Home Depot Product Search Relevance

Srivarsini Rangarajan

11101, Gainsborough Ct,
Apt #11,
Fairfax, VA-22030
Tel. No: +1 (571) 363-6092
Email:
srivarsinii.rangarajan@gmailcom

Chaitanya Krishna
Sai Kosaraju
9411, Lee Highway,
Apt# 404,
Fairfax, VA-22031
Tel. No: +1 (703) 628-3226
Email:
krishnasai.kosaraju@gmail.com

Sai Kolanupaka
Meghana Bharadwaj
4291, Cotswolds Hill Lane,
Fairfax, VA 22030

Tel. No:+1 (703) 728-6038
Email:
meghanabharadwaj8@gmail.com

Abstract:

Customers of Home Depot are looking for a highly reliable and best solution for enhancing their home needs. This paper outlines the problem of the existing Home Depot Search and various machine learning algorithms that are used in the industry to cater similar needs. Also it briefly explains the limitations of the existing solutions, approach towards the problem and the results.

Problem Statement and Opportunity:

The objective of the project is to achieve maximum accuracy in product search result with minimum response time by predicting the relevance factor for search term and product pair. The threshold limit for the response time will be decided based on the search response time of various other shopping websites such as Amazon and Wal-Mart. Other opportunities associated with this project are predicting the customer purchasing pattern and placing the products of interest according to the relevance score to the customers. The project is basically identified to be a Classification text mining problem involving supervised learning.

Importance and Existing Solutions:

Lot of big organizations are spending billions of dollars in making their search better such as Google, Amazon, Expedia and many other firms emphasizing the increasing need of optimal search solutions.

Oracle uses the Text Scoring algorithm based on the Salton's formula. Inverse frequency scoring assumes that frequently occurring terms in a document set are noise terms, and so these terms are scored lower. For a document to score high, the query term must occur frequently in the document but infrequently in the document set as a whole. Google uses PageRank algorithm for ranking the websites. Other algorithms used are Hilltop and Topics Sensitive PageRank. PageRank is based on the probability distribution to represent the likelihood of person randomly clicking on links will arrive at any particular page. To get better results, this algorithm requires a lot of iterations.

Amazon uses A9 algorithm that is primarily based on indexing and previous search pattern recognition history.

Other algorithms that are widely used to solve the problem are

***Decision Tree Algorithm* -** Decision tree works with nodes represented by decisions and leaves represented by the class labels. It basically involves two steps Tree induction and Tree pruning.

***Neural Networks* -** This algorithm is best applied for non-linear relationships. It works well with the missing or and dynamic data.

***Naïve Bayes* -** Naïve Bayes is a conditional probability model that derives the evidences out of the training set and applies the evidence to predict the outcome in the test data set or new dataset. This model is widely used in clustering and supervised learning.

Support Vector Machines - SVM is a linear machine learning algorithm used for supervised learning, classification and regression analysis. The SVM models can be extended to non-linear problems by applying few modifications such as kernel tricks. This method is widely used in text mining especially in medicine industry to determine the compound accuracy.

Limitations:

The limitations of existing solutions are number of iterations to achieve better results and the time taken for each iteration. Home Depot is using a manual rating to evaluate the impact of potential changes to their search algorithms that is slow and subjective. So they are looking for better solutions to minimize the human input in search relevance evaluation.

Improvements in the existing solutions:

We are aiming to build a classification model to achieve higher accuracy in obtaining the related products using Naïve Bayes Algorithm with Feature selection and other statistical improvements.

Technical approach and Proposed Solution:

Our approach involves the classification and supervised learning.

- Pruning the data set – OpenRefine/Data Wrangler/Data Cleaner
- Create the frequency set for the terms – n-gram search – Python/Java
- Implement the search algorithm - Naïve Bayes
- Determine the relevancy factor
- Result Visualization - R

Data Used:

The data is taken from the kaggle competition dataset. This 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 crowd sourced the search/product pairs to multiple human raters.

Files

train.csv - the training set, contains products, searches, and relevance scores

test.csv - the test set, contains products and searches. Our aim is to 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.

Data Fields under consideration are

- **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

Metrics:

The metrics to measure the success of the problem is to measure the prediction accuracy. One way of visualizing the accuracy in classification is the Root Mean Square Error and Cross Validation Error.

RROC can also be used to determine the underestimated and over estimated values.

