Home Depot Product Search Relevance

Rebecca Correia, Salil Kanetkar

Department of Computer Science
University of California, Los Angeles

March 10, 2016

Overview

Problem Definition

- Given a product and the search query which resulted in the product, develop a model that can accurately predict the relevance of search result.

Product Title	Search Query	Relevance
ClosetMaid ShelfTrack 4-Drawer Kit	closet baskets	???
Linzer 3 in. Chip Brush	throwaway chip brush	???

- Data science competition hosted on **Kaggle** by **Home Depot**

Dataset

- Provided by the hosts of the competition
- Dataset consisted of $\sim$ 74K observations
- Two variables : *Product Title* and *Search Query*
- Outcome: *Relevance Score*
- Outcome is the average of three human raters
- Split this data into train (70%) and test (30%)
- Other datasets, *Product Description* and *Product Attributes* also provided

Data Pre-processing (1)

- Appended product description to the dataset based on the *Product ID*
- Extracted brand information from the attributes dataset and appended to the main dataset
- Eliminated observations that had missing values i.e. that contained NA, na or unbranded

Data Pre-processing (2)

- Converted all words to lowercase
- Performed stemming using Snowball Stemmer
- Removed stop words such as the, is, which, on , etc.
- Eliminated punctuations

Feature Engineering - Text to numbers

- According to us, what features we give to the model is a very important step.
- Can't pass textual data to a model, need to convert it into numbers
- *Textual Data −> DTM −> TF-IDF Matrix −> Dimensionality Reduction using SVD*
- *TF-IDF*: Term Frequency - Inverse Document Frequency

$$TF - IDF = \{f_{t,d}\}.\{log \frac{N}{|\{d \in D : t \in d\}|}\}$$

- Used Singular Value Decomposition (SVD) to reduce the dimensions of the data
- Thus, textual data converted into numbers. But that's not enough!

Feature Engineering -Extracted Features

- Length of query
- Length of title
- Length of description
- Length of brand
- # query in title and description
- 0/1 depending on whether last word of query is present in title and description
- # words of query in title and description
- # words of query in brand name
- $\frac{\#\text{words of query in title/description}}{\text{len(query)}}$
- $\frac{\#\text{words of query in brand name}}{\text{len(query)}}$
- Brand id
- # word of search term in product info

Feature Engineering - Feature Sets (FS)

- Feature Set 1: Extracted Features + TF-IDF (Search) + TF-IDF (Product Title + Description)
- Feature Set 2: Extracted Features + TF-IDF (Product Info)
- Feature Set 3: Extracted Features + TF-IDF (Search) + TF-IDF (Product Title + Description) + Cosine Similarity
- Feature Set 4: Extracted Features + Cosine Similarity
- Feature Set 5: Extracted Features

Model Training

1. Support Vector Regression (SVM)- Linear Kernel
2. Support Vector Regression (SVM)- Radial Kernel
3. Neural Network (NN) Regression
4. Random Forest (RF) Regression
5. XGBoost: Gradient Boosting + Regularization

Results - RMSE Values

Algorithm\Feature Set	FS1	FS2	FS3	FS4	FS5
SVR Linear	0.495	0.497	0.496	0.496	0.497
SVR Radial	0.484	0.497	0.488	0.491	0.494
Random Forest	0.466	0.48	0.145	0.126	0.494
Neural Network	0.479	0.485	0.479	0.487	0.486
XGBoost	0.312	0.319	0.129	0.114	0.320

Analysis and Conclusion

- Machine learning is not just about training models, feature selection is also important
- Important to perform dimensionality reduction when working with textual data
- For us, XGBoost turned out to be a rockstar
- Incorporating the feature of cosine similarity seems to have been helpful to model the relationship between two strings
- It would be interesting to see the performance with respect to other similarity indices like Jaccard etc.

Thank You! Questions?