Contents

PROJECT SPECIFICATION 2

Student provides a high-level overview of the project in layman's terms. Background information such as the problem domain, the project origin, and related data sets or input data is given. 2

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made. 4

Metrics used to measure performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem. 5

If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified. 5

A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined. 6

Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem. 8

Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained. 12

Data Preprocessing 13

All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified. 13

Creating features from data 14

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed. 17

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary. 17

The final model's qualities — such as parameters — are evaluated in detail. Some type of analysis is used to validate the robustness of the model's solution. 20

A visualization has been provided that emphasizes an important quality about the project with thorough discussion. Visual cues are clearly defined. 21

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult. 22

Discussion is made as to how one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution. 23

Project report follows a well-organized structure and would be readily understood by its intended audience. Each section is written in a clear, concise and specific manner. Few grammatical and spelling mistakes are present. All resources used to complete the project are cited and referenced. 23

Code is formatted neatly with comments that effectively explain complex implementations. Output produces similar results and solutions as to those discussed in the project. 23

PROJECT SPECIFICATION
Machine Learning Capstone Project

Definition

CRITERIA

MEETS SPECIFICATIONS

Project Overview

Student provides a high-level overview of the project in layman's terms. Background information such as the problem domain, the project origin, and related data sets or input data is given.
Answer:

This challenge is from Kaggle and below is the url for competition.

https://www.kaggle.com/c/home-depot-product-search-relevance/

In this competition, Home Depot is asking to help them improve their customers' shopping experience by developing a model that can accurately predict the relevance of search results. Search relevancy is an implicit measure Home Depot uses to gauge how quickly they can get customers to the right products. Currently, human raters evaluate the impact of potential changes to their search algorithms, which is a slow and subjective process. By removing or minimizing human input in search relevance evaluation, Home Depot hopes to increase the number of iterations their team can perform on the current search algorithms.

Problem Statement

Answer:

The problem is to come up with a model that can predict relevance of search results with almost same results as human raters.

For example based on the first example in train.csv as shown below, if human rater search on homedepot.com for search term "angle bracket" and if the search engine provides the below search term results of product title, product description and attributes they gave a relevance score of 3 thinking that search term and search results are perfect match . Below are the overall scores that human raters can give based on what they feel the relevance between search term and search results.

- Irrelevant (1)
- Partially or somewhat relevant (2)
- Perfect match (3)

We need to come up with a model that can predict the relevance of search results with search term and provide almost same results as human raters.

<u>Example of search term results for search term “angle bracket” which had a relevance score of 3 by human raters</u>

Search_term = “angle bracket”

<u>Search_term results</u>

Product_uid =” 100001”

product title = “Simpson Strong-Tie 12-Gauge Angle”

product description = “Not only do angles make joints stronger, they also provide more consistent, straight corners. Simpson Strong-Tie offers a wide variety of angles in various sizes and thicknesses to handle light-duty jobs or projects where a structural connection is needed. Some can be bent (skewed) to match the project. For outdoor projects or those where moisture is present, use our ZMAX zinc-coated connectors, which provide extra resistance against corrosion (look for a "Z" at the end of the model number).Versatile connector for various 90 connections and home repair projectsStronger than angled nailing or screw fastening aloneHelp ensure joints are consistently straight and strongDimensions: 3 in. x 3 in. x 1-1/2 in.Made from 12-Gauge steelGalvanized for extra corrosion resistanceInstall with 10d common nails or #9 x 1-1/2 in. Strong-Drive SD screws”

Attributes:

product_uid	Name	Value
100001	Bullet01	Versatile connector for various 90Â° connections and home repair projects
100001	Bullet02	Stronger than angled nailing or screw fastening alone
100001	Bullet03	Help ensure joints are consistently straight and strong
100001	Bullet04	Dimensions: 3 in. x 3 in. x 1-1/2 in.
100001	Bullet05	Made from 12-Gauge steel
100001	Bullet06	Galvanized for extra corrosion resistance
100001	Bullet07	Install with 10d common nails or #9 x 1-1/2 in. Strong-Drive SD screws
100001	Gauge	12
100001	Material	Galvanized Steel
100001	MFG Brand Name	Simpson Strong-Tie
100001	Number of Pieces	1
100001	Product Depth (in.)	1.5
100001	Product Height (in.)	3
100001	Product Weight (lb.)	0.26
100001	Product Width (in.)	3

The problem which needs to be solved is clearly defined. A strategy for solving the problem, including discussion of the expected solution, has been made.

Since the search term and search term results based on above example provided by homedepot doesn’t have predefined features as shown in below sample data, we need to come up with features based on combining the data from different files of train, product descriptions and attributes as shown in example below. Once the features are defined we try to use different machine learning algorithms and pick the one that predicts the output with low root mean square error.

Below is the sample data from different files provided by Home Depot.

