Course - SYS 6018, Fall 2018

Project - Sentiment Analysis on Amazon Fine Food Reviews Data

Members - Charu Rawat (cr4zy) and Vaibhav Sharma (vs3br)

Contents

Identifying the problem

Descriptive Analysis

Amazon reviews are often the most publicly visible reviews of consumer products. We know that Amazon Product Reviews Matter to Merchants [1] because those reviews have a tremendous impact on how we make purchase decisions. Studies have shown that there is a strong relationship between reviews and sale conversation rates. Reviews and ratings also help increase discoverability as potential buyers tend to filter their options based on ratings. Hence, customer feedback in form of reviews and ratings has become an important source for businesses to improve upon.

The problem today is that for many businesses, it can be tough to analyze a large corpus of customer reviews and quantify how good or bad a product is and with that understand what features make a product favorable or unfavorable. Additionally, relevant information about products and its features can sometimes be hard to identify or extract from a large volume of reviews. In Amazon specifically, there are also numerous cases where products have an overall average rating of 3 which is considered neutral. Such a situation can primary arise in 2 scenarios - one where majority of the customers generally rate a product 3, and the second case being where there's a nearly equal number of customers giving a rating to a product that falls on the either extreme side of the rating scale. This happens when certain features specifically appeal to certain people. In such cases, it become pertinent for merchants to evaluate reviews on a user level rather than looking at the overall average rating. This can be very hard when the reviews are numerous. And so, there needs to be an easier way for merchants to be able to gauge user sentiment - be it positive or negative and also understand on a broader level, the common reasons for users projecting that sentiment.

One of the major factors contributing to this issue is the subjectivity involved on the user's part in rating products. For instance, for the same product, two users can give exactly opposite rating based on the same reason - a user would probably give 5.0 to a dessert because it is sweet but another user would probably give 1.0 also because it is too sweet.

Understanding what online users think of its content can help a company market its product as well as mange its online reputation. The purpose of this paper is to investigate a small part of this large problem: positive and negative attitudes towards products. Sentiment analysis attempts to determine which features of text are indicative of its context (positive, negative, objective, subjective, etc.) and build systems to take advantage of these features. The problem of classifying text as positive or negative is not the whole problem in and of itself, but it offers a simple enough premise to build upon further.

There exists some data on the web to understand and address such issues such as the 'Fine Food Reviews' dataset sourced from Amazon[2] that we have leveraged for this analysis. The data contains reviews of various food items reviewed by users on Amazon. Each product reviewed has a review and rating associated with it given by a single user. There are other attributes provided as well. The dataset has been elaborated on in the 'Data' section.

Stakeholders

Amazon as a business heavily relies upon the relationship formed between its users and vendors through the use of their robust feedback system based upon product reviews and ratings. Hence, the stakeholders in this case is Amazon itself, the vendors selling products on Amazon that depend upon customer feedback for product intel, and customers or Amazon users who depend upon the feedback given by other customers to decide which products they should buy.

Impact

In this analysis, we have applied techniques such as TFIDF, Word2Vec and LDA for feature extraction that are eventually used in the binary classification of sentiment of reviews. Topic modeling of reviews using LDA can help cluster reviews of a similar kind under a group as well. Such structure can help merchants better understand their user feedback in context with the product features. It's also equally helpful to other users who depend upon reading these reviews before they make a judgement about buying a product.

Apart from providing valuable market intelligence to businesses by helping them better understand their products and customers, the impact of this project can also be extended across other areas as this project can serve as a layer for transfer learning onto other sectors where a problem of such nature persists. For example - there can be websites where there is no provision for ratings or websites where there's only user comments on articles and blog posts and there is need to quantify or understand customer perception.

