Stanley Black & Decker – Emory MSBA

Capstone Final Project Report

Stella Guo, Mayur Kumar, Anuraaj Sonawala, Robert Wei

Business Understanding

Recognizing customers' pain point or improving an existing product according to market feedbacks is always a challenging problem for manufacturers. How does a company decide to improve products for such an undertaking that it stands a good chance of being successful in the marketplace? From a company perspective, success means introducing products while maintaining a healthy profit margin. While from a consumer perspective, success is understanding their product feature expectations.

Customers online feedbacks can be a direct and valuable reflection on products and the features. And nowadays, we can find platforms and tools that help transform these comments into quantitative results with Natural Language Processing methods. Various customer experience software (e.g. InMoment, Clarabridge) collect feedback from numerous sources, alert on mentions in real-time, analyze text, and visualize results. Text analysis platforms (e.g. DiscoverText, IBM Watson Natural Language Understanding, Google Cloud Natural Language, or Microsoft Text Analytics API) have sentiment analysis in their feature set. However, commercial software may be less accurate when analyzing texts from specific domains and industries. For example, there can be negative words aren't negative if used in particular contexts. For these cases, our team come to cooperate and develop a solution that fits our industry.

So, as for the business understanding, we want to know all the customer review intelligence on competitor's products and SBD products across different retailers, for product developers to understand pain points and sentiments on features, on both category levels and product levels, in order to make decisions on new product introduction or product improvement.

Data Preparation

Every entry of the data consists of customers' feedback and the other information about the product and the purchase. There are four online retailers we collect data from: Amazon, the Home Depot, Walmart and Lowes. There are all kinds of brands in this industry including SBD's brands like DeWalt and Black+ Decker; and there are also competitor brands like Lufkin and Milwaukee. We also collect information of product categories like Power Drills, Tape Measure, Reciprocating Saws, Impact Wrenches and for every category there are all kinds of different products. And for every product, we focus on their typical features.

To prepare the data, we first combined datasets from web scraping, removed duplications, filtered out unreasonable information.

Secondly, we extracted retailer information. We could find url information for every entry like this "https://www.homedepot.com/.....", so we could tell what the retailer is and group every entry.

After that, we found there can be category discrepancies of the same product from different retailers, so we synthesized categories for the same model from different retailers. For example, there are 95 Model 0882-20 for category of Specialty Power Tools and 5 for category of Stick Vacuums, and we take the majority category.

Thirdly, we Used SBD's competitor list to focus on main competitors. There are 10 SBD brands include STANLEY, Stanley FatMax, DEWALT, Black & Decker, Craftsman, IRWIN, Porter Cable, LENOX, BOSTITCH, PROTO; and The other brands are from competitors.

To narrow down the scope and make it feasible, we only focus on tool categories Aggregated reviews to remove categories with less than 100 reviews. So, we Joined category-level reviews and conduct text-processing on Top reviewed categories and as a result, we got a dataset consisting of 102648 comments, 396 categories, 7747 products and 48 brands.

We also collect sale data as a reference. It's a dataset July to September in 2019, not suitable for long-term prediction. We aggregate by product on four time periods on daily level, calculated daily sale and daily revenue (price*sale_unit), visualized in line chart and apply time series forecasting on sale.

Modelling

Text Processing

- Lowercase all the letters in the reviews.
    - Otherwise, for example: Right and right are considered as different for LDA model. This doesn't matter for computing sentiments, but it matters when we extract features
    - Used tolower() function
- Stemming/ Lemmetization:
    - Example: break is the root word for broke, broken, be broken, been breaking, breaks, will break, will be breaking, did break and many more. They all carry the same meaning. We don't want them to be treated as different, so let's stem all these words
    - Used
- Removing Stop words
    - Is/am/are/was/were/be and many such words are removed
    - Used standard dictionary for this
- Remove short words
    - Words of length less than 4 typically wouldn't become a feature. So, let's remove them to avoid being picked up as a feature by the model
- Remove Long words

- Words of length more than 12 typically wouldn't become a feature. So, let's remove them to avoid being picked up as a feature by the model

- Remove unnecessary punctuations.
  - We don't want unnecessary punctuations such as "-" "," "!" "_" and many more to differentiate the same word as different words
- Named Entity Recognition:
  - We are telling model to not pick some standard and commonly recognized words such as locations, organizations, and names.
  - Used standard nltk dictionary
- Postags:
  - Tag the words with their parts of speech. Thereby, removing the verbs, adjectives from the data.