<u>Train file sample data</u>

Train data first row

id	product_uid	product_title	search_term	relevance
2	100001	Simpson Strong-Tie 12-Gauge Angle	angle bracket	3

<u>Product _descriptions file sample data</u>

Product description row for product_uid 100001

product_uid	product_description
100001	Not only do angles make joints stronger, they also provide more consistent, straight corners. Simpson Strong-Tie offers a wide variety of angles in various sizes and thicknesses to handle light-duty jobs or projects where a structural connection is needed. Some can be bent (skewed) to match the project. For outdoor projects or those where moisture is present, use our ZMAX zinc-coated connectors, which provide extra resistance against corrosion (look for a "Z" at the end of the model number).Versatile connector for various 90 connections and home repair projectsStronger than angled nailing or screw fastening aloneHelp ensure joints are consistently straight and strongDimensions: 3 in. x 3 in. x 1-1/2 in.Made from 12-Gauge steelGalvanized for extra corrosion resistanceInstall with 10d common nails or #9 x 1-1/2 in. Strong-Drive SD screws

<u>Attribute file sample data</u>

Attribute row for product_uid 100001

product_uid	Name	Value
100001	Bullet01	Versatile connector for various 90Â° connections and home repair projects
100001	Bullet02	Stronger than angled nailing or screw fastening alone
100001	Bullet03	Help ensure joints are consistently straight and strong
100001	Bullet04	Dimensions: 3 in. x 3 in. x 1-1/2 in.
100001	Bullet05	Made from 12-Gauge steel
100001	Bullet06	Galvanized for extra corrosion resistance
100001	Bullet07	Install with 10d common nails or #9 x 1-1/2 in. Strong-Drive SD screws
100001	Gauge	12

100001	Material	Galvanized Steel
100001	MFG Brand Name	Simpson Strong-Tie
100001	Number of Pieces	1
100001	Product Depth (in.)	1.5
100001	Product Height (in.)	3
100001	Product Weight (lb.)	0.26
100001	Product Width (in.)	3

Since Home Depot in kaggle competition didn't provide features and label, we need to come up with features as described in section Data Preprocessing and Creating features from data and once we have the features and label try different algorithms to come up with an algorithm mentioned from Algorithms and Techniques and come up with an algorithm and parameters for that algorithm which can predict relevance with less root mean square error.

Metrics

Metrics used to measure performance of a model or result are clearly defined. Metrics are justified based on the characteristics of the problem.

Answer:

Metrics that is already requested by Home Depot is to use root mean square error. I think that is reasonable because it is a regression problem and also it penalizes the relevance values that are outliers compared to actual values.

Analysis

CRITERIA

MEETS SPECIFICATIONS

Data Exploration

If a dataset is present, features and calculated statistics relevant to the problem have been reported and discussed, along with a sampling of the data. In lieu of a dataset, a thorough description of the input space or input data has been made. Abnormalities or characteristics about the data or input that need to be addressed have been identified.

Answer:

As mentioned above since Home Depot in kaggle competition didn't provide features and label, we need to come up with features as described in section Data Preprocessing and Creating features from data.

Below is sample of features and label that I came up with in the attached document samplefeaturesandlables.xlsx. The label is relevance which is in yellow color. Apart from the label the remaining columns are features.

There are some abnormalities while creating features for example brand name is not present for all products and in that case cosine similarity between search term and brand name and other feature creation techniques are showing zero values since brand name is not there and impacting the predicting label as these values are zero, so I concatenated product title to brandname feature so that for example even if brand name is not present but if product title matches the search term, features are not zero and had appropriated prediction.

Samplefeatureandlab
els.xlsx

Below results can be find in the Statistics section of the python notebook

Train data statistics

Total number of product_uids in train data 74067
Total number of unique product_uids in train data 54667
Total number of search_terms in train data 74067
Total number of unique search_terms in train data 11795
Total number of unique relevance scores in train data 13

Test data statistics

Total number of product_uids in test data 166693
Total number of unique product_uids in test data 97460
Total number of search_terms in test data 166693
Total number of unique search_terms in test data 22427