Objectives and Metrics

Objective

By implementing sentiment analysis on the 'Amazon Fine Food Reviews' dataset, we intend to address the abovementioned problems faced by many businesses and merchants by -

- ➢ Building a model that can predict the sentiment (positive or negative), given a review

In transitioning to a sentiment classification model, we will be able to develop a method or a process that will have the ability to process text and classify it as a negative or positive sentiment. In the case where there is a mismatch between rating and a review - a review that projects a positive sentiment but is otherwise rated 3, our process might be able to correct that and classify it as a positive sentiment relative to other reviews and give the probability associated with that classification. Leveraging topic modelling can also help optimize the reviews but giving a more concise summary or a headline of a review.

Metrics

Some of the standard metrics that we will look at to evaluate model performance is F1- score, accuracy, area under the ROC curve and error rate in misclassification to assess model performance. This is elaborated upon detail in the sections later in this report.

Understanding the State-of-the-Art

Sentiment analysis is actually a pretty mature field where many algorithms already exist and one in which modules and libraries have been developed for automatic detection of sentiment. However, the accuracy goes drastically low when the rating is twisted with individually unrelated, unique and random features. Traditional approaches on sentiment analysis from scratch use word count or frequencies in the text which are assigned a sentiment value by experts and they often disregard the order of words [3]. To accurately rate a review that can be classified into 1 of the 5 categories still remains a very big challenge. For example - a review that contains the phrase "this candy is not that bad" and originally rated 4 can be misclassified as rated 1 or 2 because it contains the word "bad". Such nuances in the way people characterize their sentiments towards a product make it a difficult task but using methods such as n-grams can be addressed. Recent work has been focused on other complicated RNN models such as recursive neural tensor network [4]. The data set and the question were actually drawn from a paper coming from Stanford. [5] Though, in their paper the main challenge was not sentiment analysis, their highest test accuracy was about 40% in their studies of users tastes and preferences changing and evolving over time. This low accuracy also showed that this was a challenging data set to analyze on.

Hypothesis and Approach

Hypotheses

- A review with a higher helpfulness score is more likely to be verbose specific to each product type
- Adding predictor 'helpfulness_score' to model M will improve the ROC/accuracy performance of model M.

Data

This dataset sourced from Stanford SNAP[4] consists of reviews of fine food products sold on Amazon.
The data spans over a period of more than 10 years, including all ~500,000 reviews up to October 2012 in one sqlite database. Reviews include product and user information, ratings, and a plain text review. Below is more information on the dataset:

Number of reviews:	568,454
Number of users:	256,059
Number of products:	74,258
Timespan:	Oct 1999 - Oct 2012

Number of Attributes/Columns in data: 10
The column or features in the dataset:

1. Id
2. ProductId - unique identifier for the product
3. UserId - unique identifier for the user
4. Profile Name
5. Helpfulness Numerator - number of users who found the review helpful
6. Helpfulness Denominator - number of users who indicated whether they found the review helpful or not
7. Score - Rating between 1 and 5
8. Time - Timestamp for the review
9. Summary - Brief summary of the review

   Text - Text of the review

The ratings and reviews are both given by users whose economic, demographic, and other important data such as personal taste and preference is unknown. In such a situation the reviews and ratings will definitely be very biased, for e.g. one user might feel a product is rightly priced while the other user might feel that the same listing is overpriced. Another bias that exists is that the data is highly skewed towards the higher ratings of 4 and 5.

