Project-012: Home Depot Search Relevance Data Analysis

Saif Ahmed
Indiana University Bloomington
F16-IG-3000
ahmedss@umail.iu.edu

Snehil Vishwakarma
Indiana University
Bloomington
F16-IG-3022
snehvish@umail.iu.edu

Sruthi Mallina
Indiana University
Bloomington
F16-IG-3014
smallina@umail.iu.edu

ABSTRACT

Search query relevancy and product recommendation systems are at the very core of any web based application and serve as a tool for the user to obtain meaningful results for their query. If a user has to search for a product, then the application has to return the best related products available in their catalog and in doing so even recommend various products that might be associated with it. All of this has to occur without displaying underwhelming results with little or no association at all with the original search query. For the analysis of this concept we have chosen Home Depot Search Relevancy dataset from Kaggle.
The underlying objective is to predict the relevancies of the search query with various attributes of the search term provided by the user. We have preprocessed the dataset using various techniques such as stemming of data, finding the common words between different attributes, cleaning the data with the use of regular expressions for the purpose of making the search queries uniform and more intuitive. In addition to this, Random Forest and Bagging regression techniques are applied to the dataset to successfully predict the search relevancies. Finally, we have tried to analyze the dataset and the output obtained by visualization techniques to attain robust and meaningful results.

Keywords

Ecommerce; Search Relevance; Data Exploration; Data Preprocessing; Random Forest; Visualization; Word Cloud; Pie Chart; Histogram; Line Chart

1. INTRODUCTION

Performing analysis on big data sets becomes complex due to the size of the data and the various inter-dependent attributes and features. Also the raw data contains many issues of redundant and unclean data. The data has to be cleaned and unnecessary information has to be removed so as to get the relevant and useful data. But the processed data obtained in the output generation containing the predictions and trends, from such complex data, becomes the source of marketing strategy and generating revenue for many industries.

1.1 Importance of Search in E-commerce

E-commerce has been a booming industry for the past few years. People do online shopping to save time and effort. However it is the responsibility of the e-commerce websites to make their application safe, secure and user friendly. Their main motto is to satisfy the customer and provide the online services efficiently.
It is hassle free for the customer if he gets his products with a single mouse click without any geographical limitations. The search results need to be specific and should match to the needs of the customer in order to satisfy them so that they use the application again. Websites rely heavily on efficient search algorithms to map the resultant documents with the search query. This implicit measure improves the accuracy of the results and enhances the user experience. It is very annoying for a user to search for a document or a product online particularly on e-commerce websites and get back underwhelming responses wherein the search results are not at all what they were expecting. For instance, if a user searches for 'making pancakes' and 'frying pan' and results such as 'spatula' or let's say 'cooking oil' are displayed then a user can deduce that there is some degree of correlation between these terms. This happens many a time and there are scenarios where the product that we require isn't available at the database of the application so no results are found instead of generating near-relevant ones. In a few sites, irrelevant results are generated which annoys the customers browsing the website.

1.2 Search Relevancy Factor

The search relevance is an important concept in any website that involves search bars to search the content on their website. Google being the giant in tech industry employs complex algorithms to yield accurate search results quickly and efficiently. Similarly the e-commerce websites like Home Depot employ algorithms to yield accurate search results that match the users input.
Search relevancy is the process of sorting the results in a manner so that the documents/products that are most relevant to the search-term query are displayed first. The search relevance score in the Home Depot dataset [9] is a key factor which describes how relevant are the search results to the user. It helps us estimate if the customer has got the intended right product.
Efficient search algorithms consist of sophisticated text matching.
Some of these techniques are listed as follows:

- Text Analysis - normalizes the text from both search query and search result which makes fuzzy matching.