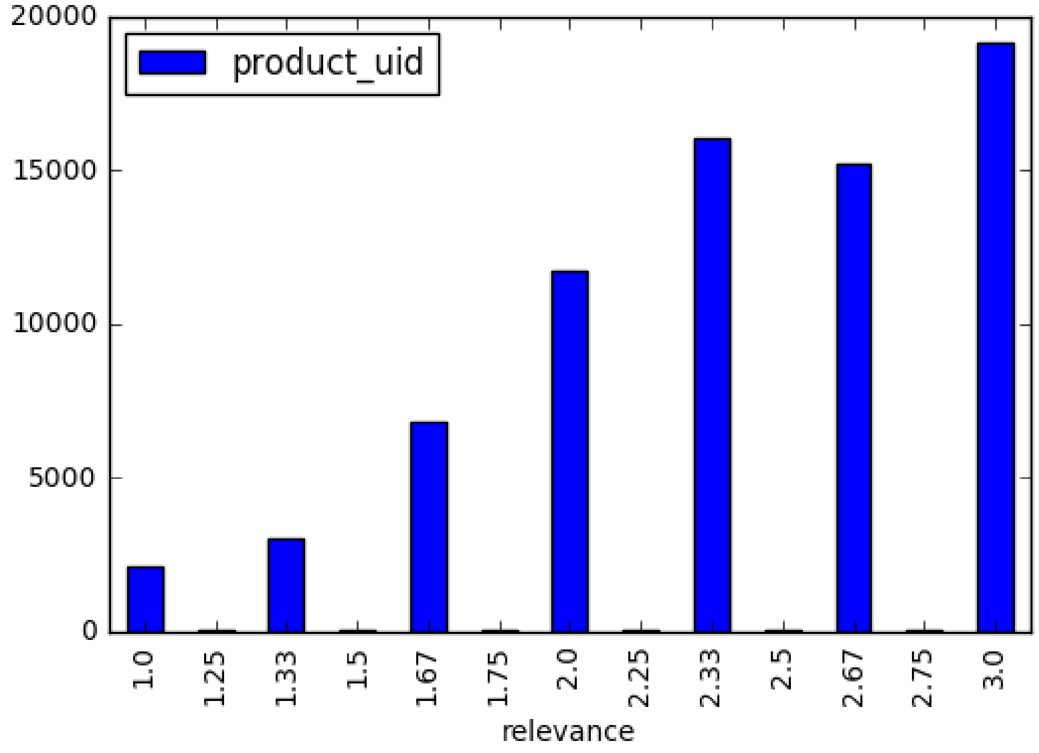
Similarity between Train and test data

Number of productid in train but not in test 26968
Number of productid in test but not in train 69761
Number of search_term in train but not in test 2174
Number of search_term in test but not in train 12806

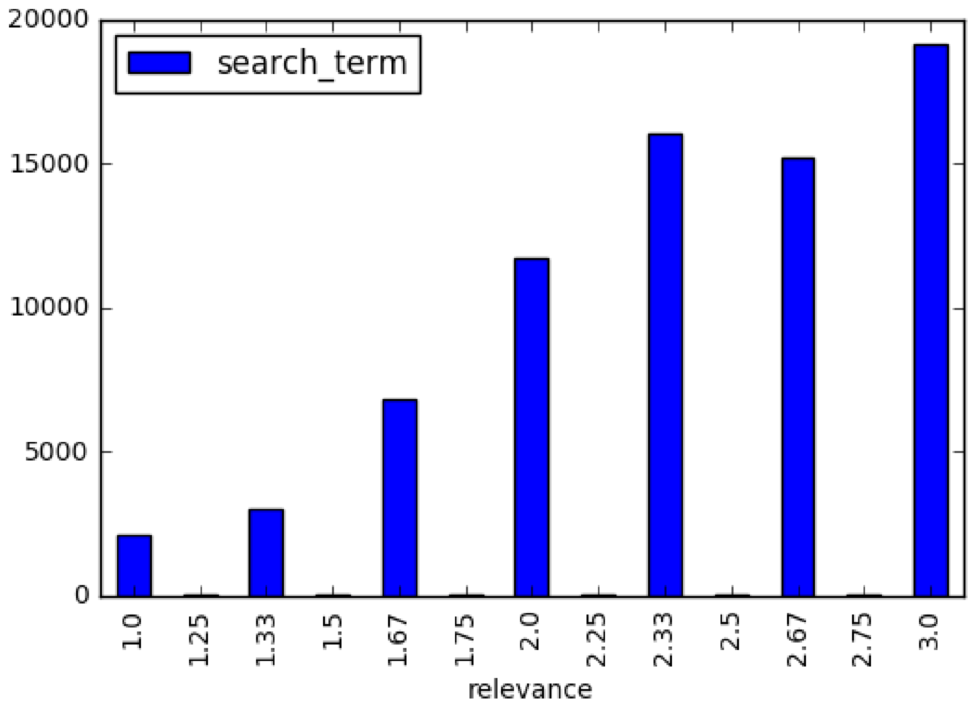
Exploratory Visualization

A visualization has been provided that summarizes or extracts a relevant characteristic or feature about the dataset or input data with thorough discussion. Visual cues are clearly defined.

Below is the graph of relevance scores (label) and number of product_ids that had those relevant scores. This shows most of the product_ids in train data has a relevance score of greater than are equal to 2.



Below is the graph of relevant (label) scores and number of search_terms that had those relevant scores. The above graph is same as below graph because product_id indirectly represents search_term. This shows most of the search_term in train data has a relevance score of greater than are equal to 2.



Algorithms and Techniques

Algorithms and techniques used in the project are thoroughly discussed and properly justified based on the characteristics of the problem.

Answer:

I tried the below regression algorithms for this problem as this is a regression problem.

1) Linear regression

Linear regression methods attempt to solve the regression problem by making the assumption that the dependent variable is (at least to some approximation) a linear function of the independent variables, which is the same as saying that we can estimate y using the formula:

y = c0 + c1 x1 + c2 x2 + c3 x3 + … + cn xn

where c0, c1, c2, …, cn. are some constants (i.e. fixed numbers, also known as coefficients, that must be determined by the regression algorithm). The goal of linear regression methods is to find the “best” choices of values for the constants c0, c1, c2, …, cn to make the formula as “accurate” as possible by reducing the root mean square error between prediction and actual value. The best constants can be found out by gradient descent algorithm.