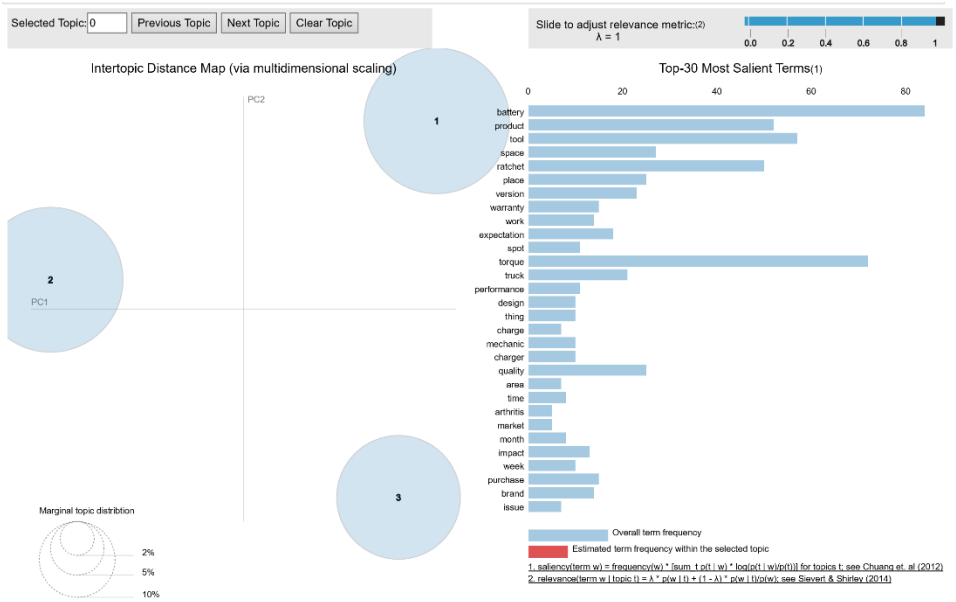
Modeling (Read the code manual at the bottom before reading this section)

We tried 3 Models. LDA ,LSA ,and PLSA. Since LDA is our best model. I will be demonstrating LDA model.

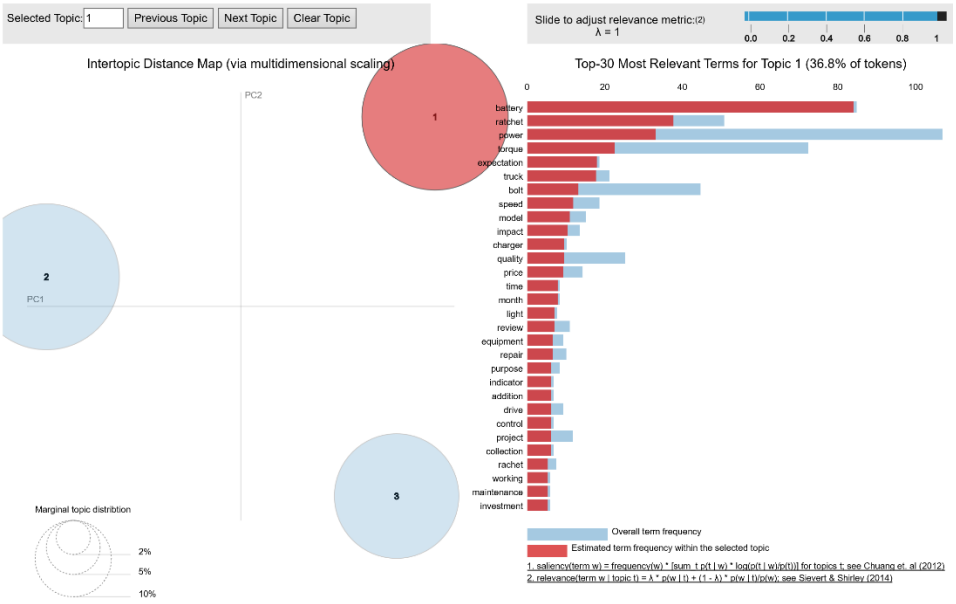
- Python code is self explanatory. After running the LDA model on your selected category. It spits output in a very good visualization format. Screenshots are attached in theoutput section.