Relevancy Decision: The relevancy is determined by the rules.

Perfect Match if:

- The result matches the query intent perfectly.
- The product is exactly what the customer was most likely trying to find.
- The query was for a brand and the result is made by that brand.

Partially or somewhat relevant if:

- The result generally matches the query, but there is ambiguity in either search terms or product details that prevent you from choosing 'Perfect Match.'
- Actual product is the same but differs in brand, dimensions, color or specifications.
- Product specifications are not clear or product is somewhat ambiguous synonym of the search term.
- One of the many products searched in the query are shown.

Irrelevant: Paired with Search by Mistake if:

- The result is not at all related to the intent of the search term.
- The result is an item that is used with search term such as tool, accessory, extra part, cover or case.
- The result has a completely different meaning, even if there are words in common with the search term.
- The result is a collection of items and the search term does not appear in the set.

Initial analysis of data set showed the potential of using a frequency based solution to solve the problem. Upcoming sections explain various approaches followed to solve the problem.

Section 1: Term Frequency – Inverse Document Frequency

TF-IDF is used to determine the important terms in a document and eliminate the more noisy terms in the document.

Important considerations of the algorithm are

1. Term weight gets lower when the term is present less number of times or it is present in many documents or gets even less weighing when it is present in almost all the documents.
2. Term weight is highest if the term occurs many time in less number of document

Since the product description had a lot of data this algorithm seemed to be a good one to solve the problem. Almost all the variables in the data set are words. The frequency generation of these different variables will help to calculate the relevancy between the search term and the other determinants.

The approach to use TF-IDF in solving the problem is

- Stem the product description data using stop words.
- Generate a frequency table for product description table
- Create a dictionary containing search term for all the search ids.
- Iterate over the frequency table to calculate the term frequency and inverse document frequency

Drawbacks:

- The data set was huge to be processed.
- Frequency generation process followed is not good enough to process all the words.
- Difficult to deal with words having same meaning. For example angles, angle, angled.
- Theoretical implementation of algorithm on few products-search combination failed to predict the correct product for the search term

Example:

Consider a product with product id, 1 having 6 related search queries in train and 2 related search queries in test data set.

Assume the search query to be angle bracket and that refers to the product id - 1. Split the search query into unique words such as angle and bracket and for each term in search query using the below formula

- Find tf-idf in product title
- Find tf-idf in product description

tf = count(angle) in record 1/total number of words in record 1
Idf = log[count(total_records)/1+No. of records containing angle]

The relevance score is 0.9 which is very low compared to the given relevancy score of 3 for that particular product and search term.

Section 2: Feature Vector Generation

Another approach very similar to TF-IDF is to generate a frequency table for all the variables of interest but in a different way. Here we take the search query and find the words in individual column (variable) and add up their frequency for each record separately.

Merge all the variables into one single data frame. This will help to predict the relevancy score more accurately. This approach also makes it easy to apply different regression algorithm with same input and allows us to verify the performance of them.

Steps involved:

- Import all the data into a data frame
- Eliminate all the missing values
- Attach the product description column to it based on product id
- Apply stemming on all the columns containing string values – search term, product title, product description, product attribute
- Find the length of the query
- Find term in title, description and attribute and count the frequency
- Merge the frequency columns to the existing data frame

Now create a data frame containing only the numeric labels – search-id, length of query, product id, term description, term attributes, relevancy.

Search ID	Product ID	Relevancy	Query Length	Title Count	Desc. Count	Attribute Count
2	100001	3	2	1	1	1
3	100001	2.5	2	1	1	1
9	100002	3	2	1	1	2
8168	101384	?	3	0	1	1
8169	101385	?	4	1	1	1
8170	101386	?	2	1	2	2

Fig. 1.1 Sample Data Frame of Numeric Vectors:

The table 1.1 represents the numeric vectors

- Query Length – The number of words in the search query
- Title Count – Number of words that matches the query search term in the title
- Description Count – Number of words that match the search word in description.
- Attribute Count – Number of words that match the search word in description
- Relevancy – Measure of similarity between the search term and product id. Test data relevancy is represented by '?' that needs to be determined.

These numeric vectors now represent the features for the machine learning algorithms.

Section 3: Support Vector Regression

Support Vector Regression is a supervised learning model that helps to analyse the data and perform regression upon the feature vectors.