- Query Time Weights and Boosts - reassigns the importance of different fields.
- Phrase/Position Matching - boosting on the appearance of the entire or parts of query as a phrase [11]

2. DATA SET

The data set of Home Depot Search Relevance has been obtained from Kaggle dataset repository [10] and it includes the following attribute fields:

- **ID**: a unique Id field which represents a (Search Term, Product ID) pair
- **Product Unique ID**: an Id for the products
- **Product Title**: the product title
- **Product Description**: Text description of the product (may contain HTML content)
- **Search Term**: Search query by users.
- **Relevance**: Average of the relevance ratings for a given Id

The typical difference between the attributes of id and product uid is that the id corresponds to the unique identifier for the pair of a product and its search term whereas a product uid corresponds to a unique identifier of a particular product. Relevance attribute present in the data corresponds to the average of the relevance ratings for a given id. The data provided on kaggle is a group of dataset files like:

- Train Dataset : The train data is a training dataset containing 74067 records and five attributes with the main ones being uid, title of the product along with search term and its relevance. It basically contains the information about the products, search terms related to the product and search relevance score. The relevance is a floating point number ranging from 1 to 3 with 3 being the most relevant and 1 being the least relevant.
- Test Dataset : The test data contains 1,66,693 records with lots of duplicates and redundant data.
- Product Descriptions : The dataset of product descriptions provide data on information about the products.
- Attributes : The datasets of product descriptions and attributes provide data on descriptions of products and the extended attributes if present for a specific product.

3. RELEVANT WORK AND MOTIVATION

As discussed earlier, most of the applications involving search scenarios employ efficient algorithms to yield good search results. Big data analytics also comes into the picture here. We can employ data analysis and suitable algorithms to develop a model that predicts the relevance of search results accurately by looking at their previous data. This can help improve the market of Home Depot to provide quick and satisfiable results. This has inspired us to work on the huge dataset of Home Depot Search Relevance. The goal of the project is to predict a model that predicts the search relevance so that the team of Home Depot can improve their current search algorithms[9] . A relevance score has to be built for the combination of products and search terms available in the data. Along with that, the preprocessed and analyzed file with all attributes is also provided at the end of this project.

Many people have worked on the dataset however the speed and performance of the algorithm was very slow. Also the data was not preprocessed efficiently as it resulted in low relevance scores inspite of the highly relevant search terms in the data. In order, to avoid these loopholes, we have analysed various algorithms like SVM, SVD, Clustering etc and finally decided to go with Random Forest regression to predict accurate relevance score. The steps of cleaning, data preprocessing, feature extraction, analysis and algorithm implementation have been performed in such a manner that in spite of the huge dataset, the algorithm provided the results within a short span of time.

##########Product Statistics##########

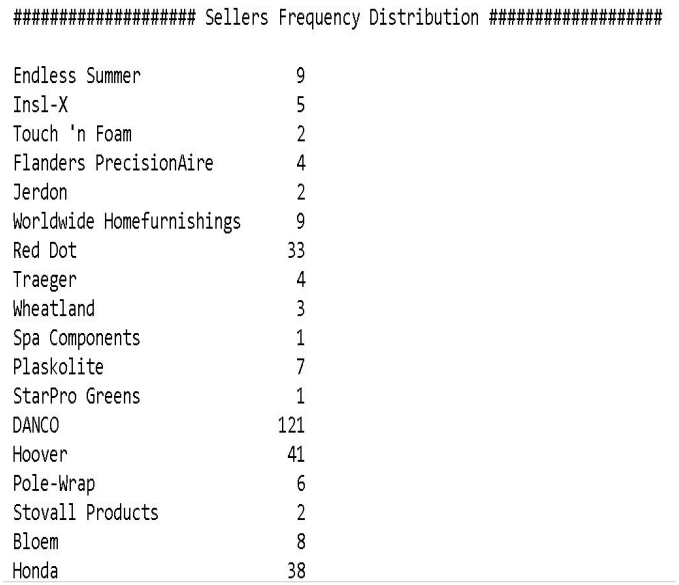
	id	product_uid	relevance	Count_Of_Product
count	74067.000000	74067.000000	74067.000000	74067.000000
mean	112385.709223	142331.911553	2.381634	1.935450
std	64016.573650	30770.774864	0.533984	1.706594
min	2.000000	100001.000000	1.000000	1.000000
25%	57163.500000	115128.500000	2.000000	1.000000
50%	113228.000000	137334.000000	2.330000	1.000000
75%	168275.500000	166883.500000	3.000000	2.000000
max	221473.000000	206650.000000	3.000000	21.000000