Execution, Approach, and Results

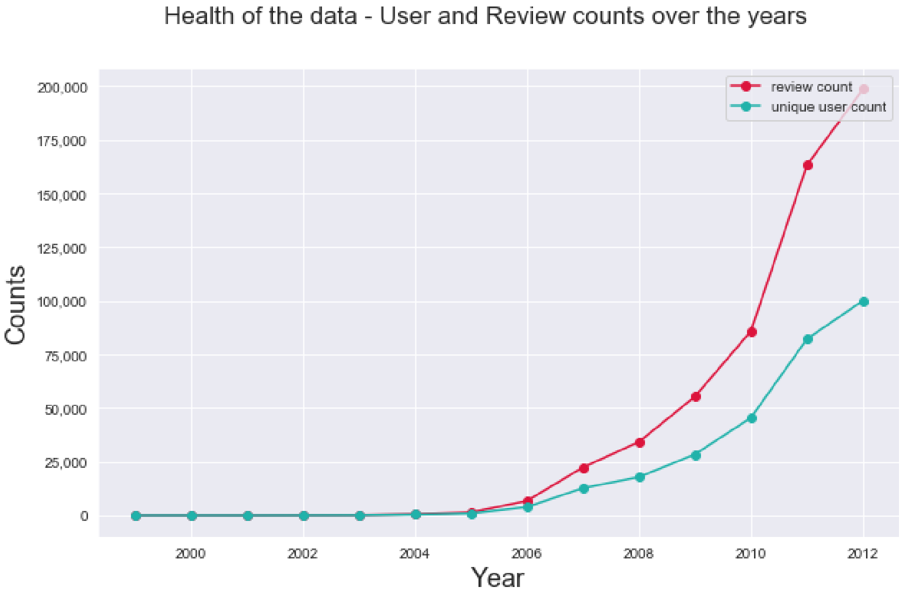
Our data pipeline involved pre-processing data, performing exploratory data analysis, leveraging text processing on it followed by building classification models and evaluating them. Below are the methods and steps followed -

Data pre-processing:

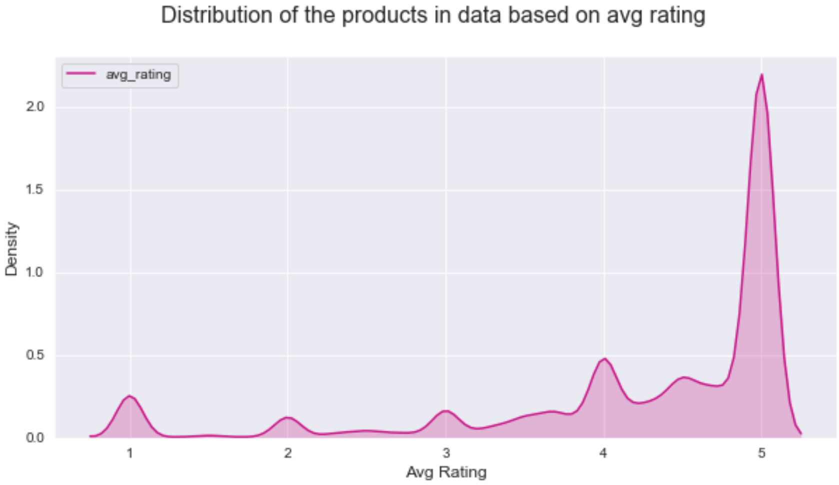
- Re-naming columns
- Checking datatype and appropriate datatype conversion of variables (timestamp conversion)
- Handling missing values and dealing with duplicated data – dropped 0.05% of the data
- Feature Engineering - creation of variables – upvotes and total responses using variable ‘helpfulness_score’, creation of variable ‘extended_review’ which is a concatenation of ‘summary’ and ‘text’, and creation of variable word count for each review
- Removal of neutral reviews - with rating 3
- Classification of reviews rated 4 and 5 to 1 (positive sentiment) and 1 and 2 to 0 (negative sentiment)