In SVR, the input is mapped onto a multidimensional feature space (query length, title count, description count and attribute count – Table 1.1) using non-linear mapping and then a linear model is build in this feature space. In mathematical notation the linear model is given by

$$f(x,w) = \sum_{i=1}^{m} w_j g_j(x) + b$$

$g_j(x)$, j= 1 to m is the set of non linear transformations on the multidimensional space

b = bias term

The performance of the SVR depends on the selection of variables for the multidimensional space. Most of the implemented algorithms leave this selection to the user. The selection should be well defined and the variables should be considered wisely according the domain under consideration.

From the given dataset and numeric vector(Table 1.1), we derived three important data frames

1. Train Data relevancy - TR
2. Train Data features - TrF
3. Test Data Features – TeF

The model is trained for predicting the relevancy score for the training data based on the training data features.

Fit the model for the prediction values that are generated during training.

Apply the model to the test data to determine the relevancy score.

The cross validation value for this model is about 0.67 which shows over-fitting of data to some extent.

The root mean square(RMS) value that calculate the difference between the observed and the prediction value is about 0.60. The higher value indicates greater difference.

The model execution time is about 424 seconds.

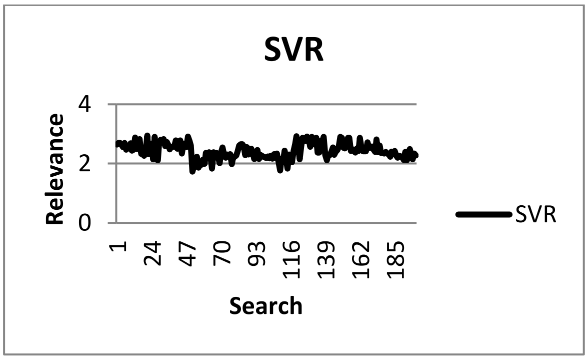

Fig. 1.2 Search term vs Relevancy Score

The results show that SVR is not performing well in prediction for the data under consideration.

Section 4: Decision Tree:

Decision Tree Regression is another effective method of predicting unknown values. It uses the concept of fitting a sine curve with addition of noisy observation. So, it approximates the sine curve learning the local linear regressions.

Another important factor determining the performance of decision tree is the depth of the tree. The predictions get finer with increasing depth. At the same time the model execution time increases.

Threshold of selecting the appropriate depth is crucial.

In our model we use the max depth of 10 as the data is huge and other computational limitations. Using the same data frames that are generate for SVR, we got the below results.

Cross validation value is about 0.55 showing less over-fitting compared to SVR. The root mean square value is 0.77 showing high difference between the predicted an observed value. This might be due to the noisy data as mentioned earlier.

The execution time is 277 seconds and is low compared to SVR.

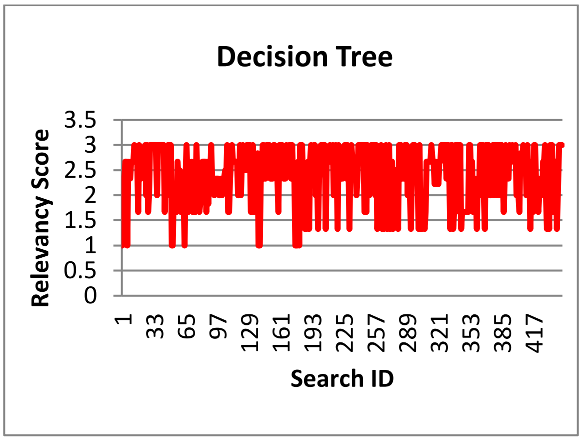

Fig.1.3 Search Term vs Relevancy Score – Decision

Section 5: Random Forest:

Random Forest is an ensemble technique involving randomizing the samples into subsets and apply decision tree upon the random subset samples. It involves building an ensemble classifier that represents a forest.

The data set in our consideration has independent features so random forest should work effectively as the error rate in random forest is proportional to the correlation between the features.

The issue with the random forest is that it over-fits the data when it moves down the tree resulting in more prediction error. So it is important to determine an optimal value of m.

One way of avoiding the over-fitting is to use a good sampling method. Bagging is one of the widely used sampling techniques to avoid the over-fitting of data. It makes use of averaging technique over a large number of bootstrap samples. Bagging ensures a wide variety of classifiers to apply regression.