Pros:

1. Its implementation on modern computers is efficient, so it can be very quickly applied even to problems with hundreds of features and tens of thousands of data points

2. It is easier to analyze mathematically than many other regression techniques

3. It is not too difficult for non-mathematicians to understand at a basic level.

4. It is easy to implement on a computer using commonly available algorithms from linear algebra.

Cons:

1) Outliers

Linear regression can perform very badly when some points in the training data have excessively large or small values for the dependent variable compared to the rest of the training data. The reason for this is that since the least squares method is concerned with minimizing the sum of the squared error, any training point that has a dependent value that differs a lot from the rest of the data will have a disproportionately large effect on the resulting constants that are being solved for.

2) Non-Linearities

All linear regression methods suffer from the major drawback that in reality most systems are not linear.

3) Dependence Among Variables

The linear regression model can sometimes lead to poor predictions when a subset of the independent variables fed to it are significantly correlated to each other.

4) Too Many Variables

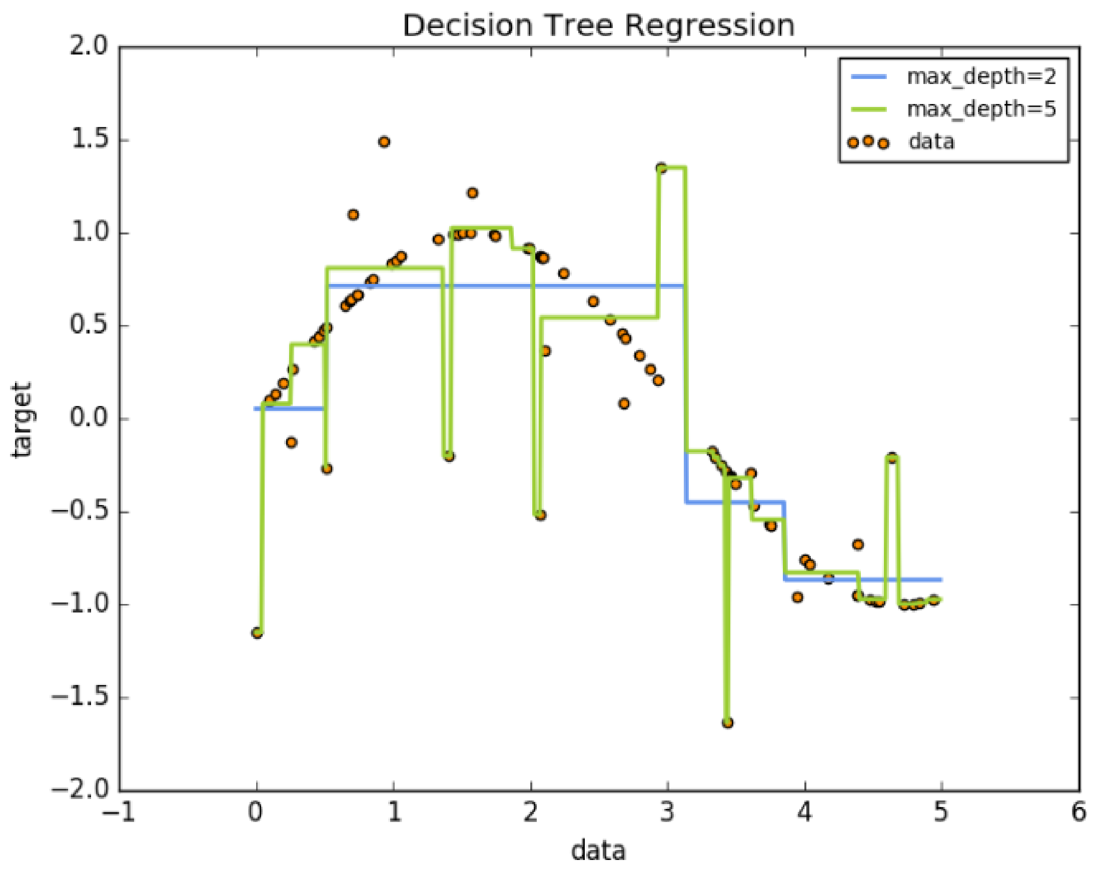
As soon as the number of features used exceeds the number of training data points, the least squares solution will not be unique, and hence the linear regression algorithm will fail.

Reference: http://www.clockbackward.com/2009/06/18/ordinary-least-squares-linear-regression-flaws-problems-and-pitfalls/

2) Decision tree

Decision Trees (DTs) are a non-parametric supervised learning method used for regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

For instance, in the example below, decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.



Pros:

1. Simple to understand and to interpret. Trees can be visualised.
2. Requires little data preparation. Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.
3. The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
4. Able to handle both numerical and categorical data.
5. Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.

Cons:

1. Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting.

2. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This problem is mitigated by using decision trees within an ensemble.
3. The problem of learning an optimal decision tree is known to be NP-complete under several aspects of optimality and even for simple concepts. Consequently, practical decision-tree learning algorithms are based on heuristic algorithms such as the greedy algorithm where locally optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees in an ensemble learner, where the features and samples are randomly sampled with replacement.
4. There are concepts that are hard to learn because decision trees do not express them easily, such as XOR, parity or multiplexer problems.
5. Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Reference:

http://scikit-learn.org/stable/modules/tree.html#tree

3) Random forest regressor.

A random forest regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. Random forest is used in the applications where decision trees are used to control over fitting.