Exploratory Data Analysis

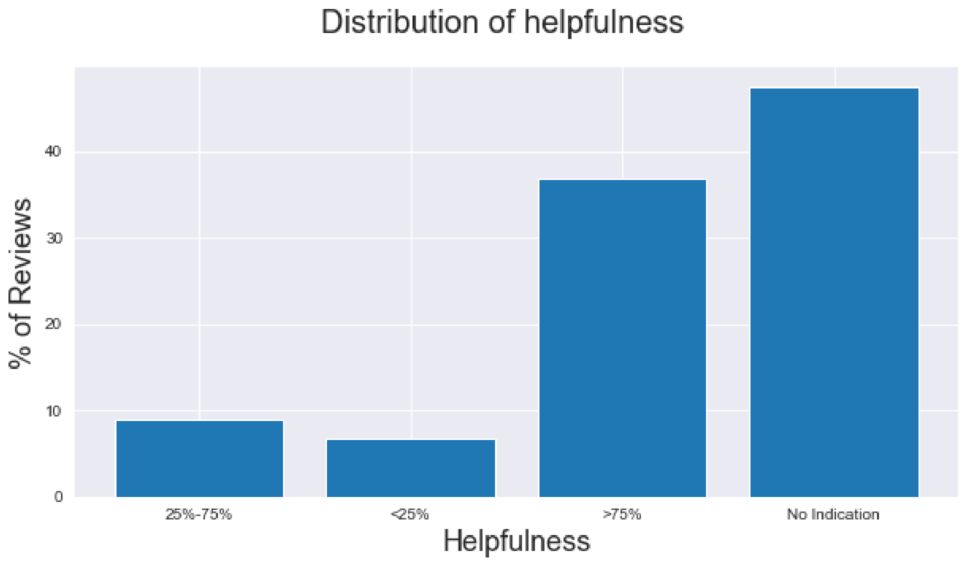
Our final processed data contains 568k reviews for 74k unique products given by 256k unique users. Below are the main takeaways from our exploratory data analysis


Health of the data - User and Review counts over the years

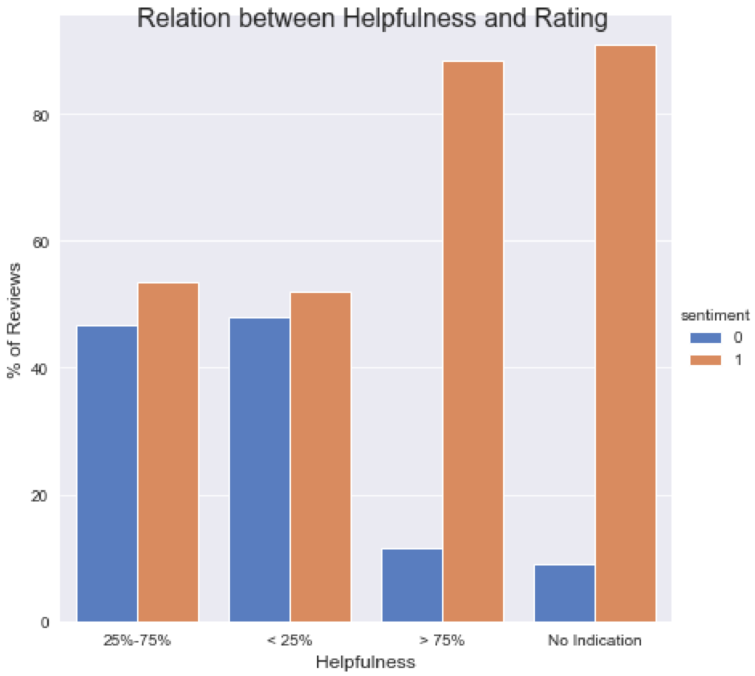
- There is an increase in the records or reviews for Fine Foods in Amazon starting 2007 onwards but this is probably an artifact of the business.


Distribution of the products in data based on avg rating

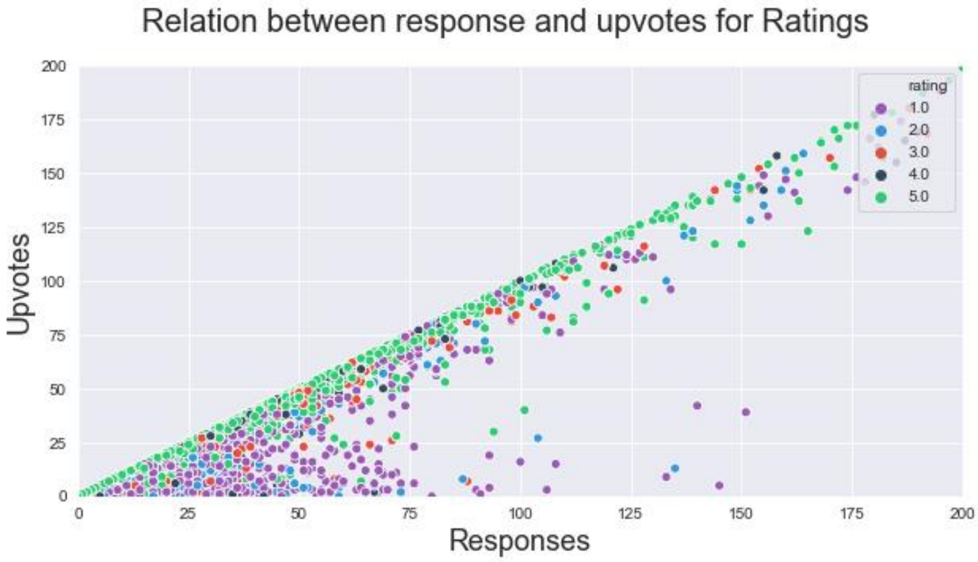
- We took a look at the calculated average rating of products, and it is visible from the plot above that the data is heavily skewed towards positive sentiment.