The random forest with bagging model upon the numeric vectors provided a cross validation value of 0.62 and root mean square value of 0.58. The results are good compared the previously discussed models in Section 4 and Section 5. Also the results prove that the random forest with bagging highly improves the predictions compared to decision trees. The model execution time is 296 seconds which is not very high than decision trees.

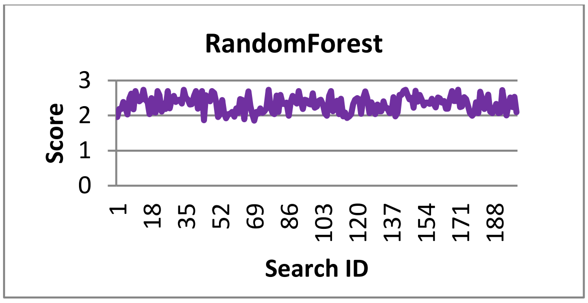

Fig. 1.4 Search term vs Relevancy Score

Section 6: Multiple Linear Regression

Multiple Linear Regression works best for the regression analysis. This model is simple compared to the other discussed models.Given n observations, the model fits the variables $x_0,x_1,....x_n$ into $k_0,k_1,......k_n$ with the deviation recorded as residuals given by $r_0,r_1,......,r_n$.

The multiple linear regression model upon the numeric vectors for home depot product search resulted in a cross validation value of 0.68 and RMSE value of 0.57. The model execution time is 270.31 seconds.

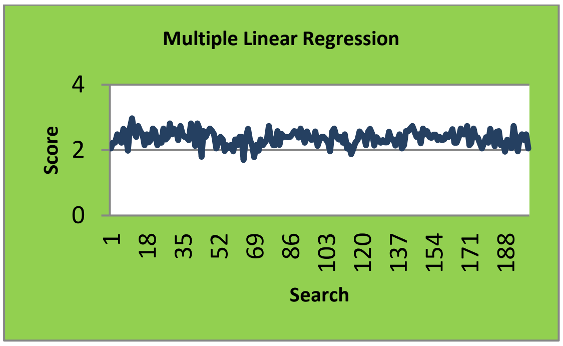

Fig. 1.5 Search vs Relevancy

Section 6: Comparison of Models:

Based on the previous discussion, Multiple Linear Regression performed well on the dataset under consideration and given feature vectors. Other models that performed well are Random Forest with Bagging followed by Support Vector Regression.

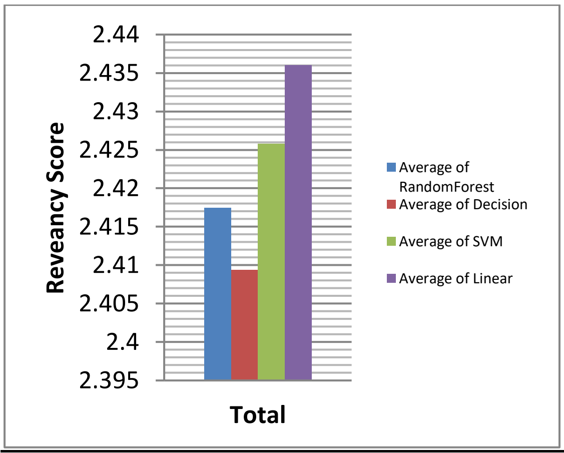

Fig 1.6 Average of prediction across the algorithms

Validation of results using other models:

Neural Networks:

Neural networks basically works upon the models having lot of input variables. Even though we have only few variables, the huge amount data gave us a hint that this model may give us a better results.

Upon executing a random sample in XL Miner platform provided us a root mean square error of about 0.488 which is the lowest of all the outputs.

But this doesn't tell us that this model performed well because the amount of data used to find the result is really low.Because of system limitation we could not execute the algorithm for the subset we considered for other algorithms.

The results are significant as this may provide us better results if we run the model for the entire data set

Multiple Linear Regression – XM Miner:

Executing a sample of data in XL Miner platform provided us a low RMS value. But as for neural networks the sample set is really low compared to the sample set for other algorithms.

The results provided other significant factors such as the p value for each input variable that helped us to derive the stronger predictors such as query length, title count followed by attribute count.

RROC – Regression Receiver Operating Curve

RROC represents the predictive power of a model by plotting the overestimates and underestimates with a fitting line.

Both Neural Networks and Linear Regression had underestimated the relevancy score.

