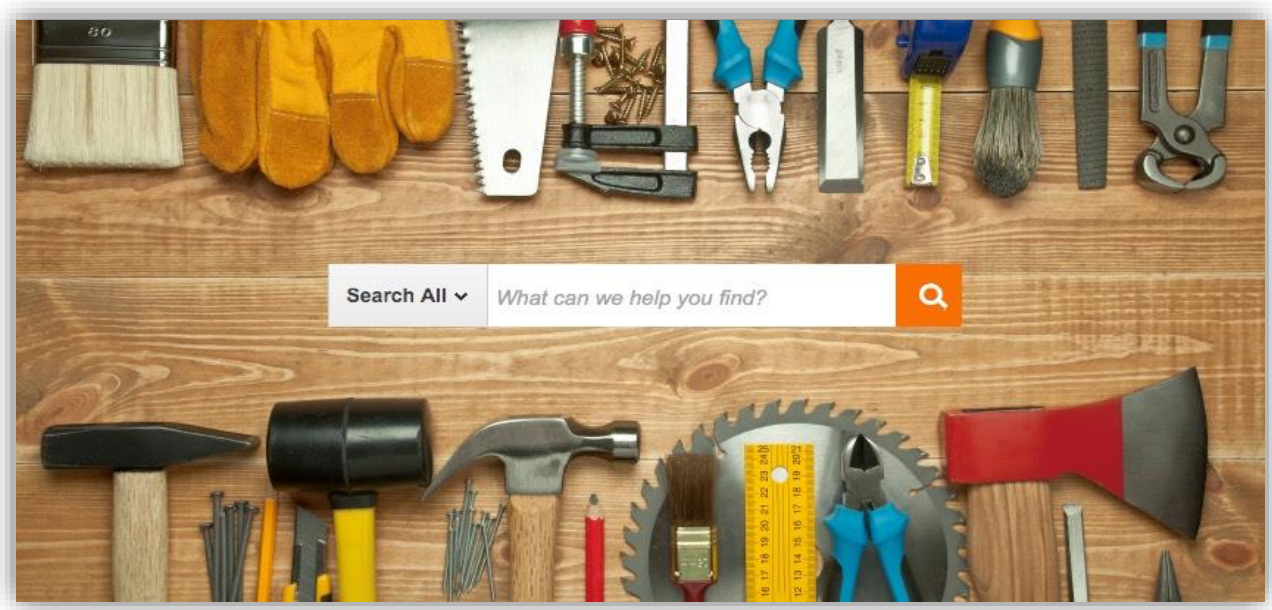


Home Depot Product Search Relevance

Group - 02 Members:

Vikas Srikanth (VXS180022), Gowri Dath (GXD180003), Aditi Prakash (AXP180022),

Sindhu Bindiganavalae Manjunath (SXB170071), Mitali Bharali (MXB180027)

TABLE OF CONTENTS

Enhancing Online User Experience	3
Project Motivation/Background	4
Executive Summary	5
History	6
Industry Analysis	7
Swot Analysis	10
Data Description	11
Exploratory Data Analysis	12
Text Cleaning and Preparation	14
Feature Engineering	16
Graphs for Feature Analysis	18
Graphs for Training Feature Analysis	19
Graphs for Testing Feature Analysis	20
Training set Heat Map	22
Testing set Heat Map	23
Data Modeling	24
Future Scope	28
Findings and Managerial Implications	29
References	30

ENHANCING ONLINE USER EXPERIENCE

Why a search relevance model?

- Bad shopping experiences come from the difficulty in finding the right products
- When customers are unfamiliar or unclear with the products they want, it makes the searching harder and the shopping experience annoying
- However, if online retailers can more accurately predict the relevance between search terms and products and pop out the products that can better match customers' need, this is extremely attractive and interesting
- Therefore, many online retailers are working on such a relevance model

PROJECT MOTIVATION/BACKGROUND:

Have you ever tried a site search and wondered about the accuracy of the results? Do you find yourself feeling frustrated and leaving when the search doesn't return what you're looking for? If yes, you've just experienced bad search relevancy.

Our team has made an analysis regarding Predict Search Relevance using Machine Learning for Online Retailers. The relevance between search/product pair can be evaluated by building a model based on search/product pair and its relevance score and use it to predict the relevance score of out-of-sample pairs.
We obtained several interesting results in our analysis.

EXECUTIVE SUMMARY:

Large online retailers typically use query-based search to help consumers find information/products on their websites. They can use technology to provide users with a better experience. Because they understand the importance of search relevance, and that long and/or unsuccessful searches can turn their users away because users are accustomed to and expect instant, relevant search results like they get from Google and Amazon.

In our project, we have implemented a search relevance algorithm for Home Depot's retail products so as we can retain its customers and ease out the process of checkout.

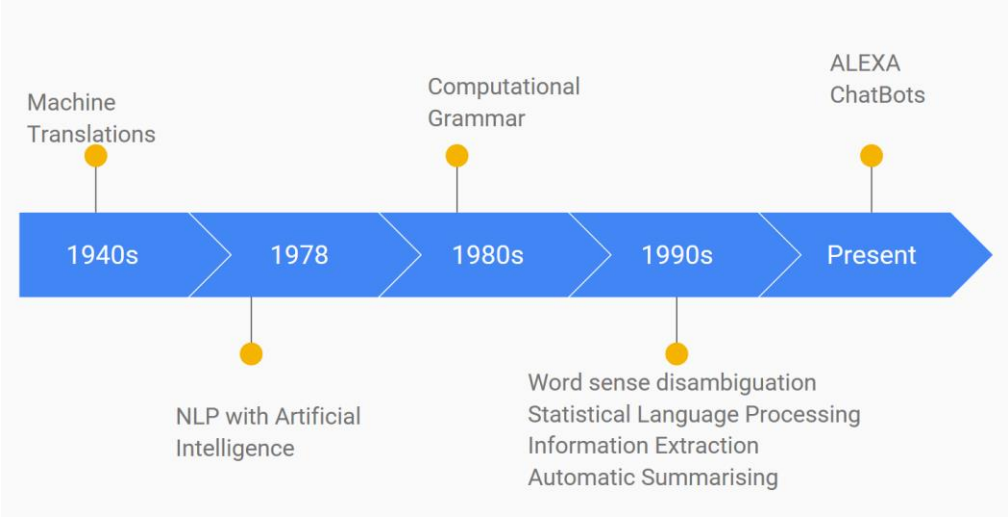
We predicted the relevance score, from 1-3 as encoded below:

- 1 Low Relevance Score - Unhappy Customer
- 2 Mildly Relevant - Neutral
- 3 High Relevance Score - Happy Customer

Techniques used in this is general data cleaning in python, text cleaning and NLP techniques and then using feature and distance measures to predict the score.