Distribution of helpfulness

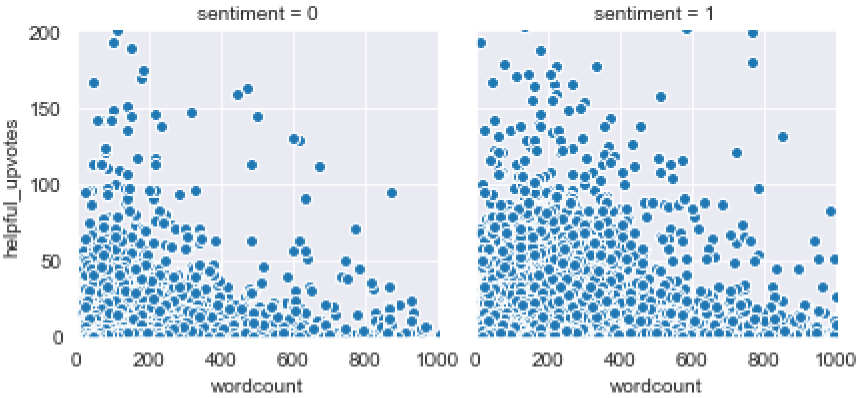
- While 45% of the data has no indication of whether a review was helpful or not, 35% of the data did consist of reviews voted upon and over 75% of the voters found those reviews helpful. Among reviews that are voted on, helpful reviews are the most common.



- If we look at the relation between helpfulness and ratings, over 75% of the users who did find a review helpful was mostly when it pertained to a positive sentiment.
- If we look at the relation between responses to a rating versus users who upvote a response for every rating, we observed that for product reviews that are rated 5, people generally tend to upvote them. That is, customers find positive reviews more helpful as opposed to negative reviews.



- The scatter between word count or review length and helpfulness below shows that positive reviews or reviews rated higher don't tend to be more verbose. In fact, positive reviews get higher upvotes when they are more concise. We see similar trends for negative reviews that have been upvoted as well. This could point to a trend that indicates that shorter reviews are more likely to be upvoted.



- We also implemented a word cloud on the positively rated reviews over a small sample of the data. We did come across some words indicative of a positive sentiment like quality,appreciate,smell, etc. Though this isn't accurate as it hasn't been weighted or normalized yet and is only over a small random sample of the data.

dog
several
have
meat
this
food
of
more
to
my
be
vitality
found
all
process
product
it
can
than
most
quality
smell
labrador
is
and
the
like
appreciate
looking
stew
she
finicky
a
better
bought
them
good

Text Processing

Prior to building the classification model we implemented text processing to extract features that could be used in modeling.

- Stop words removal: stop words refer to the most common words in any language. They usually don't have any predictive value and just increase the size of the feature set.
- Punctuation Removal: refers to removing common punctuation marks such as !,?,$* etc.
- Lower case transformation: convert all upper-case letters to lower case letters.
- Stemming: the goal of both stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For e.g. the words - organize, organizes, and organizing are reduced to one form.