<u>How to interpret LDA output:</u>

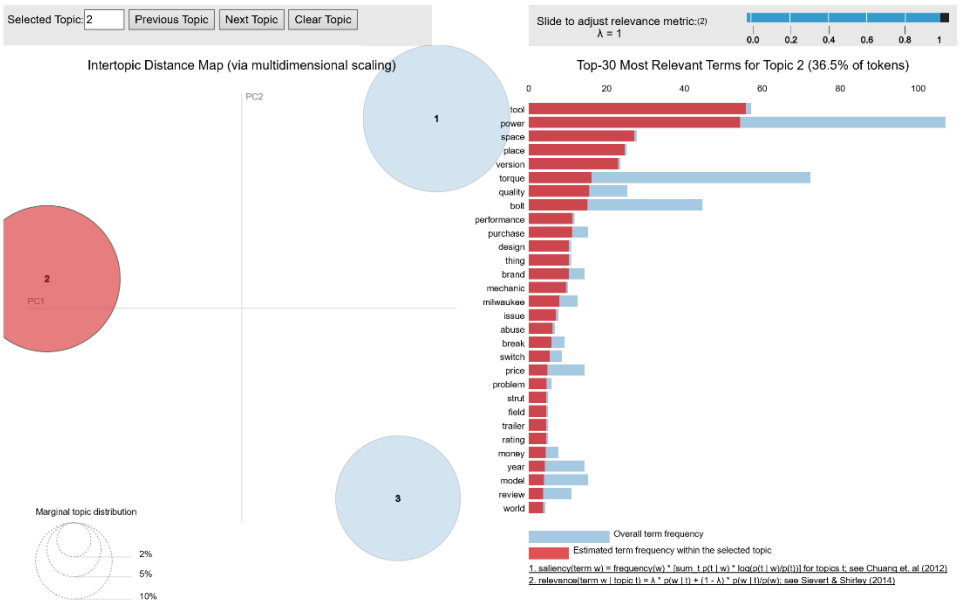
- Each Circle is a topic. Topic means set of words. Screenshot shows the output when no topic is selected.



- Circle_ 1 represents topic_1 ,and the set of words for topic_1 are seen in the screenshot. They are in the descending order of their importance.



- Topic_2 has it's own set of words.

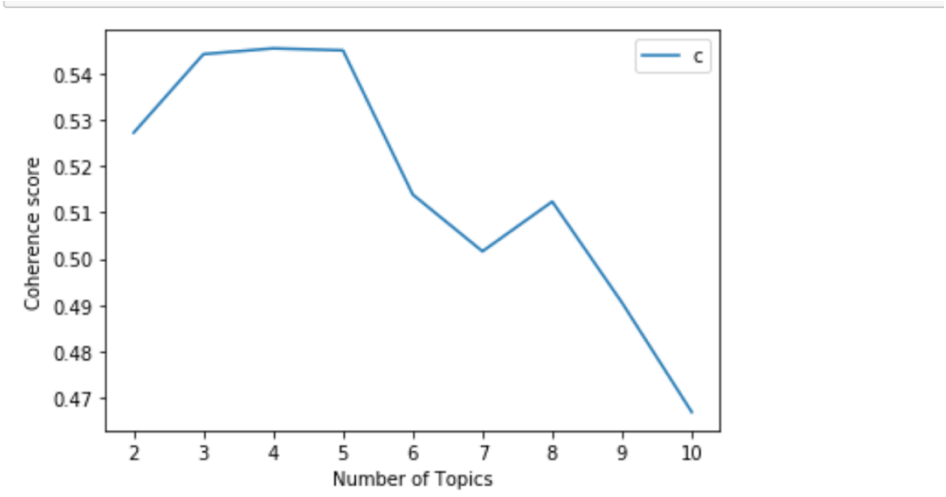


- Intersection between two circles there are some overlapping the set of words. Very far from each other means, those topics are very distinct.