Pros

1) Random forest ensembles a number of decision tree classifiers on random sub-samples of the dataset and random features and use averaging to improve the predictive accuracy and control over-fitting issue caused by decision trees.

Cons:

1) The main limitation of the Random Forests algorithm is that a large number of trees may make the algorithm slow for real-time training.

Reference:
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

4) Gradient Boosting regressor

Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems. Gradient Tree Boosting models are used in a variety of areas including Web search ranking and ecology.

Pros:

1. Natural handling of data of mixed type (= heterogeneous features)
2. Predictive power
3. Robustness to outliers in output space (via robust loss functions)

Cons:

1. Scalability, due to the sequential nature of boosting it can hardly be parallelized.

Reference:
http://scikit-learn.org/stable/modules/ensemble.html#forest

Benchmark

Student clearly defines a benchmark result or threshold for comparing performances of solutions obtained.
Answer:

Homedepot provided a benchmark script as shown below with private score of 0.51064 and public score of 0.51147. Goal is to beat this.

Below benchmark was created using linear regression and using the below features

1) Count of words in search_term matching the product title
2) Count of words in search_term matching the product description

https://www.kaggle.com/dsoreo/home-depot-product-search-relevance/benchmark-score-script

Methodology

CRITERIA

MEETS SPECIFICATIONS

Data Preprocessing

All preprocessing steps have been clearly documented. Abnormalities or characteristics about the data or input that needed to be addressed have been corrected. If no data preprocessing is necessary, it has been clearly justified.

Answer:

Below are the steps for Data preprocessing.

1) Spell check correction

   For spell check correction I am using the below dictionaries shared by users as part of disclosure of external data used. Apply that function on search_term since users can enter typos in search term.

   I am using the spell check dictionary provided as part of external data disclosure so that every one can use it for fair purpose.

   Manualcheckdict

   https://www.kaggle.com/steubk/home-depot-product-search-relevance/fixing-typos

   spelling_correction

   https://www.kaggle.com/c/home-depot-product-search-relevance/forums/t/19413/external-data-disclosure-thread/115404

   another_replacement_dict and external_data_dict from below link

   https://www.kaggle.com/buinyi/home-depot-product-search-relevance/disclosing-external-data

   https://www.kaggle.com/the1owl/home-depot-product-search-relevance/rfr-features-0-47326/code

2) HomeDepot has particular data based on train and test data for example

   1x1 rail decorative wood

   4*8 beadboard paneling

   4x8wood paneling

   Created functions to remove above x and * and convert number to its appropriate value so that if product title is 4*8 beadboard and if search term is four by eight, if we don't do the data processing it might not be able to find the match.

3) Also some of the titles etc in homedepot has inches or feet values to make it generic converted in. to inch etc so that if people search for inch it can be able to find the match.
4) Since this project has three different files train data, attributes and product description file and all of those files have important information in identifying search relevance, I combined all the files using productid and created multiple features so that we can compare search term and those features.
5) Remove stop words like and , the etc
6) Changed all the words to lower case
7) Remove non letters which are not [0-9a-zA-Z]
8) Create stemmer words for example stem(angle) is angl

9) Separate word from number and replace value of number to appropriate English value. For example concret wire five x100 to concret wire five x hundred

Implementation

Creating features from data

Answer:

Since this project doesn't have any predefined features, coming up with features is an interesting task as shown below.

The features I came up with are

1) Creating features with cosine Similarity between search term and other features like product description, product title, brandname, attributes and metal.
   When comparing two documents we use TF-IDF to get the vectors for both the documents and then we use the cosine similarity between those two vectors to determine how close the two documents are.
   For example
   Document 1 : Angle Bracket
   Document 2 : Angle loop Angle

   You calculate TFIDF by multiplying TF and IDF values for each document.
   TF = Number of times the word appears/ Total number of words in document
   (Rewards words that appear many times in same document)
   TF value of Angle in document 1 = ½ = 0.5
   TF value of Bracket in document 1 = ½ = 0.5

   IDF = Total number of documents/ Number of documents containing the word
   IDF rewards words that are rare overall in the dataset.
   Assuming only two documents exist,
   IDF of Angle in document 1 = 2/2 = 1
   IDF of Bracket in document 1 = 2/1 = 2

   TFIDF for a word = Product of the word TF and IDF values for each document
   TFIDF of Angle in document 1 = 1 * 0.5 = 0.5
   TFIDF of Bracket in document 1 = 2 * 0.5 = 1

   Very important is to remove stop words (a, the, on etc) from the documents as this can pollute the results.

   Now the TFIDF values for document1 are below which represent a 2 dimensional vector =

Angle	Bracket
0.5	1

Similarity we can calculate TFIDF values for different words in document 2 which will be another vector.

Now comparing if two documents are similar are computing cosine of angle between two vectors

Small angle (larger cosine value) means documents share many words in common.
Large angle (smaller cosine value) means documents share few words in common.