HISTORY

Natural Language Processing (NLP), a subset technique of Artificial Intelligence which is used to narrow the communication gap between the Computer and Human. It is originated from the idea of Machine Translation (MT) which came to existence during the second world war in 1940s. Then a lousy era came for MT/NLP during 1966, this fact was supported by a report of ALPAC, according to which MT/NLP almost died because the research in this area did not have the pace at that time. This condition became better again in the 1980s when the product related to MT/NLP started providing some results to customers. In the 1980s the area of computational grammar became a very active field of research which was linked with the science of reasoning for meaning and considering the user 's beliefs and intentions. In the period of 1990s, the pace of growth of NLP/MT increased. Grammars, tools and Practical resources related to NLP/MT became available with the parsers. The research on the core and futuristic topics such as word sense disambiguation and statistically colored NLP, the work on the lexicon got a direction of research. This quest of the emergence of NLP was joined by other essential topics such as statistical language processing, Information Extraction and automatic summarizing.



INDUSTRY ANALYSIS

With the advent of internet and internet industries, online retailers and internet media, search relevance has attracted huge attention and importance and snowballed into an important aspect of customer web traffic retention. A correct search term relevance can help the website retain its traffic, convert into potential buyers and help recommend customers similar products. While online streaming website like Netflix concentrates on the later with collaborative filtering, it's the online retailers whose search relevance algorithm needs to be extremely good. Walmart, Amazon and other newbies, all compete to retain their customers with a strong relevance algorithm. For established companies like the former might as well be a different case, but imagine you just started off with your online e-commerce website and if you have a hideous search relevance algorithm, it can cost you the customer and a bad reputation to start with. 72% of sites fail ecommerce site search expectations and 70% of (desktop) ecommerce search implementations are unable to return relevant results for product-type synonyms (requiring users to search using the exact same jargon as the site) and 34% don't return useful results when users search for a model number or misspell a word with just a single character in the product title. Amazon is the one that has prevailed with one of the most effective search relevance algorithms.

AMAZON

Eric Schmidt of Google has been famously quoted saying, “Almost a third of people looking to buy something started on Amazon—that’s more than twice the number who went straight to Google.”

Two factors determine how a product is displayed in Amazon search: relevance and performance.

Search relevance in Amazon can be found in things like:

- Titles
- Bullet points
- Product description
- Photos

What is Amazon doing right?

- Product titles: They are not more than 200 characters, are very selective and keyword rich as possible. They have brand name, size, color and constructive material.
- Unfortunately, product descriptions get buried under Amazon’s suggestions for other products and special offers. But the product description is the perfect place for you to go into detail about how your product works.
- Things to avoid:
    - Keyword stuffing
    - Mentioning competing products

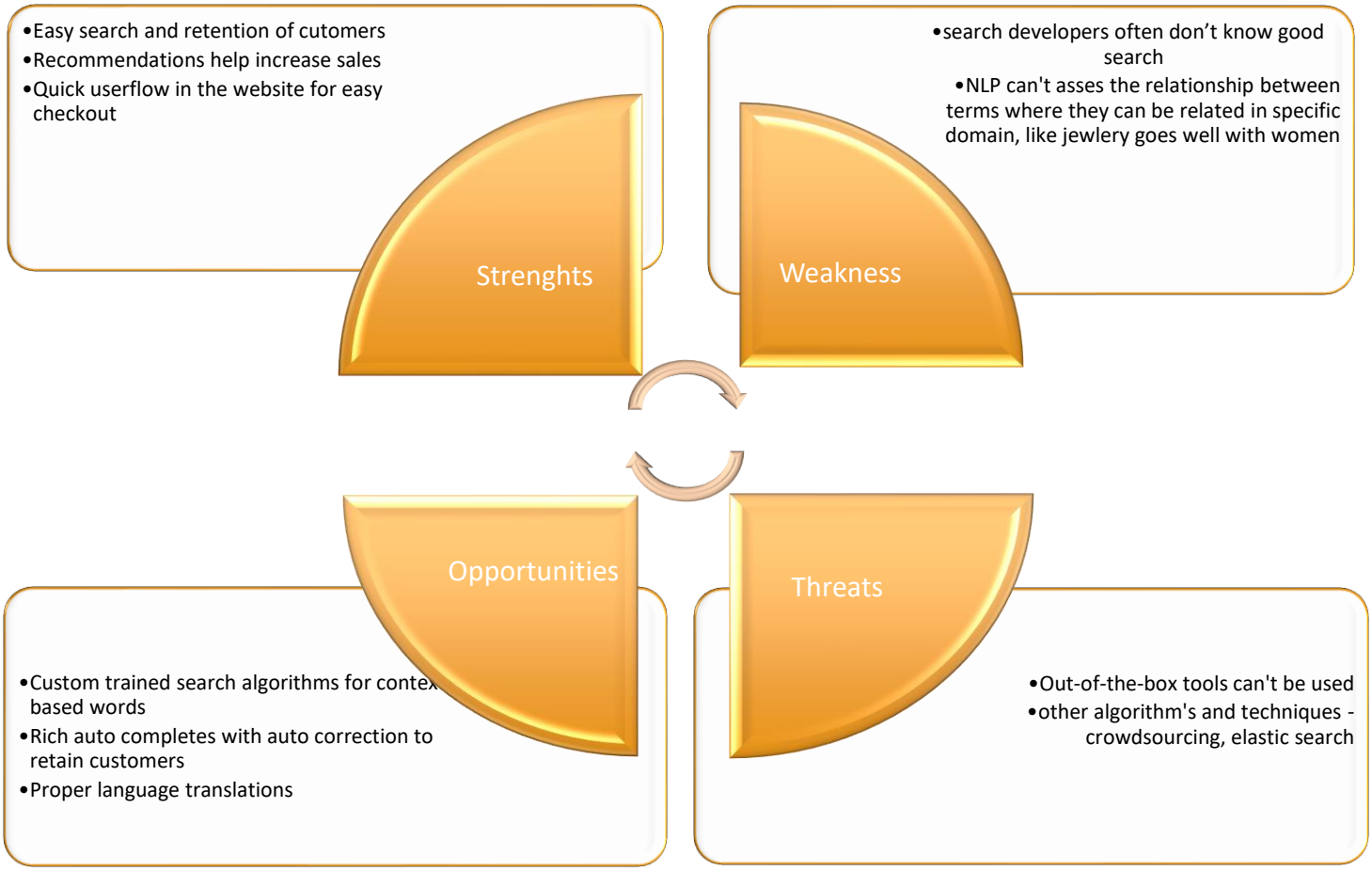
Most commercial ecommerce platforms offer some degree of native site search out-of-the-box. While native search engines' features and functionality vary, they *at a minimum* support indexing of product titles, descriptions and category associations, with the ability to autocorrect spelling, fuzzy-match (handle plural queries and stemming) and recognize synonyms (automatically or through a configurable dictionary).