How many Topics to select for each category?

LDA model has it's own performance metric which is called "coherence score". High score means the topics are very coherent and distinct. The code automatically tries out various number of topics and sees

when the highest coherence score is ocurring. For example. If you look at the following screenshot, at number of topics=3, coherence score is high. So we saw only circles earlier.



How to choose features from these words?

Method1(Manual)

Choose manually some 30 features or upto you. Look at each topic and pick what all makes sense.

Method2(Automation):

If you were to automate this entire process then pick only top 3 or 4 or 5 words each topic because they are typically good, and their prevalance in the dataset is huge as well.

Computation of sentiment scores.

- So, we have all the features ready with us. Now let's find what all features are present in each review. Refer to the PPT for this. It is very self explanatory.

Why sentiment scores are not appropriate sometimes:

- Ambiguity:
  - Consider the word "Spring". Is it a weather or a part of the machine?
  - Consider the word "Right". Is it "Right?" or "rightly working"
  - There are many ambigous words which Model cant understand
- Multiple Features in a single sentence.
  - Consider this review "batteries are good but the adapter is bad". Sentiment score for this review is slightly negative which is "–0.2"
  - There are two features in that review. Based on our logic, it assigns –0.2 to both of the features. Though these are rare ocassions , but these exist.

    - Would this hamper our results? Answer is not if your dataset is huge because there would be more right sentiment scores than only a few wrong sentiments. T the end, you would aggregate these scores ,so it would make sense.
- Sometimes, it doesn't assign score properly because of some complex sentence structures, or improper gramar.

Why only top 50 categories?(800 is the cut off for number of reviews)

We figured that 800 is an ideal cutoff as the number of reviews inorder to have good results. Other we would be facing in acurracy due to the following problems

Sentiment Score Problems

1)Top 50 categories have 800+ reviews. Anything after the sentiment scores are not great because of the following reason.

- As discussed in the previous section that if our dataset is huge then the inaccuracy acrued because of rare false sentiments will be neglected because when we aggregate all sentiment scores , right sentiments would over power wrong sentiments, and final result souwl be more right. Problem is when your dataset is small, and you are dealing with a few reviews, and if those reviews are having wrong sentiments because of various reasons, then the information presented would be bad. We found a cut off of 800 as number of reviews upon several experimentation with various categories.

2) Mathematical Explanation:

A few observations and assumptions for this:

- When you work on 5000 reviews in total say, you would see that finally only some 1000 or 2000 reviews would have atleast one feature being talked about. 3000 to 4000 reviews wouldn't be having any feature rather just about the product as a whole. They are useless now.
- Now you are remained with only 1000 to 2000 reviews.
- Inorder to have comprehensive and realisitic results on visualizations. We would have to have atleast 10 reviews for each feature. If we are going with 30 features per category, then you may have to go for 300 matched reviews atleast in total (assuming 10 reviews for each feature). Now realisitically, like if you go for 300 matched reviews, there would be the exact composition as 10 reviews for each feature. It could be that 50 reviews for one feature, 2 reviews for another feature. So inorder to balance this out, we need to push this number of reviews to atleast 800 matched reviews. Agreeing to the fact that some reviews have so many features present in them. If we take that , then probably 600 makes sense.
- Now, doing math bakwards, inorder to have 600 matched reviews we have to have 1500 to 2000 reviews. So we have to go for a cut off 1500 technically. But we observed that "800" as the cut off works like charm.

Feature extraction Problem

- Features extracted on reviews data for less than 200 reviews are usually not great , because not all features are being talked about in just 200 reviews. Typically 500+ reviews would be good for LDA to work better in extracting features

Insights

After the modeling process, we have extracted the feature words and their respective sentiment scores for each review. The sentiment scores range from -1 to 1, with 0 representing neutral sentiment. The detailed modeling outputs can be found in file "Final_SBD_sentiments.csv", here we are showing a sample snapshot of the outputs and what each column in the Excel file represents.

Features (features)	Sentiment Score (mean_score)	Product Model (models)	Brand (brand)	Review Text (text)	Product Category (category)
power	0.04	DCK299D1T1	DEWALT	Easy to assemble and use. The blower is **power**ful enough to get any job done and with no cords to maneuver around.	Power Tool Combo Kits
…	…	…	…	…	…
power	-0.17	DCK299D1T1	DEWALT	Not as much **power** as I thought it would be.	Power Tool Combo Kits

With these outputs on all the feature sentiments from reviews, we can aggregate the data to have average feature sentiments for each brand and each product category. It allows us to analyze the most positively perceived features and most negatively perceived features for each category and compare brand performance.

Reflection and Next Steps

1. Data Collection

We have review data on different product categories collected from multiple retailers. The number of reviews ranges from hundreds to tens of thousands. After examining and exploring categories with different number of reviews, we found that in order for our modeling process to yield meaningful results, the product category should have at least 1,000 number of reviews. In future application or extension of this project, we can keep this in mind for the data collection process.

2. Domain Expertise

Selecting feature words from the LDA output is a judgement call. Different end users with different objectives may focus on different set of feature words. Sometimes we also need to manually look into the review context to determine if the feature word is valid. We would recommend integrating more

domain expertise into this feature selection process, to have analysis that is more tailored to the business problem and the audience.

3. Automate on how to select the features:

   Pick only top words(may be 3 or 4 or 5) from each topic. Experiment on this

4. LDA on Product description:
   - Do web scraping on product descriptions.
   - Run LDA model on these descriptions and extract the key words. We could do three things with these keywords.

     1)Straightaway use these as features ,and it makes sense , because descriptions have all key features mentioned. Upon our research, we totally vouch for this

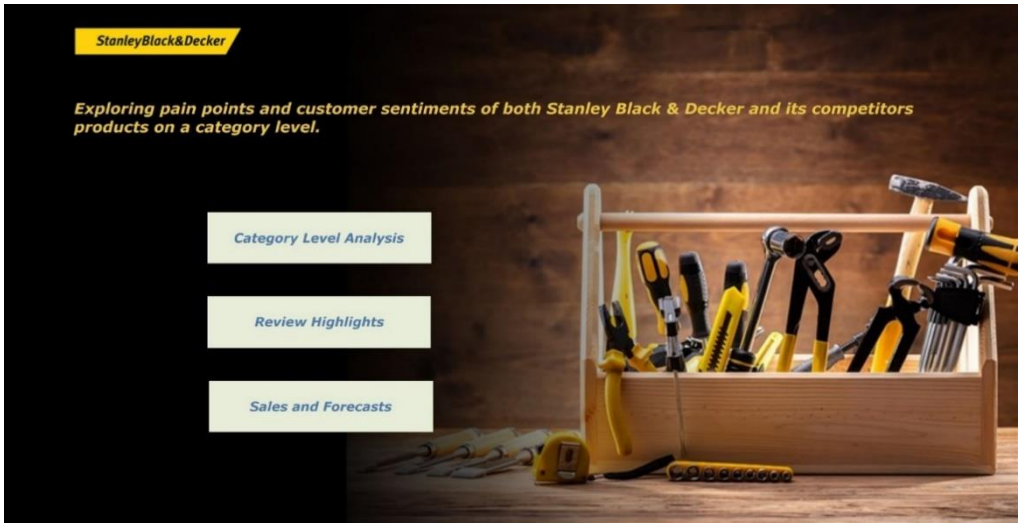
     2)Give these extracted words more weightage when you run LDA model on review data.

     3)Take an intersection between these words and the features extracted using LDA model on review data

5. Deep Learning:
   - Deep learning is the most advanced Machine Learning Technique in the world now. One of the most advanced deep learning NLP for this project is Aspect Based Sentiment Analysis. It is beyond our scope so we didn't do it. And upon our research, we figured that we have to manually train the model in the beginning (read more here : https://monkeylearn.com/blog/aspect-based-sentiment-analysis/) . But a lot of research has been going on this. It is worth trying.