Using sklearn TFIDFvectorizer and cosine similarity given in links below achieved the above cosine_similarity values between search term and product title etc.

http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

2) Creating feature that contains total number of Bullets count per product id in product_descriptions file.
This is just for one product_uid in product_descriptions file and the total number of bullets count that start with Bullet is 7. Creating a feature column with the Bullet count.

product_uid	Name	Value
100001	Bullet01	Versatile connector for various 90Â° connections and home repair projects
100001	Bullet02	Stronger than angled nailing or screw fastening alone
100001	Bullet03	Help ensure joints are consistently straight and strong
100001	Bullet04	Dimensions: 3 in. x 3 in. x 1-1/2 in.
100001	Bullet05	Made from 12-Gauge steel
100001	Bullet06	Galvanized for extra corrosion resistance
100001	Bullet07	Install with 10d common nails or #9 x 1-1/2 in. Strong-Drive SD screws
100001	Gauge	12
100001	Material	Galvanized Steel
100001	MFG Brand Name	Simpson Strong-Tie
100001	Number of Pieces	1
100001	Product Depth (in.)	1.5
100001	Product Height (in.)	3
100001	Product Weight (lb.)	0.26
100001	Product Width (in.)	3

3) Adding fuzz ratio features

Implemented below functions to check the string matching between search term and product title, product_description, attributes and brandname.

- fuzz.token_sort_ratio
- fuzz.token_set_ratio
- fuzz.partial_ratio
- fuzz.ratio

Fuzzy String Matching, also called Approximate String Matching, is the process of finding strings that approximatively match a given pattern.
The closeness of a match is often measured in terms of edit distance, which is the number of primitive operations necessary to convert the string into an exact match.

Below is the source of this functions.
https://marcobonzanini.com/2015/02/25/fuzzy-string-matching-in-python/

4) Feature column with values populate from function to find how many words in search term are present in the column that is being compared to.
For example search term = ‘angl bracket bra brac cket ket’
And product_title = 'simpson strong tie twelv gaug angl bracket';
Since all the search term words are present in product title it will give a total count of 6.

5) Feature creation using models.word2vec as shown in the below link which compares cosine similarity between two sets of words based on the vectors defined by the model.
https://radimrehurek.com/gensim/models/word2vec.html

n_similarity(*ws1*, *ws2*)
Compute cosine similarity between two sets of words.
Example:

```
>>> trained_model.n_similarity(['sushi', 'shop'], ['japanese', 'restaurant'])
0.61540466561049689
>>> trained_model.n_similarity(['restaurant', 'japanese'], ['japanese', 'restaurant']
1.0000000000000004
trained_model.n_similarity(['sushi'], ['restaurant']) == trained_model.similarity('sushi',
'restaurant'
True
```

6) Features created by function to determine percentage of search term present in comparison term.
If search term = Angle Bracket and Product title = Angle Connection tool
Percentage ratio of comparison between search term and product title = 1 (Angle is present in both terms)/2 (length of search term) = 0.5
7) Drop unnecessary features like product title etc to train the feature columns for machine learning algorithm.

The process for which metrics, algorithms, and techniques were implemented with the given datasets or input data has been thoroughly documented. Complications that occurred during the coding process are discussed.
Answer:

The algorithms used for the testing are

1) Linear regression
2) Decision tree
3) Random forest regressor with cross validation
4) Gradient Boost regressor with cross validation

The complications that occurred during the coding process are

1) Understanding word2vec and cosine similarity and coming up with features.
2) Integrating data from multiple files based on product_id.
3) Also when the program loads the word2vec model it needs more RAM, so need a powerful machine to run this job.
4) Based on amount of data cross validation takes more time and using n=-1 jobs uses all cores available on machine will speed up the process.

Refinement

The process of improving upon the algorithms and techniques used is clearly documented. Both the initial and final solutions are reported, along with intermediate solutions, if necessary.
Answer:

After combining all the different data (train, test, attributes, product description) given by Home Depot into a table, I performed data preprocessing and spell check correction. After this worked on below features and calculated test rmse score and rank.

1) Created cosine features and calculated the test rmse score and rank
2) Added fuzzy ratio feaures along with cosine features and calculated the test rmse score and rank
3) Added search term frequency ratios and word2vec similarity features and calculated the test rmse score and rank
4) Added material features with above cosine features, fuzzy ratio features, search term frequency ratios and calculated the test rmse score and rank

Results

Below is the table with all comparisons and how the test rmse score and rank progressed as I kept on coming up with new features and with different alogithms.

Model	**Tuned Prameters**	**Score**	**Rank**
Benchmark		0.5105	1681
Cosine features with linear regression		0.49293	1456
Cosine features with decision tree		0.49865	1528

Cosine features with random forest tuning	{'max_depth': 14, 'max_features': 'auto', 'min_samples_split': 250, 'n_estimators': 300}	0.48899	1377
Cosine features and fuzzy wuzzy with linear regression		0.48152	990
Cosine features and fuzzy wuzzy with decision tree		0.48571	1142
Cosine features and fuzzy wuzzy with Random forest tuning	{'max_depth': 10, 'max_features': 'log2', 'min_samples_split': 500, 'n_estimators': 200}	0.4699	342
Linear Regression with all features including material		0.47901	932
Decision tree with all features including material		0.48461	1091
Random forest with more parameters to tune with all features excluding material features		0.4699	342
Gradient descent with more parameters to tune with all features excluding material features		0.4691	313
Random forest with less parameters to tune but with wordvecsim with all features excluding material features		0.46954	330
Gradient descent with less parameters to tune but with wordvecsim with all features excluding material features		0.46932	321
Random forest with more parameters to tune with all features including material features	{'max_depth': 14, 'max_features': 'auto', 'min_samples_split': 250, 'n_estimators': 400}	0.46858	297
Gradient descent with more parameters to tune and with all features including material features	{'max_depth': 10, 'max_features': 'log2', 'min_samples_split': 500, 'n_estimators': 200}	0.46751	264