Some native engines allow merchants to apply more fine-tuning, such as custom keyword tags, variant labels, autocomplete / autosuggest and relevance control through boost and bury rules.

Many enterprise ecommerce suites and third-party site search vendors offer advanced features like semantic matching, natural language processing, federated search, *searchandising* (covered in Chapter 21 of Ecommerce Illustrated) and personalization. The most advanced systems use machine learning to continuously optimize search relevance and can incorporate Big Data sources into their algorithms.

SWOT ANALYSIS

In order to get a holistic view and clarity of what this can impact and influence directly and indirectly, we use swot analysis, and we try to implement this to our algorithm. The purpose of swot analysis is to create a synthetic view of your current state. With swot, we analyze both helpful and harmful aspects of the external and internal impacts. Helpful aspects being strengths for internal impact and opportunity as external impact, harmful aspect being Weakness as internal impact and Threat as external impact.



DATA DESCRIPTION:

The data set contains a number of products and real customer search terms from Home Depot's website. To create the ground truth labels, Home Depot has crowdsourced the search/product pairs to multiple human raters.

- Training Dataset (74.1k x 5) - contains products, searches, and relevance scores
- Test Dataset (167k x 4) - contains products and searches. We are predicting the relevance score for these pairs
- Product Descriptions (124k x 2) - contains a text description of each product. We merged this table to the test set via the product_uid
- Attributes Dataset (2.04m x 3) - provides extended information about a subset of the products (typically representing detailed technical specifications)
- Sample Submission (167k x 2) - a file showing the correct submission format
- Relevance Instructions - the instructions provided to human raters

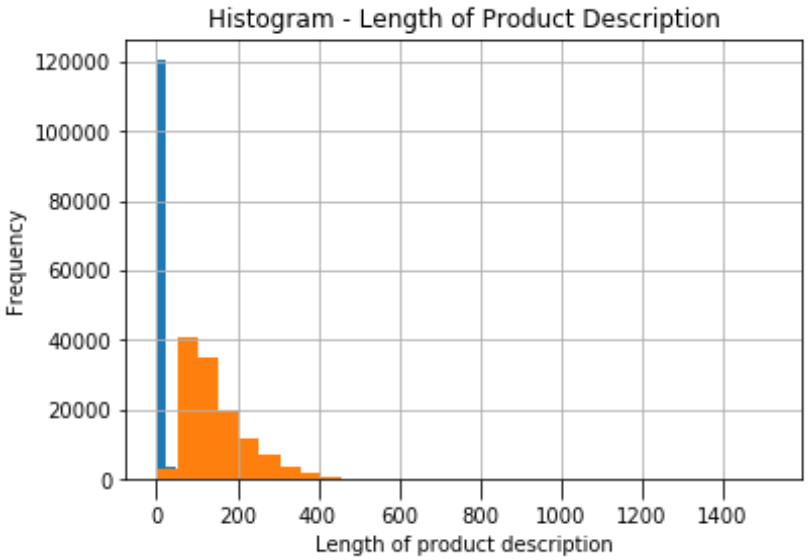
EXPLORATORY DATA ANALYSIS:

Histogram 1:

Length of Product Description

Blue bars represent the frequency of digits

Orange bars represent the frequency of alphabets

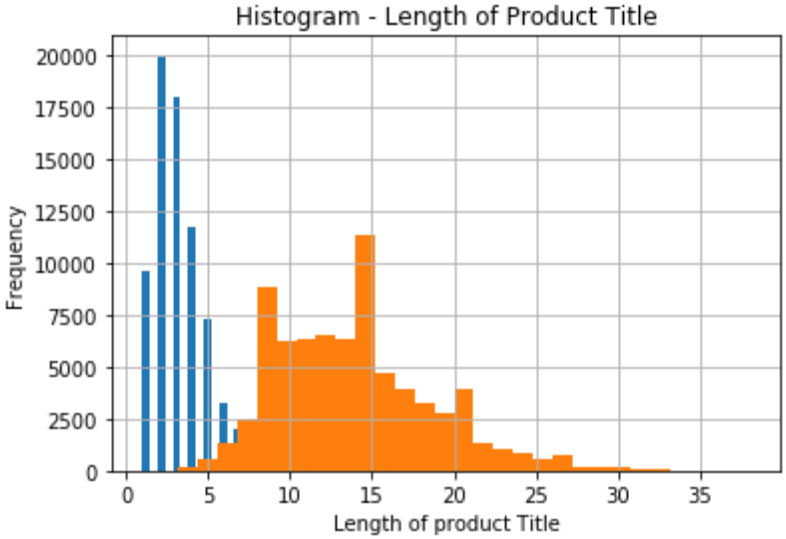


Histogram 2:

Length of Product Title

Blue bars represent the frequency of digits

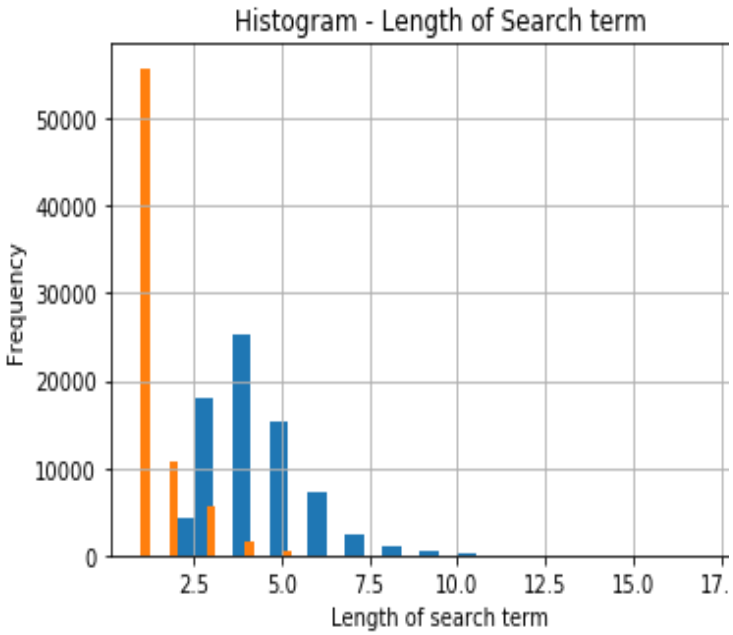
Orange bars represent the frequency of alphabets



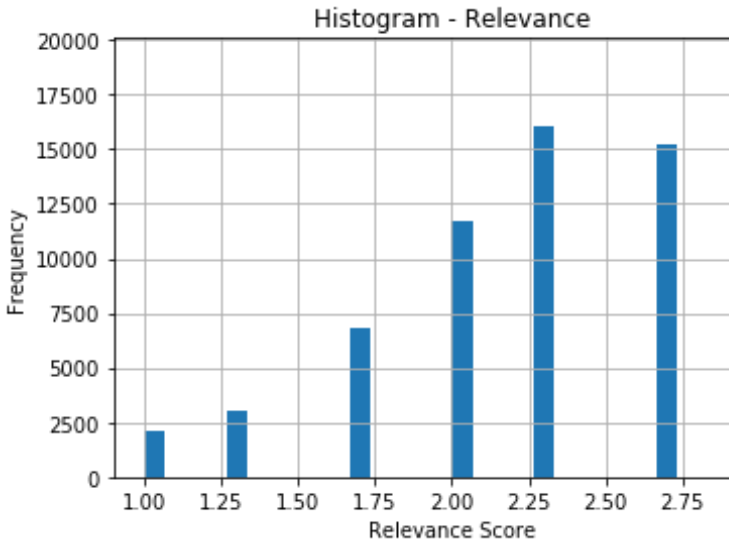
Histogram 3: Length of Product Title

Blue bars represent the frequency of digits

Orange bars represent the frequency of alphabets




```
Out[16]: 3.00    19125
         2.33    16060
         2.67    15202
         2.00    11730
         1.67     6780
         1.33     3006
         1.00     2105
         2.50       19
         2.25       11
         2.75       11
         1.75        9
         1.50        5
         1.25        4
         Name: relevance, dtype: int6
```



TEXT CLEANING AND PREPARATION:

We cannot work with the text data in machine learning so we need to convert them into numerical vectors.

Text data needs to be cleaned and encoded to numerical values before giving them to machine learning models, this process of cleaning and encoding is called as Text Preprocessing.

Tokenization is the process of splitting the given text into smaller pieces called tokens. Words, numbers, punctuation marks, and others can be considered as tokens.

The basic text cleaning we carried out were:

- converting all letters to lower case
- Removing punctuations, accent marks and other diacritics
- Removing stop words, sparse terms, and particular words
- Part-of-speech tagging (POS), it aims to assign parts of speech to each word of a given text (such as nouns, verbs, adjectives, and others) based on its definition and its context
- Lemmatization of the words, the aim of lemmatization, like stemming, is to reduce inflectional forms to a common base form. As opposed to stemming, lemmatization does not simply chop off inflections. Instead it uses lexical knowledge bases to get the correct base forms of words

- Stemming the words, it is a process of reducing words to their word stem, base or root form (for example, books — book, looked — look)

Fix Casing :	Hammer > hammer
Remove Symbols:	ft. > ft
Remove Stop Words:	hammer for nails > hammer nails
POS Tagging:	hammer > [hammer,noun]
Lemmatization:	drills > drill
Stemming:	running > run

The advanced text cleaning we carried out were:

- Standardizing numbers, converting words in number form to numeric
- Standardizing measurements, converting size information of product to appropriate numeric representation
- Splitting joined words into separate words
- Spelling correction: In natural language, misspelled errors are encountered. Companies like Google and Microsoft have achieved a decent accuracy level in automated spell correction. One can use algorithms like the Levenshtein Distances, Dictionary Lookup etc. or other modules and packages to fix these errors

Standardize Numbers :	Five > 5
Standardize Measurements:	2 feet by 4 inches > 2x4
Split Joined Words:	wiremesh > wire mesh
Correct Spelling:	insullation > insulation

FEATURE ENGINEERING

Feature extraction may be the most important part in a machine learning. In this section, we have made use of many techniques in both information retrieval and natural language processing to extract the effective features from the given corpus.

First, we calculated the number of common words from search query which appears both in product_title and product_description:

- words in title: common words between search terms and title of the product
- words in description: common words between search terms and title of the description
- words in brand: common words between search terms and title of the brand

Second, we computed Edit Distance from search query which appears both in product_title and product_description

For a certain unseen search term, we first calculated its edit distance with the vocabulary of the whole corpus. Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other. Then, I replace the unseen term with the word with minimal edit distance. If there are multiple words with the same minimal edit distance, I choose the one with the highest frequency in the whole corpus.

Third, we computed the Cosine Similarity between search query, product_title and product_description

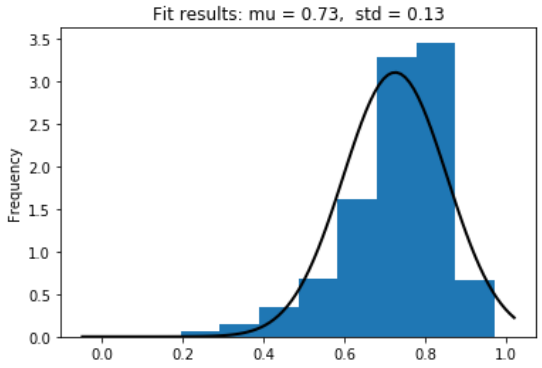
Cosine similarity calculates similarity by measuring the cosine of angle between two vectors. With cosine similarity, we need to convert sentences into vectors.

Cosine Similarity determines the dot product between the vectors of two documents/sentences to find the angle and cosine of that angle to derive the similarity. Here we are not worried by the magnitude of the vectors for each sentence rather we stress on the angle between both the vectors. So if two vectors are parallel to each other then we may say that each of these documents are similar to each other and if they are Orthogonal (An orthogonal matrix is a square matrix whose columns and rows are orthogonal unit vectors) then we call that the documents are independent of each other.

Fourth, we computed the Jaccard Similarity between search query, product_title and product_description

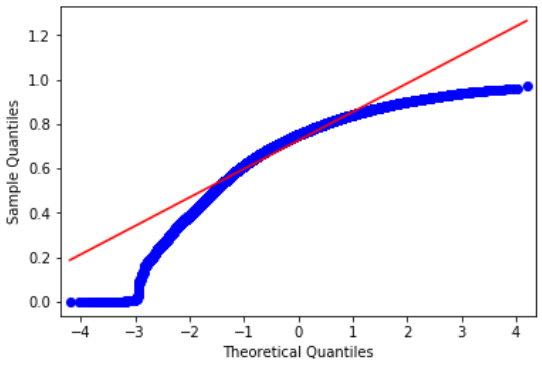
Jaccard Distance is a measure of how dissimilar two sets are. Lower the distance, more similar are the two strings.

GRAPHS FOR FEATURE ANALYSIS:



Histogram of cosine Distance:

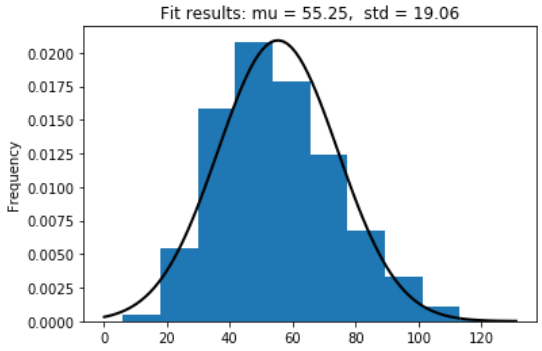
The frequency distribution of the cosine distance is normally distributed, with a mean of 0.73 and standard deviation of 0.13



qqplot of Cosine Distance:

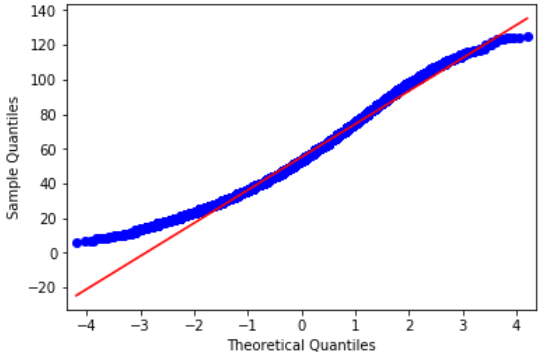
If the line is close to the data point line, it has a normal distribution

Here, it is not normally distributed



Histogram of Shared Words:

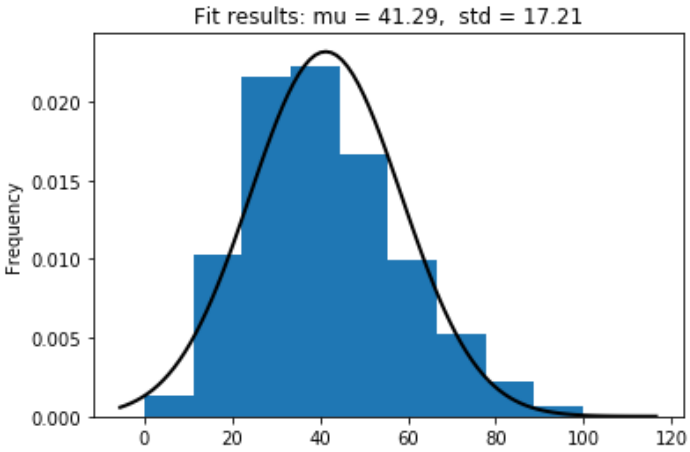
The frequency distribution of the Shared Words is normally distributed, with a mean of 55.25 and standard deviation of 19.06



qqplot of Shared Words:

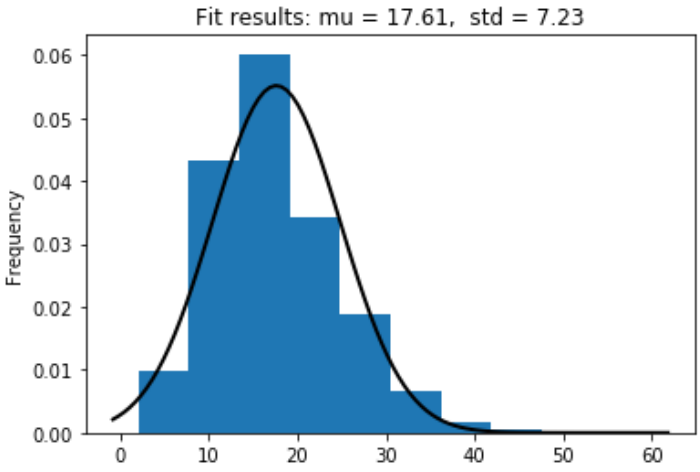
The line is close to the data point line, therefore it is closely related, that means it has a normal distribution

GRAPHS FOR TRAINING FEATURE ANALYSIS



Histogram of Edit Distance (Search term, Product Title)

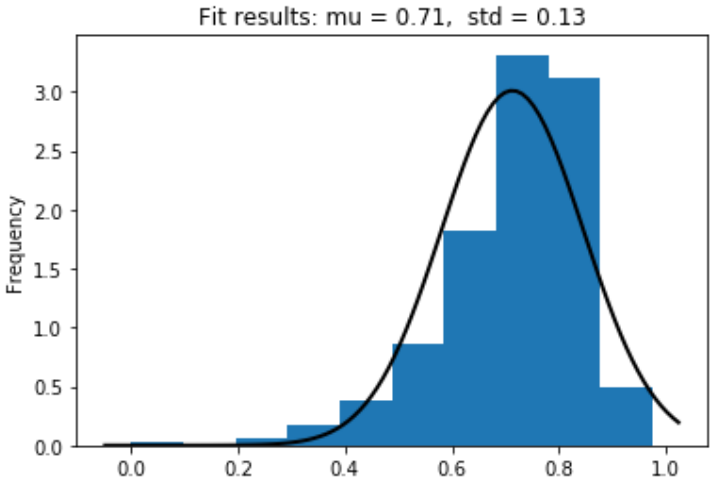
The frequency distribution of the edit distance is normally distributed, with a mean of 41.29 and standard deviation of 17.21



Histogram of Search Query Length

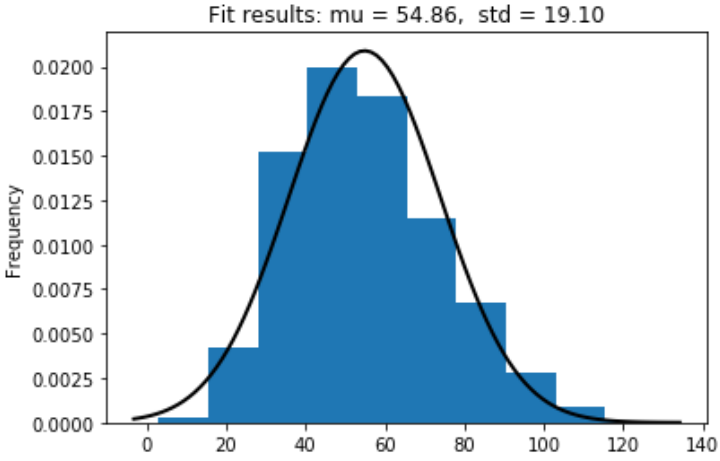
The frequency distribution of the search query length is normally distributed, with a mean of 17.61 and standard deviation of 7.23

GRAPHS FOR TESTING FEATURE ANALYSIS



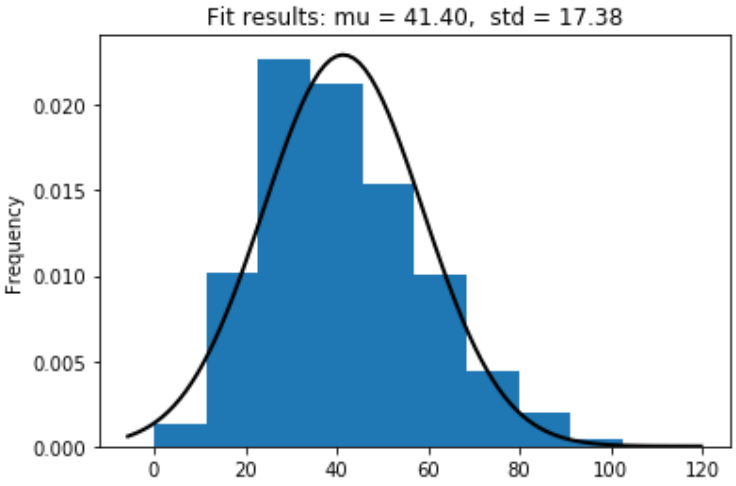
Histogram of Cosine distance:

The frequency distribution of the Cosine Distance for the testing feature is normally distributed, a little skewed towards the left, has a mean of 0.71 and standard deviation of 0.13



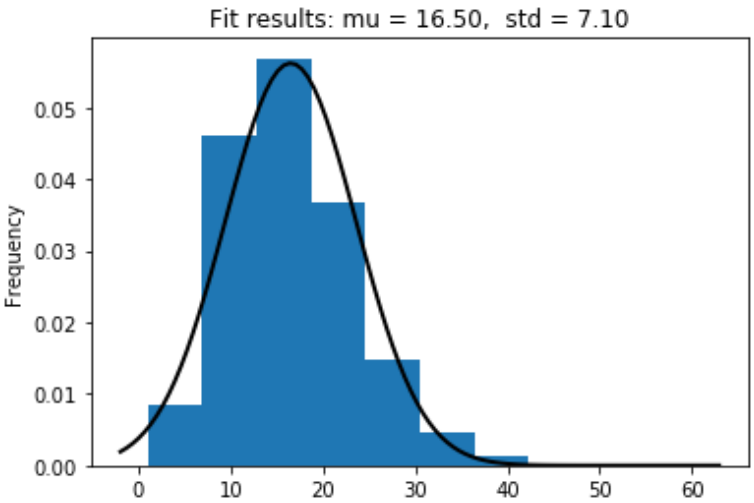
Histogram of Shared words:

The frequency distribution of the Shared Words for the testing feature is normally distributed, has a mean of 54.86 and standard deviation of 19.10



Histogram of Edit Distance (Search Term Vs Product Title):

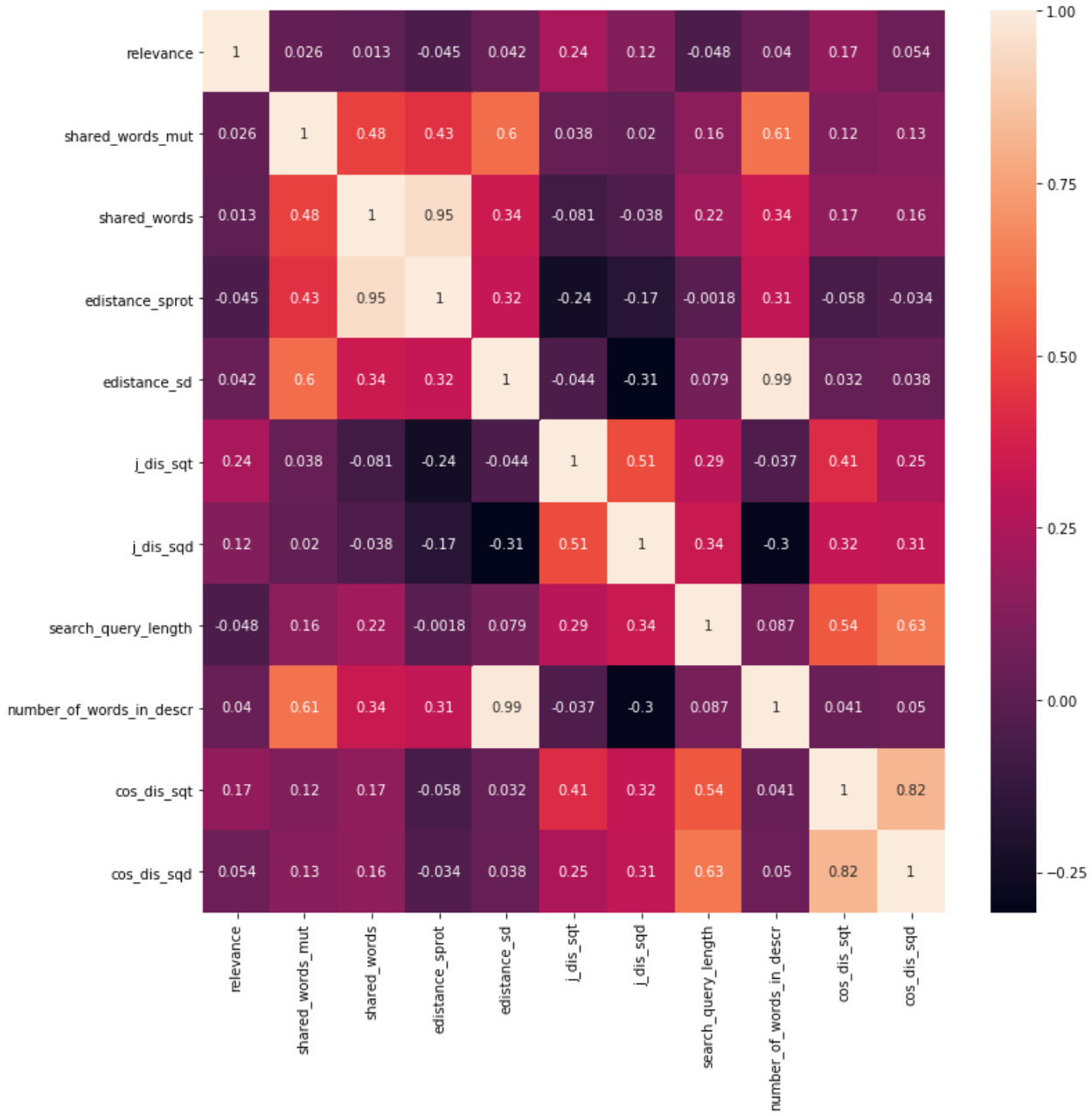
The frequency distribution of the Edit Distance for the testing feature is normally distributed, has a mean of 41.40 and standard deviation of 17.38