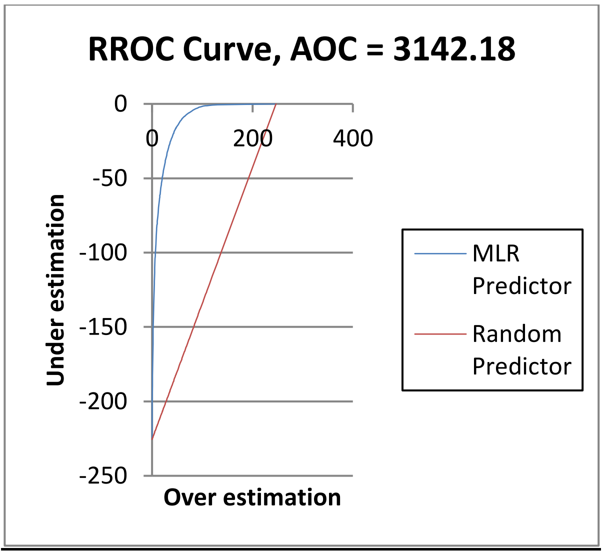

Fig 1.7 RROC Multiple Linear Regression

Scatter Plot Matrix:

Scatter plot matrix is a very good graphical representation of viewing the relationship between the input and output variables.

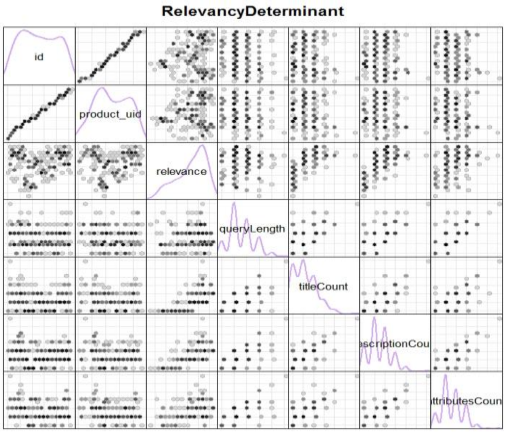

Fig: 1.8 Scatter Plot Matrix for Relevancy Determination

Scatter plot matrix shown in fig. 1.8 depicts that the relevancy score increases with increase in query length to some extent and it drops down if the query length is really high. This is a scenario where you have lot of words in the search query but not all of them are related to the product. So the relevancy drops to a lower value.

The denser plot represents more predictions on that region of scale. Since the model is based on the count of words, the prediction increases with optimal count in the count of words across all the variables. Extremely high and low values of input variables drops the relevancy score.

Conclusion:

According to the analysis described in this paper, Multiple Linear Regression gives us better prediction compared to other algorithms. Consideration of the entire data set may change the result. Classification of products based on brands and functionality will definitely improve the prediction accuracy. Ensembling of various algorithms in each step will provide even higher prediction accuracy.

References:

- Mitchell, T (1997). Machine Learning, McGraw Hill. Ng, A.Y. & Jordan, M. I. (2002). On Discriminative vs. Generative Classifiers: A comparison of Logistic Regression and Naive Bayes, Neural Information Processing Systems, Ng,
  A.Y., and Jordan, M. (2002).
  Wasserman, L. (2004). All of Statistics, Springer-Verlag.
- SURVEY PAPER - Top 10 algorithms in data mining XindongWu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang ·
  Hiroshi Motoda · Geoffrey J. McLachlan ·
  Angus Ng · Bing Liu · Philip S. Yu ·
  Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg
  Received: 9 July 2007 / Revised: 28 September 2007 / Accepted: 8 October 2007
  Published online: 4 December 2007
  © Springer-Verlag London Limited 2007
- Importance of Online Product Reviews from a Consumer's Perspective
  Georg Lackermair, Daniel Kailer, Kenan Kanmaz Munich University of Applied Sciences, Germany
  2TU Dresden, Germany_Corresponding Author: dkailer@hm.edu
- A tutorial review on Text Mining Algorithms
  Mrs. Sayantani Ghosh, Mr. Sudipta Roy, and Prof. Samir K. Bandyopadhyay
  Department of Computer Science and Engineering1,2,3
  University of Calcutta, 92 A.P.C. Road,
  Kolkata-700009, India.

- https://www.kaggle.com/competitions
- https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- http://stackoverflow.com/
- https://www.quora.com/
- http://kernelsvm.tripod.com/
- http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm