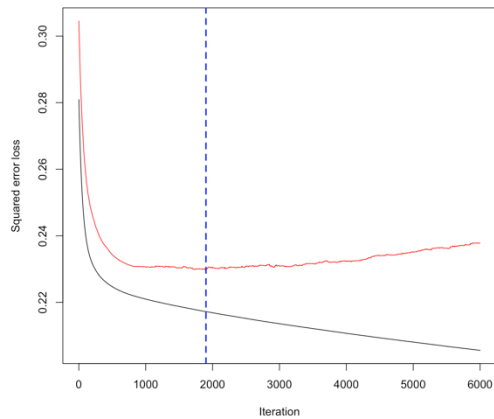


A plot of the relative influence of each variable

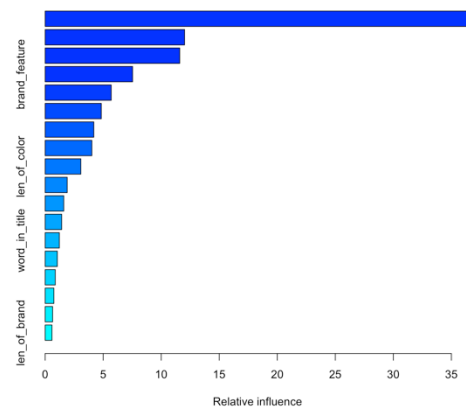
	var	rel.inf
ratio_title	ratio_title	51.066961
search_term_feature	search_term_feature	14.176141
product_uid	product_uid	7.521394
ratio_description	ratio_description	6.094807
len_of_query	len_of_query	5.560712
brand_feature	brand_feature	4.293040
len_of_title	len_of_title	3.407540
query_last_word_in_title	query_last_word_in_title	3.026593
len_of_description	len_of_description	2.693102
len_of_color	len_of_color	2.159710

A table shows the relative influence of each variable at that trial

4.1.9. 9th Trial



A plot of GBM performance



A plot of the relative influence of each variable

	var	rel.inf
ratio_title	ratio_title	36.9619141
search_term_feature	search_term_feature	12.0139895
product_uid	product_uid	11.5986418
brand_feature	brand_feature	7.5370440
len_of_description	len_of_description	5.6938480
ratio_description	ratio_description	4.8391818
len_of_title	len_of_title	4.1912256
len_of_query	len_of_query	4.0248717
len_of_color	len_of_color	3.0731631
query_last_word_in_title	query_last_word_in_title	1.9032792
ratio_brand	ratio_brand	1.6059463
query_in_description	query_in_description	1.4243951
word_in_title	word_in_title	1.2236025
query_last_word_in_description	query_last_word_in_description	1.0495696
query_in_title	query_in_title	0.8753742
word_in_description	word_in_description	0.7532471
word_in_brand	word_in_brand	0.6436765
len_of_brand	len_of_brand	0.5870302

A table shows the relative influence of each variable at that trial

5. Packages

Here is a reference of all packages that were used in our implementation:

5.1. SnowballC

This package has been used to extract the stems of each of the given words.

5.2. gbm

This package has been used to fit generalized boosted regression models.

5.3. readr

This package has been used to read tabular data.

5.4. ggplot2

This package has been used to visualize data.

5.5. dplyr

This package has been used to manipulate data (such as join, merge, ...etc.).

5.6. randomForest

It implements Breiman's random forest algorithm (based on Breiman and Cutler's original Fortran code) for classification and regression.

5.7. party

A Laboratory for Recursive Partytioning

5.8. tm

It has been for dealing with operations of text mining, specifically for building a corpus, a term-document matrix, ...etc.

5.9. hunspell

Hunspell Spell Checking and Morphological Analysis

5.10. ranger

Ranger is a fast implementation of Random Forest (Breiman 2001) or recursive partitioning, particularly suited for high dimensional data. Classification, regression, and survival forests are supported.

6. Submission

Submission to the competition on Kaggle is a part of this project. We have submitted 23 files in this competition. There was an error in three of them, so the total of the applicable submission was 20 submission files. The highest score was 0.54086, while the lowest on was 0.47626. Note as the evaluation was based on RMSE, **the low score resulted in higher ranking**. The first submission in this competition for us in Kaggle got 740 place. In the introductory report, we reported that it was achieved 0.50291 score at that time, resulting in ranking 1071. Now, we recover our early place in the competition; we come in 742nd place.

#	Submission	Public Score
1	Fri, 15 Apr 2016 04:20:09	0.48265
2	Thu, 14 Apr 2016 00:15:44	0.47802
3	Wed, 13 Apr 2016 08:29:50	0.47843
4	Wed, 13 Apr 2016 04:51:14	0.47798
5	Wed, 13 Apr 2016 04:18:28	0.47815
6	Wed, 13 Apr 2016 04:08:25	0.47818
7	Wed, 13 Apr 2016 03:43:59	0.47862
8	Tue, 12 Apr 2016 15:56:49	0.48604
9	Mon, 11 Apr 2016 10:05:36	0.48069
10	Mon, 11 Apr 2016 08:42:26	0.47862
11	Mon, 04 Apr 2016 19:48:58	0.47889
12	Mon, 04 Apr 2016 19:29:52	0.47845
13	Mon, 04 Apr 2016 19:21:32	0.47887
14	Mon, 04 Apr 2016 10:24:17	Error
15	Mon, 04 Apr 2016 10:03:51	Error
16	Mon, 04 Apr 2016 09:58:42	Error
17	Wed, 16 Mar 2016 06:21:24	0.47626
18	Wed, 16 Mar 2016 05:53:47	0.48378
19	Wed, 16 Mar 2016 05:47:42	0.47667
20	Wed, 16 Mar 2016 05:12:46	0.48378
21	Mon, 22 Feb 2016 06:58:50	0.53752
22	Sat, 20 Feb 2016 20:10:23	0.54086
23	Fri, 19 Feb 2016 23:22:31	0.50291

kaggle

Host

Competitions

Datasets

Scripts

Jobs

Community


Mohammed Alharbi

Logout

Mohammed Alharbi


Verified account

NOVICE



?

Joined 3 months ago



Profile


Results

Scripts

Forum

Account

Activity



Home Depot Product Search Relevance

20 entries in team MAYW

Current

742nd/2038

Ending 9 days from now

A screenshot shows our ranking in the competition

7. Conclusion and Thanks

To sum it up, Home Depot Product Search Relevance competitive project urges us to learn deeply and practically data analytics. As applying some technique being thought in this course, we were able continuously to improve not only our record in the competition but also our ability in data analytics. **Thank you for this opportunity.** Kaggle was a great place where we could exercise what we have been taught. We will continue participating in such those competitions to gain a great deal of experience in this field.

8. Reference

- [1] “Home Depot Product Search Relevance | Kaggle” <<https://www.kaggle.com/briantc/home-depot-product-search-relevance/homedepot-first-dataexploration-k/notebook>>
- [2] “Using a GBM for Classification in R on Vimeo” <<https://vimeo.com/71992876>>
- [3] “R-bloggers | R News And Tutorials Contributed By (573) R Bloggers” <<http://www.r-bloggers.com/>>
- [4] “inside-R | A Community Site for R | A Community Site for R – Sponsored by Revolution Analytics” <<http://www.inside-r.org/>>
- [5] “Setting up Hadoop 2.6 on Mac OS X Yosemite” <<http://sungsoo.github.io/2015/09/01/hadoop-installation-on-mac-os-x-yosemite.html>>
- [6] “Step-by-Step Guide to Setting Up an R-Hadoop System” <<http://www.rdatamining.com/big-data/r-hadoop-setup-guide>>
- [7] “How to work with Google n-gram data sets in R using MySQL” <<http://rpsychologist.com/how-to-work-with-google-ngram-data-sets-in-r-using-mysql>>
- [8] “Cryptanalysis with N-Grams” <<https://jeremykun.com/tag/ngrams/>>
- [9] “R Programming/Text Processing” <https://en.wikibooks.org/wiki/R_Programming/Text_Processing>
- [10] “Text Prediction Using R” <http://rstudio-pubs-static.s3.amazonaws.com/69859_25baf3e46b3646ad8be0f1657817dfef.html>
- [11] “How to Write a Spelling Corrector” <<http://norvig.com/spell-correct.html>>
- [12] “Text Data Mining with Twitter and R” <<https://heuristically.wordpress.com/2011/04/08/text-data-mining-twitter-r/>>
- [13] “Classifying text with bag-of-words: a tutorial” <<http://fastml.com/classifying-text-with-bag-of-words-a-tutorial/>>
- [14] “Naive Bayes and Text Classification” <http://sebastianraschka.com/Articles/2014_naive_bayes_1.html>