Histogram of Search query:

The frequency distribution of the Search Query for the testing feature is normally distributed, has a mean of 16.50 and standard deviation of 7.10

TRAINING SET HEAT MAP

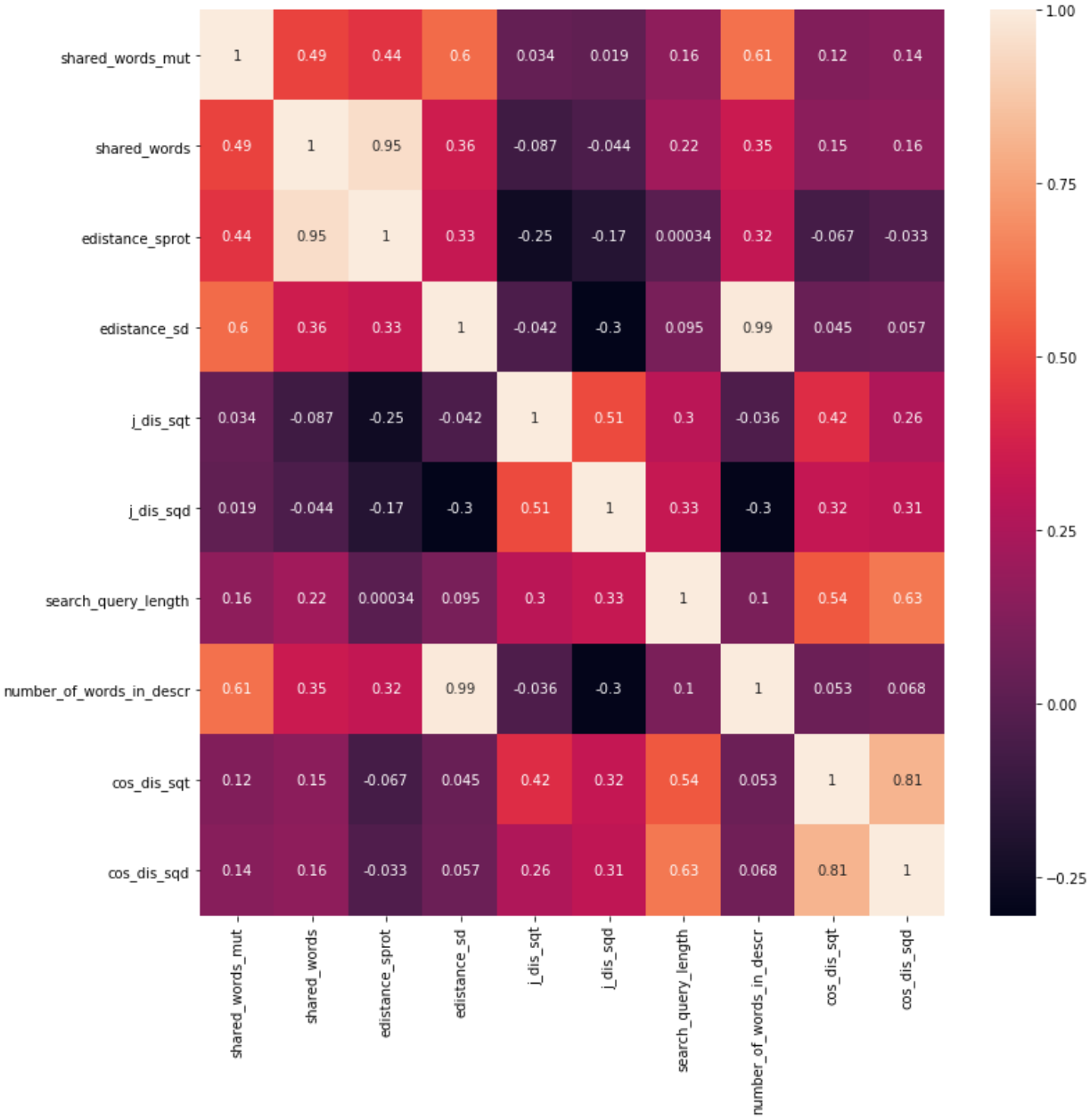


The heat map shows the correlation matrix for the training data set:

Shared_words and edistance_sprot are highly correlated at 0.95

edistance_sd and number_of_words_in_descr are highly correlated at 0.99

TESTING SET HEAT MAP



The heat map shows the correlation matrix for the testing data set:

Shared_words and edistance_sprot are highly correlated at 0.95

edistance_sd and number_of_words_in_descr are highly correlated at 0.99

DATA MODELLING

We used three models to estimate the predictive power and accuracy of the model

- Random Forest
- Naïve Bayes
- Multiple Linear Regression

Random Forest Regression:

Random Forest is a part of ensemble learning model, usually a subset of Decision Tree base model. It, like any other decision tree, makes several subsets (more than one) of training set picked at random (like bagging models) and the makes prediction for each of the subsets. The final accuracy of the model is then formulated as an average of all the subsets (for regression) and as a majority vote for classification models. It has high computational power, and hence is widely used and popular machine learning technique.