Figure 1: Statistics of Products

4. METHODOLOGY

4.1 Data Exploration

A data exploratory analysis was performed on the dataset to look into the attributes of the dataset and explore the trends in the data. This analysis requires the data of train, product descriptions and attributes in order to produce the statistics of each feature and provide frequency distribution for those features. After the reading of data, the training dataset and product descriptions data is merged. The resultant data frame is grouped by the product uid in order to get the actual count of products. This data is again merged. Similarly the MFG brand name feature available in the attributes dataset has its column name renamed to seller name and this data is again merged. The resultant output describes the statistics of the merged dataframe that contains the count, mean, standard deviation, min and max values for each of the features of id, product uid, relevance and



```
################## Sellers Frequency Distribution ##################

Endless Summer                9
Insl-X                        5
Touch 'n Foam                 2
Flanders PrecisionAire        4
Jerdon                        2
Worldwide Homefurnishings     9
Red Dot                      33
Traeger                       4
Wheatland                     3
Spa Components                1
Plaskolite                    7
StarPro Greens                1
DANCO                       121
Hoover                       41
Pole-Wrap                     6
Stovall Products              2
Bloem                         8
Honda                        38
```

Figure 2: `Sellers` Frequency Distribution

product count.

Now this resultant dataframe is inputted to obtain the frequency distribution for each of product titles, search term and sellers.

From these frequency distributions, it is clearly evident that most of the product titles have frequencies of 1 to 3. The frequency distribution of search terms is interesting as the highest search terms had a frequency of 13 and the lowest one just had 1. The Advanced Drainage Systems had the highest frequency in the frequency distribution of the seller names available in the data. These trends are very interesting and give the data analyst a new dimension to explore and analyze the data.

4.2 Initial cleaning and processing of data

Initially the input data is cleaned and preprocessed before the analysis. Also it is merged with the training set, and features are extracted in order to implement the suitable algorithm of Random Forest so as to predict the relevance score for the pairs of products and search terms in the test dataset.

The first step of the data analysis involves the reading of the files of training set, test data and product descriptions. All these files are loaded into the dataframes using pandas python package. After the files are loaded, we perform the steps of data cleaning and feature extraction that involves data preprocessing. In the first step of data processing, stemming is performed on the data to remove the suffixes and extract the root words (called stems) of a particular word. This process of stemming is very useful in natural language processing scenarios such as querying and search[1].
Also, various operations such as common words between two strings, getting the frequency of words, replacing the query strings with a simple convenient notation, checking if the string is a word, splitting a string into words etc., are performed. All these operations ensure that the data is properly formatted and structured.

All these functions comprise the data preprocessing step and this data preprocessing step takes place in the feature extraction step where the features of search term, product title and product description are extracted. These three attributes are merged into a single feature called product_combined_info. The length of search query and product title is computed into the data frame. Further, the common words in title and description of product are processed in the main data frame. The data preprocessing techniques were used to clean the data and make it uniform.

Some of these modulations are:

- String stemmer - When we write sentences we generally tend to use different forms of the same word to make sense grammatically. For instance, since this dataset is from Home Depot, we will consider one such example from their search queries. 'Wash', 'Washer', 'Washed', 'Washing' are indicating that the user is searching for products related to wash clothes or washing machine. However, when we the regression model encounters these separate forms of the same word, it regards them differently and thus the score may vary accordingly. This is not correct as we know what the user is trying to search for and they will expect search results that are associated with these words in any form. Thus we have used the concept the stemming for English dictionary which reduces all these similar words into their root form. Like in the aforementioned example, all the words will be reduced to 'wash'. Thus, the prediction will be more accurate as, now all the words will be in their root form. To incorporate this concept in our project we stem 'search query', 'product title', 'product description' and 'attributes' in the given data.

- Find common words - while using the different attribute fields there are lot of common words. Through this technique we try to find all the common words between attributes.