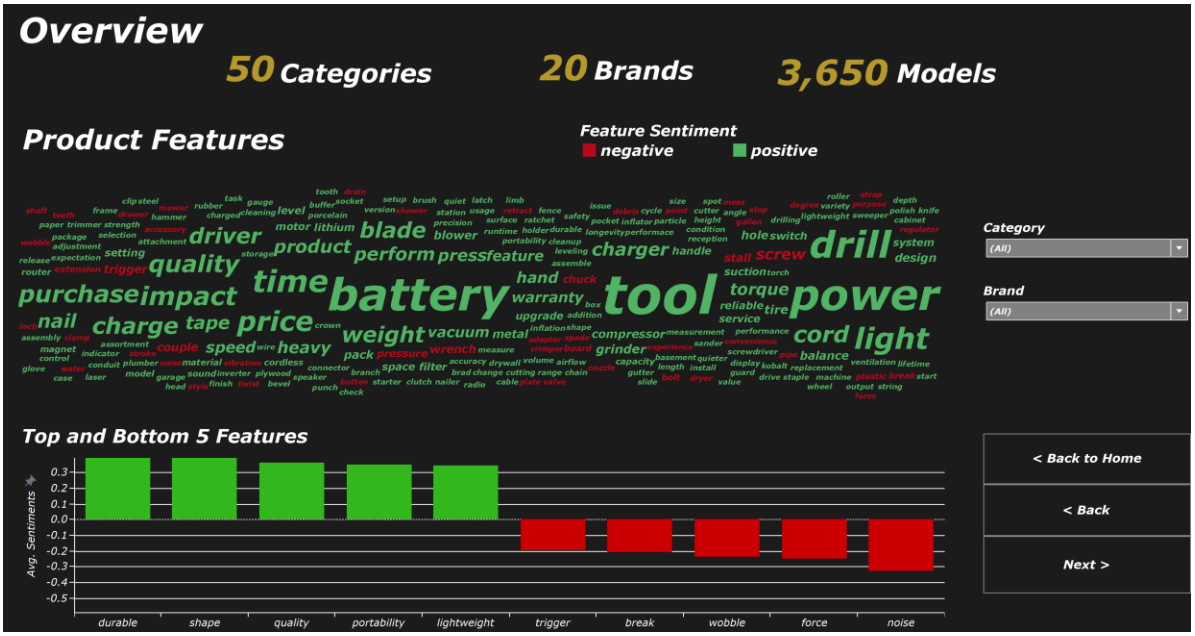
Data Visualization Manual

Home Page



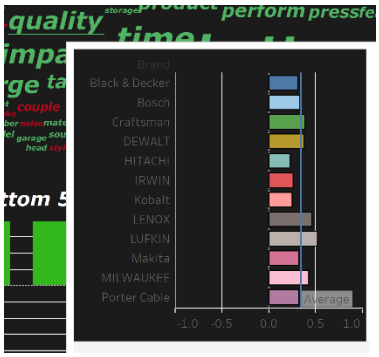
The Home page (shown above) shows the overall objective of our project: To explore pain points and customer sentiments of both Stanley Black and Decker and its competitors products on a category level. The way we explored this is by conducting a category level analysis, review highlights and forecasting sales on inventory data.