RandomForestRegressor

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from math import sqrt

rfr = RandomForestRegressor(n_estimators = 3, n_jobs = -1, random_state = 17, verbose = 1, criterion='mse')
rfr.fit(X_train, y_train)

y_pred = rfr.predict(X_test)
model=rfr.predict(X_train)
print(y_pred)

print("Number of predictions:",len(y_pred))

meanSquaredError=mean_squared_error(y_train, model)
print("MSE:", meanSquaredError)
rootMeanSquaredError = sqrt(meanSquaredError)
print("RMSE:", rootMeanSquaredError)
pd.DataFrame({"id": id_test, "relevance": y_pred}).to_csv('submission.csv',index=False)
```

We have used hyper parameters like n_estimators, n_jobs and random state for the random forest as shown in the picture above. After fitting the model to our X_train, we predict it for both MSE and RMSE values.

For our model, we get the Random Forest accuracy (through RMSE) as 0.57807.

1	id	relevance
2	1	1.776667
3	4	2
4	5	2.333333
5	6	2.223333
6	7	1.666667
7	8	2.333333
8	10	2.666667

Relevance scores predicted by Random Forest Model

Naive Bayes:

This algorithm comes from the Bayes theorem of probability, of conditional probability. However, it usually takes the situation ideally, with an assumption as all variables are independent of each other. Its popular use comes in Text Classification. Despite the naïve assumption, it is one of the most powerful supervised learning model for predictive modelling.

```
#Naive Bayes
from sklearn.linear_model import BayesianRidge

gnb = BayesianRidge()
param_grid = {}
model_nb = sklearn.model_selection.GridSearchCV(estimator = gnb, param_grid = param_grid, n_jobs = -1)
model_nb.fit(X_train, y_train)

y_pred = model_nb.predict(X_test)
pd.DataFrame({"id": id_test, "relevance": y_pred}).to_csv('submission.csv',index=False)
```

We have used model evaluation technique like GridSearch with Cross Validation to estimate the right parameters for the model to have a good biase-variance trade off.

For our model, we get the Naïve Bayes accuracy as 0.49694

1	id	relevance
2	1	2.054608
3	4	2.105725
4	5	2.574921
5	6	2.724375
6	7	2.493083
7	8	2.192348
8	10	2.64861

Relevance scores predicted by Naïve Bayes Model

Multiple Linear Regression:

The simplest in terms of model complexity is the linear Regression. It is easier also since there is no tuning parameters in the model. Even when there is always a chance of underfitting in linear regression, it is most widely used model.

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression(n_jobs = -1)
model=lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

pd.DataFrame({"id": id_test, "relevance": y_pred}).to_csv('submission.csv',index=False)
```

With an n_jobs = -1 as our parameter for linear regression, we fit our training set to it.

For our model, we get the Linear Model accuracy as 0.49692

With lowest RMSE, we choose linear model as our predictor for our dataset.

1	id	relevance
2	1	2.054543
3	4	2.104789
4	5	2.574086
5	6	2.723245
6	7	2.491682
7	8	2.191429
8	10	2.648653
9	11	2.655394

Relevance scores predicted by Multiple Linear Regression Model

FUTURE SCOPE

We notice that our accuracy isn't very high. While this remains within the industry standards of models, however this can also be due to other explanatory variables that influence the score. For this, we are required to regularize the parameters of the model, thus reducing the weightage of the coefficients in the regression. Regularization terms like alpha, penalty function of L1, L2 norm can be used to do the same. Mostly L1 helps to reduce overfitting while L1 helps in removing correlations and feature selection.

Higher order models from deep learning like Xgboost and Gradient Boost can be used to improve the accuracy. Bagging and Pasting can also be used. Bagging chooses randomly the data points with replacement, while pasting is used with replacement. Boosting is a sequential way of ensemble learning where the weightage is given to weak learners, and eventually improves the score on that.

FINDINGS AND MANAGERIAL IMPLICATIONS

We can predict the product relevance search score for various search queries based on mutual shared words of (search query, product title) and (search query, product description), edit distance (search query, product title) and (search query, product description) , search query length, number of words in description, Cosine Similarity of (search query, product title) and (search query, product description), Jaccard similarity of (search query, product title) and (search query, product description) with the help of Linear Regression, as this method provides the lowest RMSE.

Relevant product search term accuracy

- Can significantly boost revenues
- Positive effects on the user experience
- Increase CTRs i.e. click through rates
- Conversions of a customer lead to a sale
- Boosts customer satisfaction and retention rate
- A loyal customer can be a possible lead for cross sell and up sell of products

With the growing amount of information on the internet and with a significant rise in the number of users, it is becoming important for companies to search, map and provide them with the relevant chunk of information according to their preferences and tastes.

REFERENCES

- The dataset was taken from Kaggle.
  - Link:https://www.kaggle.com/c/home-depot-product-search-relevance/data
- Research Paper: Predict the Relevance of Search Results on Homedepot.com by Luyang Chen, Ruoxuan Xiong
  - Link: https://cs224d.stanford.edu/reports/ChenXiong.pdf
  - Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global Vectors for Word Representation. In EMNLP (Vol. 14, pp. 1532-1543)
- Research Paper: Measure Search Relevance for Home Depot Products, by Peng Xu
  - Link: http://billy-inn.github.io/papers/cmput690.pdf
  - Y. Bengio, H. Schwenk, J.-S. Sen´ecal, F. Morin, and J.-L. Gauvain. Neural probabilistic language models.Innovations in Machine Learning, pages 137–186, 2006
  - D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. the Journal of machine Learning research, 3:993–1022, 2003
  - N. Fuhr. Language models and uncertain inference in information retrieval. Proceedings of the Language Modeling and IR workshop, pages 6–11, 2001.
- Blogpost: Data Science Project Workflow by SHUAI'S AI & DATA BLOG
  - Link:https://shuaiw.github.io/2016/07/19/data-science-project-workflow.html
- Blogpost: Home Depot Kaggle: Feature Engineering Section
  - Link: https://nycdatascience.com/blog/student-works/home-depot-kaggle-feature-engineering-section/