- Frequency of words - This technique is used to find all overall frequency of each word and store them. This is not directly useful for the analysis model but is used later to explore data and estimate performance metrics.

- Make changes to query - In this particular data set there are a lot of discrepancies within the search query which need to be cleaned in order to make the data uniform. For example, users have searched for 'inches', 'inch' and 'in' looking to define the size of the particular product. Even though all these mean the same thing and depict the uniform size metric but the model won't consider them as one and will regard them as separate string entities. To rectify this, we have used regular expressions to replace and substitute such ambiguous string terms in the query attributes. In this case, all these terms will now correspond to 1in universally. Similarly all the size, weight, energy and distance metrics will be cleaned.

- Replace extra characters - User while writing tend to make a lot of grammatical mistakes and and are lazy enough not to rectify these errors even upon noticing.

They want that the search algorithm should handle all such discrepancies and edge cases and treat it as a black box. Users should not be burdened with the overly complex technical aspects of the search algorithm and neither are they interested in knowing any of it. They just want accurate results. For example, users sometimes tend to use extra special characters mistakenly like '...' or '//' or '???'. Thus, as a part of data preprocessing it is absolutely necessary to clean the data of such edge cases and make it as clean and simple as possible. Therefore, we have tried to replace all such extra special characters edge cases by replacing them with suitable alternatives. '...' changes to '.', '???' is replaced with '?'.

- Split two words - The datasets obtained from the various sources are not clean and the users use different notations and abbreviations to denote a term in their search queries. Also, there can be spelling and grammatical errors in the query as well. For example, we found that in the dataset there were a quite a few instance where two words in a string were not properly spaced of joined by a period ('.') like wash.machine. Thus, we had to rectify this in the dataset and preprocess it so that the data is uniform throughout all the rows and columns.

4.3 Analysis of data

For our dataset we first had to decide whether we have to use classification or regression techniques to predict the search relevancy of the query terms. Thus, after studying upon these different techniques, we decided to use regression. Regression analysis technique takes continuous values and predicts a value from this continuous set whereas classification technique takes class labels and predicts belonging to the class. Since, our project required us to predict the search relevancy we had to choose regression.

We have used Random Forest ensemble learning method to be used as a regression technique for predicting the values. Random forests can also be called Random decision forests as it operates by creating a lot of decision trees while training the data and producing the predictions as output by taking the mean of all the decision tree values. A decision tree takes the input data and breaks it down to smaller sub samples of data and creates and incrementally develops trees. The final result is a tree with leaf nodes and decision nodes, and leaf nodes represents decision on the data. The notable decision trees are of the types ID3, C4.5 and CART which use Information Gain, Gini impurity and variance reduction as some of the tree construction metrics. Decision trees require little data preparation, is robust and using this ensures possibility to validate the model using statistical tests to verify the reliability of the prediction values of the model. However, this model tends to construct complex trees and generally overfit the data [3]. Over-fitting occurs when the criteria used to train the model is different from the criteria to validate the efficiency of a model. Thus, to rectify the limitation of over-fitting the data, random forests are used.
In this project we have made use of the ensemble random forest regression library made available by scikit-learn and we have made the model to run on a total of 12 trees with each tree having a depth of 5. Random forest rectifies each decision tree's issue of over-fitting and then we pass this to the Bagging regression algorithm. Bagging divides the initial input data into smaller 'm' data sets with each set of equal size. It uniformly samples each of the data set with the original data set along with replacement. The 'm' models are fitted and combined by averaging the output [2]. As mentioned earlier, we have created a 'hd_main' file from which drop the columns - 'search_term', 'product_title', 'hd_description', 'prod_combined_info' and then further segregate this file into subsequent train and test files. We create 'xTrain', 'yTrain', 'xTest' and 'id_test' files which are then passed as parameters to the Bagging regression model which also takes in the Random Forrest as one of the parameters. This classifier is then used to predict the search relevancy score for all the queries.