As you can see in the table below

Result Analysis Table

Feature Set	Linear regression RMSE score	Linear Regression Ranking	Decision tree RMSE Score	Decision tree Ranking	Random forest Regressor RMSE Score	Random forest Regressor Ranking	Gradient Descent RMSE score	Gradient Descent Ranking
Cosine features	0.49293	1456	0.49865	1528	0.48899	1377		
Cosine features and fuzzy wuzzy	0.48152	990	0.48571	1142	0.4699	342		
All features excluding material features					0.4699	342	0.4691	313
All features excluding material features but with wordvecsim					0.46954	330	0.46932	321
All features	0.47901	932	0.48461	1091	0.46858	297	0.46751	264

Linear regression RMSE score and rank was somewhat better compared to decision tree RMSE score and rank as we kept on adding more features. But since all the features and label are not correlated the RMSE score and rank didn't perform that great.

Decision tree RMSE score and rank didn't improve much as we kept on adding more features because of overfitting to train data but didn't perform well with test data.

Since Random forest regressor ensembles a number of decision trees on random sub-samples of the dataset and random features and use averaging to improve the predictive accuracy and control over-fitting issue caused by decision trees, we can see the Random forest score and rank was lot better than Decision trees.

Since Gradient boosting regressor also ensembles a number of decision trees the score and rank was lot better than Decision trees and was little bit better than random forest regressor.

For random forest and gradient descent I tuned with three fold cross validation and with gridsearchcv with below parameters and picked the best parameter with low rmse score.

n_estimators : The number of trees in the random forest or gradient boosting

max_depth : The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : The minimum number of samples required to split an internal node.

max_features :The number of features to consider when looking for the best split.

Reference: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

Tuned parameters:

1. 'n_estimators': [200, 300, 400],
2. 'max_depth': [10, 12, 14],
3. 'min_samples_split' : [250,500,750],
4. 'max_features' : ['auto','sqrt','log2']}

Based on the above gridsearchcv process for Gradient boosting algorithm the best parameters with best rmse score and ranks as per above table is

```
{'max_depth': 10,
 'max_features': 'log2',
 'min_samples_split': 500,
 'n_estimators': 200}
```

CRITERIA

MEETS SPECIFICATIONS

Model Evaluation and Validation

The final model’s qualities — such as parameters — are evaluated in detail. Some type of analysis is used to validate the robustness of the model’s solution.

As mentioned in above section based on grid search with tunable parameters and with cross validation of 3, the gradient boost regressor algorithm choose the best model parameters as

```
{'max_depth': 10,

 'max_features': 'log2',

 'min_samples_split': 500,

 'n_estimators': 200}
```

and provided a test rmse score of 0.4671 and rank of 264.

Justification

As you can see in the Result Analysis Table decision tree was overfitting compared to random forest and gradient descent even with more features being added. Linear regression performed well compared to decision tree but random forest and gradient descent were better than Linear regression. Gradient descent and Random forest both fared well since they are ensemble of number of trees but Gradient descent score and rank was little better than Random forest.

The final results of Gradient descent are compared to the benchmark result or threshold with some type of statistical analysis. Justification is made as to whether the final model and solution is significant enough to have adequately solved the problem.
Answer:

The final results 0.4671 with a rank of 264 has a better score compared to benchmark result of 0.5105 and rank of 1681. I think the final model is significant enough to have adequately solved the problem as rmse difference based on the rank of 264 and 1st rank of 0.43192 is pretty less.