Category Level Analysis



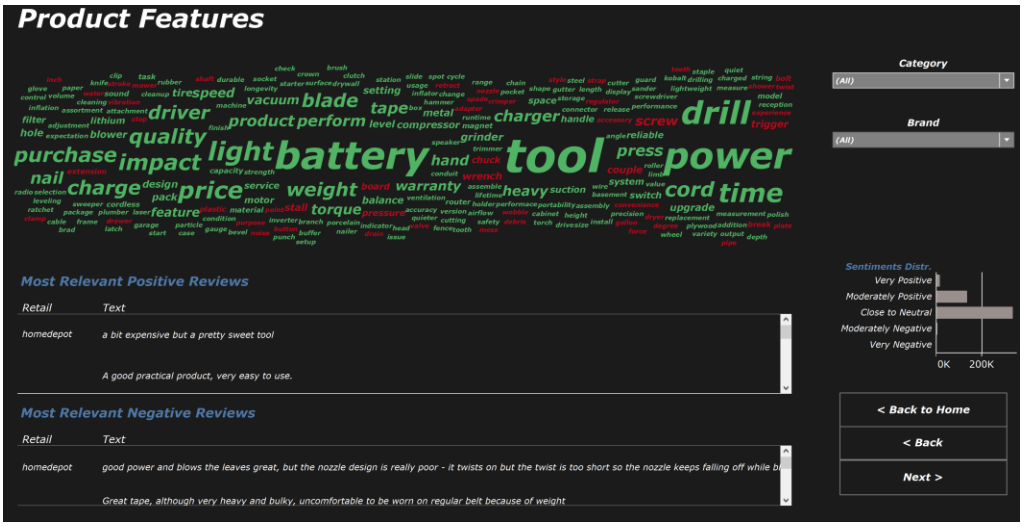
The page above shows the category level analysis for our project. On the top, we can see an overview of the number of categories, brand and models. Below it, we can see a world cloud of the product features. Words in green represent a positive sentiment while those in red represent a negative sentiment. Furthermore, the larger the text, the more frequently it appears in the customer reviews. On the right, there are two drop down menus: one for category and one for brand. By filtering through these options, one can select the features for a specific brand within a category (Eg: Category: Power Drills and Brand: Black and Decker shows the features for only Black and Decker within Power Drills).

In addition, if you hover over a specific category, you can see the sentiment score for that feature by brand. As shown on the right, the sentiment scores by brand for the feature 'quality' are shown, as well as the average of all the brands.



Finally, the graph at the base of the page shows the top 5 and bottom 5 features of all the products in the data. This is helpful as it shows the most popular features that users liked and did not like from a high level perspective.

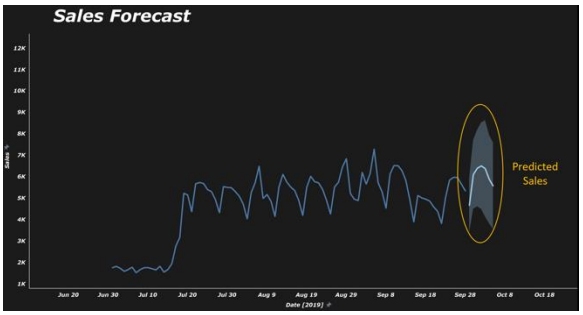
Review Highlights



The word cloud on the top of page is similar to that from the Category Level Analysis section where one can filter through features using the Category and Brand drop down menus on the top right. Although there is no hovering element here, when one clicks on a certain feature, the most relevant positive and negative reviews for that feature are shown on the bottom. This is helpful if one wants to looking at reviews regarding a specific feature. For example, if one filters 'Dewalt' products from the 'Tape Measure' category and selects 'quality', they would see the most relevant positive and negative reviews for the quality of Dewalt Tape Measures.

In addition to this, on the right there is a sentiments distribution graph. This works for both when a feature is selected or not and shows the spread of sentiments, from very negative to very positive.

Sales and Forecasts



The final dashboard shows the sales for Stanley Black and Decker products from June 30th to September 28th, obtain from the inventory data. Using a specific forecasted model, we were able to predict the sales for the week of Sept 29th to October 6th.

You can view the dashboard online:
https://public.tableau.com/profile/aj4424#!/vizhome/SBD_Demo/Home?publish=yes

Code Files Manual

Code Files

1)Extract Features

Files :

A) **R file** : generate_category.R

B) **Python file**: LDA_Model.ipynb

2)Computing Sentiment scores

Files : final_sentiment_scores.R

Order of Execution:

1)generate_category.R

2)LDA Model.ipynb

3)final_sentiment_scores.R

Purpose of each file and How to use each file

1)Generate_category.R

- tool_data.csv is your main data file that has all 396 categories
- Now, we want to build model using only one category right? So we are just extracting one category information from this whole file.

Note:

- Run first 16 lines of code and see the category_count dataframe. This dataframe has all categories names and the number of reviews in each of those categories in the descending order. From there, select the category you want to work with.

2)LDA_Model.ipynb

- This code does all the text processing and building the model.
- Just run the entire code.

Final_sentiment_score.R

- This computes the sentiment scores for each feature
- Features that are extracted from python, are to be pasted/overwritten in a vector called "topics_new"
- Just run the entire file.
- Comments in the file is self explanatory.