4.4 Visualization

The visualization is working over the generated relevancies and the given search terms for all the generated relevancies in the dataset. The main objective of working on this information is to show how relevant are the searches based on the search terms in the data, and also the frequency of relevancies. This gives us a good idea of which search terms are used the most and how relevant are these search terms to the product description for them. This gives us a good idea of registering search terms with lower relevancy, which can help in optimizing the algorithm of making these searches more relevant.
The different visualizations used here are Linecharts, Histograms, Wordcloud, Piecharts and Scatterplots, spanning over the visualization libraries used.
The visualization libraries used in this project are primarily matplotlib, pyplot and WordCloud. Python modules like pylab and numpy are being used, which helps in processing data for visualization and preserving those visualizations. Other libraries used for intricate function calls are csv, os and sys.

The primary fields to visualization are the relevancy for every product id, and search terms in training data, test data, and the combined data (by combining training and test data). The respective files holding all of this data are being accessed, and data is being fetched and cleaned before being processed for visualization

4.4.1 Different Visualizations

Line Chart - A line chart is a type of 2-dimensional visualization displaying data points connected by line segments. Usually the y-axis depicts a growing range of information against a set of values over x-axis. For every point of x-axis, it has a corresponding value on y-axis, which generate these data points (called 'markers') joined together by line segments. [5]

This project visualizes 3 different Linecharts. The Figure-1 and Figure-2 linecharts work around the output generated by the analysis, i.e. 'productid' and 'relevance'. 'productid' is the unique id of the 'hd_combinedfile.csv' which is being generated after clean and preprocessing. It is unique to every tuple in the above csv file. 'relevance' is the relevance calculated by the regression technique used during the generation of output. Both of these information is stored in the output file, called the 'BigDataOutput.csv'.

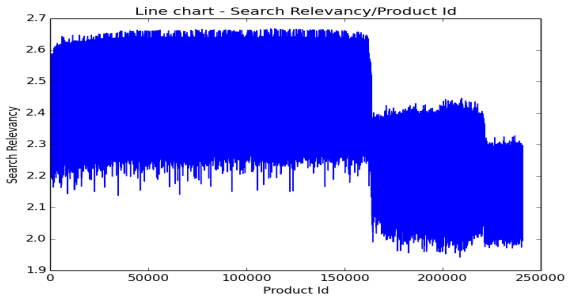


Figure 3: `Linechart - 1`

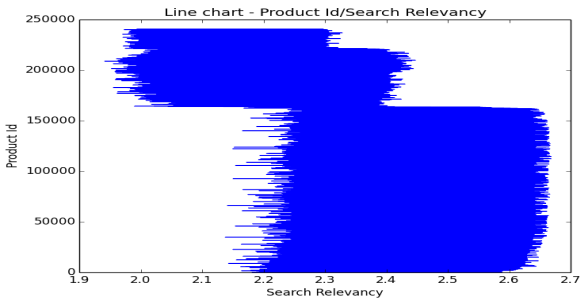


Figure 4: `Linechart - 2`

The Figure-1 linechart is being generated between search relevancies against product id. It tells us that the search relevance is more for the initial data and the latter data has comparatively lesser search relevance. This tells us that maybe this regression and bagging technique is not adequate for achieving a higher search relevancy.

The Figure-2 linechart is the other view to the first line chart, as it is between product id against search relevancies. It gives us the same information, but with a better view. We can clearly see the range of search relevancy of every product id in the data.

These 2 line charts are being generated using the 'pyplot' module of 'matplotlib'.

The Figure-3 linechart is an interesting analysis to the search relevancies generated. This linechart depicts the count of every relevancy. As the relevancies span up to 9 or 10 decimal places, generating a linechart for every one of them will give a hard to understand linechart. Thus, the relevancies are first rounded off to 2 decimal places, and then all the unique relevancies are counted. This gives the count of every unique relevancy, which shows having the maximum count with 2.5, which is a good sign.