Conclusion

CRITERIA

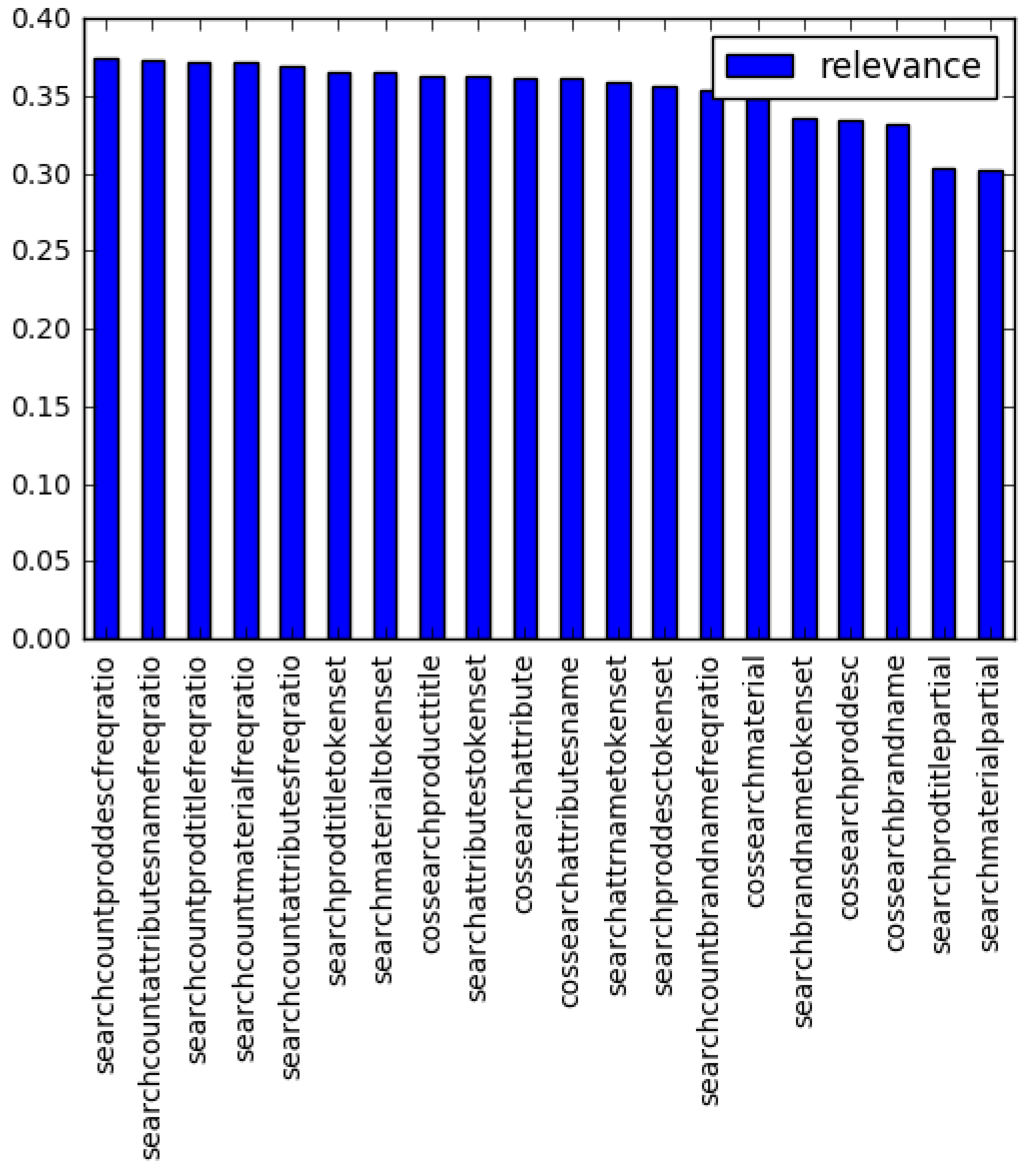
MEETS SPECIFICATIONS

Free-Form Visualization

A visualization has been provided that emphasizes an important quality about the project with thorough discussion. Visual cues are clearly defined.

Below are the top 20 features and their correlation with relevance features in descending order. Based on this the below feature creation methods between search term and other columns like product description, product title etc are more important feature creation techniques for this project.

1) Frequency ratios
   Feature create by frequency ratio of search term and other columns.
   Ex: If search term = Angle Bracket and Product title = Angle Connection tool
   Frequency ratio between search term and product title = (Count of words in search that are present in product title) / (count of words in search term)
   In the above example since Angle in search term is present in product title , frequency ratio = 1/ 2 (count of words in search term) = 0.5
2) Cosine search
   Features created by cosine similarity between search term and other columns as explained in section Creating features from data
3) Tokenset
   Features created by fuzztokensetratio between search term and other columns as explained in section Creating features from data
4) Partial
   Features created by fuzzpartialratio between search term and other columns as explained in section Creating features from data



Reflection

Student adequately summarizes the end-to-end problem solution and discusses one or two particular aspects of the project they found interesting or difficult.
Answer:

The problem is to come up with a model that can predict relevance of search results with almost same results as human raters. I came up with features as mentioned in above document and was able to get a test rmse score of 0.4671 and rank of 264. The two things I found interesting in this project are

1) Finding similarity of two documents with cosine angle and TFIDF since I never thought the documents can be represented as vector space model and the similarity can be measured using cosine angle between the documents and this is one of the important process of NLP (natural language processing)
2) Finding similarity of two documents using word2vec since we are using google news corpus for this operation. This was an interesting topic when I researched for this project. **word2vec** tool takes a text corpus (GoogleNews-vectors-negative300.bin.gz in our project) as input and produces the word vectors as output and can find the similarity using model.n_similarity.

Improvement

Discussion is made as to how one aspect of the implementation could be improved. Potential solutions resulting from these improvements are considered and compared/contrasted to the current solution.
Answer:

One thing that can be further improved upon is come up with additional features using glove word embeddings as shown in below link.

http://nlp.stanford.edu/projects/glove/

This will help in adding additional features to find similariy between search document and other documents and will help reduce the test rmse score further and also the ranking.

Quality

CRITERIA

MEETS SPECIFICATIONS

Presentation

Project report follows a well-organized structure and would be readily understood by its intended audience. Each section is written in a clear, concise and specific manner. Few grammatical and spelling mistakes are present. All resources used to complete the project are cited and referenced.

Functionality

Code is formatted neatly with comments that effectively explain complex implementations. Output produces similar results and solutions as to those discussed in the project.