Histogram - A histogram is a type of 2-dimensional representation of the distribution of numerical data. That means the numerical data (which is a continuous variable) is spread out over the x-axis, whose probability distribution estimate is depicted as bars. The continuous numerical data is divided into bins, over which the probability is distributed to make the histogram. The bins can have different properties depending on size, continuity, color and many more. [4]

The histogram visualization is performed for the rounded off (2 decimal places) relevancies, generated from the relevancies in the output file 'BigDataOutput.csv'. The histogram generated is the frequency of every bin in the domain of the unique relevancies. This gives us the same idea like the third linechart, i.e. the maximum frequency bin (range) of the relevancies lies around 2.5, which confirms the produc-

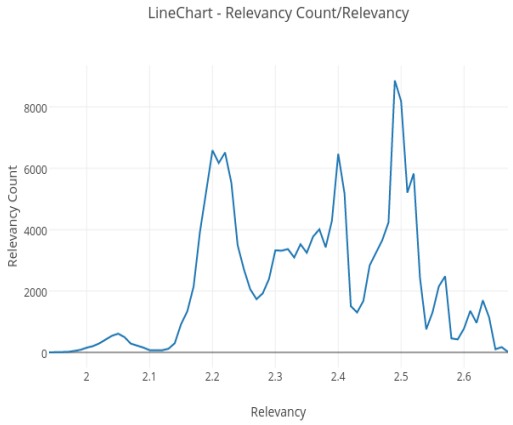


Figure 5: `Linechart - 3`

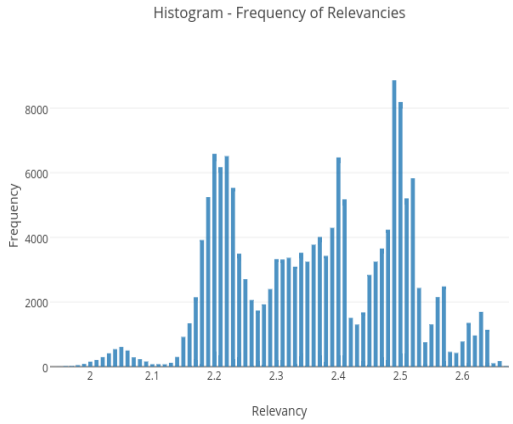


Figure 6: `Histogram`

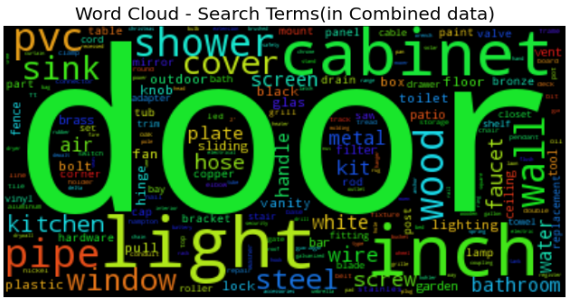


Figure 7: `Word-Cloud`

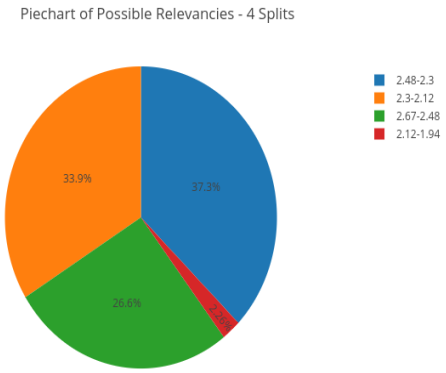


Figure 8: `Pie-Chart (4-splits)`

tivity of the prediction done by the implemented Random Forest Regression and Bagging.

Word-Cloud - A word cloud is a visual representation of text data, depicting the frequency of every text (usually words) in the text data. The larger the frequency of the text, the larger will be the size of the word in the visual representation (usually image). The text with the largest frequency is in the middle and is of the largest size. [8]

The project generates 3 different word clouds. These wordclouds are generated over the same textual data, ie. 'search terms'. 'search terms' is the search queries given in to search for a particular product. Every 'search term' has a relevance to the particular product it is been mapped to. Visualizing the 'search terms' as a word-cloud gives us an idea of the most searched terms in the dataset.

The 3 word-clouds are made from the same textual data, but generated from three different files, namely training data file ('train.csv'), testing data file ('test.csv') and the combined data file ('hd_combinedfile.csv'). Careful analysis of the word cloud shows minute difference in sizes of a small number of words, for example the word 'cover'. This difference is due to the different sets of data in the training, test and combined file. The minute anomalies in sizes occur more with the training data, against the test and combined data, which are more similar. This can easily imply that the testing data is large enough than the training data, as the search terms frequencies are more similar in the test data and combined data.

Pie-Chart - A pie chart is a circular visualization, which spans over a range of data to illustrate it's numerical proportion. Every slice of the circle represents the quantity percentage of the numerical bin (sub-range of numerical data), which is visually proportional to the arc length, central angle and area of the circle, ie. pie chart.[6]

This project visualizes three pie charts, spanning over the splits made to the range of relevancies. These pie charts use

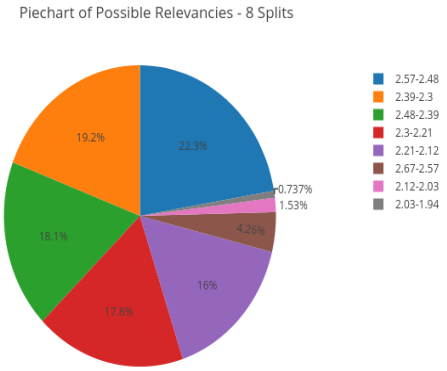


Figure 9: `Pie-Chart (4-splits)`

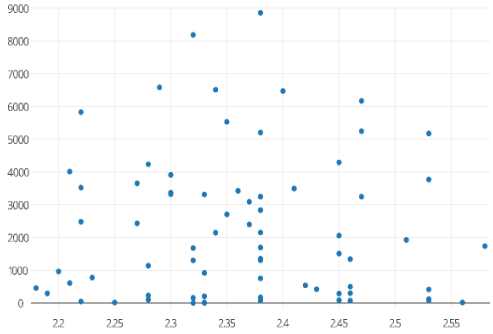


Figure 10: `Scatter-Plot`

'relevance' stored in the output file 'BigDataOutput.csv'. The three pie charts depicts the count percentage of the different sub-ranges of relevancies generated. They vary according to the splits made to the entire range of relevancies, here in the visualization is over 2, 4 and 8 splits. The splits can be increased to show more intricate count percentage, but those are not required over the given dataset. This gives us the good idea of the relevancy range of the entire dataset.

Scatter Plot - A scatter plot is a type of 2-dimensional plot using Cartesian coordinates. The scatter plot is usually used for multiple variables for a set of data, to present there values on the two axis of the Cartesian plane. [7]

The project builds a scatterplot for the frequency of relevancies, in the range of generated relevancies in the output. The scatterplot gives us the count of every unique relevancies and also the rate at which they are fluctuating. The scatterplot generated is usually rising, which depicts a high number of better search relevancies, implying the analysis and prediction give us positive results.

5. OUTPUT AND RESULTS

The random forest and bagging regression models produce output of a direct mapping of the id's of the search query terms and their relevancy scores in the range of 1-3 wherein 1 represents low relevancy and so forth. Good search query relevancy with the results displayed determines superior results and helps the customer find the item that they are looking for quickly. It is very important for web based application especially an e-commerce website to have an advanced search algorithm in order to display better results. The output attained through this project helps Home Depot website to analyze the efficiency of their algorithm and grasp a better understanding of what needs to be done in future to help enhance their user experience to be as seamless efficient as possible. Also the algorithm of Random Forest ensures that the huge dataset can be analyzed in a short span of time compared to other algorithms such as SVM, kmeans etc.,

The other output file is the combined attributes file whose features are preprocessed and extracted. As this is a Big Data project, it is of utmost importance that we visualize the data from the output to represent meaningful information in a manner which is easy to comprehend and make useful derivations from. For this, we have tried to plot the search relevancies and their mapping to search-term using word-cloud, line-chart, histogram, pie-chart and scatter-plot.

6. SOFTWARE DOCUMENTATION

The dataset that we have used for this project is *Home Depot Product Search Relevance* from Kaggle data repository. This data is publicly avaolable and can be easily accessed from the url:
https://www.kaggle.com/c/home-depot-product-search-relevance

We tried writing a fetchdata.py script but we faced an issue when this python script accesses the provided url and tries to download the particular file, kaggle asks for user authentication which we can't provide in the script. Therefore, we have pushed the dataset to our project-012 gitlab repository in the dataset directory. So if the user clones the entire project-012 repository, the dataset is also downloaded to their local machines and can easily access it from there. The python scripts are written to address this issue and will fetch the data directly from the 'dataset' directory path without the user having to perform any action or edit anything manually.

This project is completely built using Python 2.7 language and it's associated libraries which include pandas, numpy, sklearn, matplotlib, os, wordcloud to name a few. Visualization is also done using python 2.7 along with the use of plotly library. All the related code, dataset, output (with images) are documented into separate directories. We have used gitlab (project-012 repository) for version control as per the requirement of Professor Laszewski. In order to run

the project files, user will have to first install all the dependencies i.e the python libraries, on their local machine using the terminal. The user will have to install the 'requirements.txt' file to install all the python library dependencies required to execute the project successfully. Alternatively, user can install libraries individually also, for example: "pip install sklearn" for installing scikit-learn machine learning algorithm. Further, the user will have to clone this repository from gitlab on to their local machine. Using the terminal, go to the cloned project-012 folder and then go to code directory. To run the project, give the command "python -W ignore homeDepotMain.py".
We are using plotly for generating graphs and since the data points are so large that plotly package starts to generate warnings to notify the user about it. So, it is recommended to execute the script with "-W ignore" flag to remove all the unwanted warnings. This file upon successful execution will generate "BigDataOutput.csv" which has the relevancy score for all the unique search id's for the corresponding search queries. In addition, the program will output "hd_combinedFile.csv" which has the all the extracted features and other performance metrics. Also, the program will generate various visualization such as word-clouds, charts and graphs. As we have used plotly's python library for visualization of output, it will open all of these files in the default browser of the user's local machine using *ahmedss's* (Syed Saif Ahmed) public plotly profile. In the case where a message is prompted, the user just has to close the alert box and then access all the plotly files generated. It is very simple and intuitive to use and the user won't encounter any hassles while accessing the images.

7. CONCLUSION

Through this project we have used Random Forest and Bagging regression techniques to successfully predict the search relevancy scores for search-term queries provided in the dataset. For the initial part of the project we have preprocessed the raw data present in the dataset and applied techniques like word stemming for the English dictionary, removed special characters, replaced redundant data to make the dataset clean and uniform.
From the output generated by the program we can conclude that the relevancy scores are all between the range of 1-3 and the majority of the score lies in between 2-3 indicating that there are less search-term queries which had underwhelming accuracy for the search results returned. The word-cloud generated using the output file indicates that the user searched for the term 'door' and door related products the most when quantitatively compared with all the other search-terms. Finally, visualization graphs generated depict the obtained results and provide meaningful information and visually appealing mapping to the resultant data.

8. FUTURE WORK

Even though we have tried to be very comprehensive and inclusive in our approach towards the concepts applied in the project, we still feel that there is some scope for future related work. We have used Random Forest and Bagging regression techniques to predict the search relevancy scores, we could also use Decision Trees regression model along with Root Mean Squared Error method to fine tune the score prediction more efficiently. Our team would like to work on these improvements in near future and make this project more robust and efficient so that we could improve the universal search algorithms.

9. REFERENCES

[1] Python stemming package and usage. Web Page.
[2] Wiki page of bootstrap aggregation (bagging) algorithm. Web Page.
[3] Wiki page of decision tree. Web Page.
[4] Wiki page of histogram. Web Page.
[5] Wiki page of line chart. Web Page.
[6] Wiki page of pie chart. Web Page.
[7] Wiki page of scatter plot. Web Page.
[8] Wiki page of tag cloud. Web Page.
[9] Home depot search relevance. webpage, 2016.
[10] Home depot search relevance datasets. Web Page, 2016.
[11] D. Turnbull. Search relevance definition. web page, June 2014.