The first problem that needs to be tackled is that most of the classification algorithms expect inputs in the form of feature vectors having numerical values and having fixed size instead of raw text documents (reviews in this case) of variable size. We tackled this by using the Bag-of-Words framework which involved tokenization and normalization of words in the text. We used the following techniques to obtain feature vectors from text that could be leveraged in model building.

- **Tf-idf** : Tf-idf allows us to weight terms based on how important they are to a document. In a large text corpus, some words will be present very often but will carry very little meaningful information about the actual contents of the document (such as "the", "a" and "is"). If we were to feed the count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. We instantiated the tf–idf vectorizer and fit it to our training data. The generalized formula for this measure:

$$Tfidf\ (t,d,D) = tf\ (t,d) \times idf\ (t,D)$$

Where *t* denotes the terms; *d* denotes each document; *D* denotes the collection of documents.

- **Word2Vec**: Word2Vec is a group of models (first introduced by Mikolov et al. in 2014) used for constructing vector representations of words, also known as word embeddings. Word2Vec (w2v) uses a shallow neural network to learn how words are used in a particular text corpus. The dense vector representations of words learned by word2vec have remarkably been shown to carry semantic meanings and are useful in a wide range of use cases ranging from natural language processing to network flow data analysis. These vector encodings effectively capture the semantic meanings of the words. For instance, words that we know to be synonyms tend to have similar vectors in terms of cosine similarity and antonyms tend to have dissimilar vectors. Even more surprisingly, word vectors tend to obey the laws of analogy. For example, consider the following –

  $$v_{queen} - v_{woman} + v_{man} \approx v_{king}$$

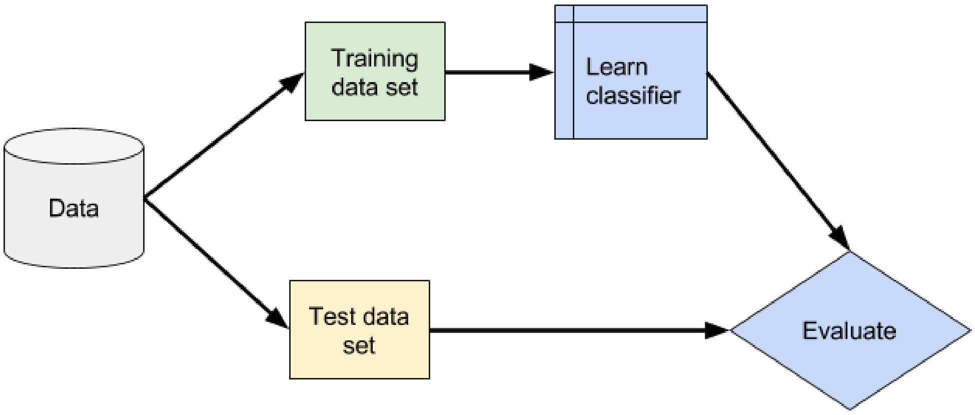
  where v_{queen}, v_{woman}, v_{man} and v_{king} are the word vectors for queen, woman, man, and king respectively. These observations strongly suggest that word vectors encode valuable semantic information about the words that they represent.

- **n-grams**: Our model contained misclassifications owing to loss in context interpretability of sentences. For eg – misclassifying the text "The candy is not good, I will never buy them again" as a positive review, and misclassifying the text "The candy is not bad, I will buy them again" as a negative review. We addressed this issue by implementing a n-grams or in this case a bi-grams model below the word2vec approach. Bi-grams count pairs of adjacent words and could give us features such as bad versus not bad.
- **Latent Dirichlet Allocation**: There were no product names or descriptions in the dataset that were easily accessible. Review summaries sometimes mentioned the product under review, otherwise there is no category label that provides a way to simply group products and user preferences. We attempted to represent text reviews in the terms of the topics they describe, i.e. topic modeling. The technique used to extract topics from Amazon fine food reviews is Latent Dirichlet Analysis (LDA). We assume that there is some number of topics (chosen manually), each topic has an associated probability distribution over words and each document has its own probability distribution over topics; which looks like the following

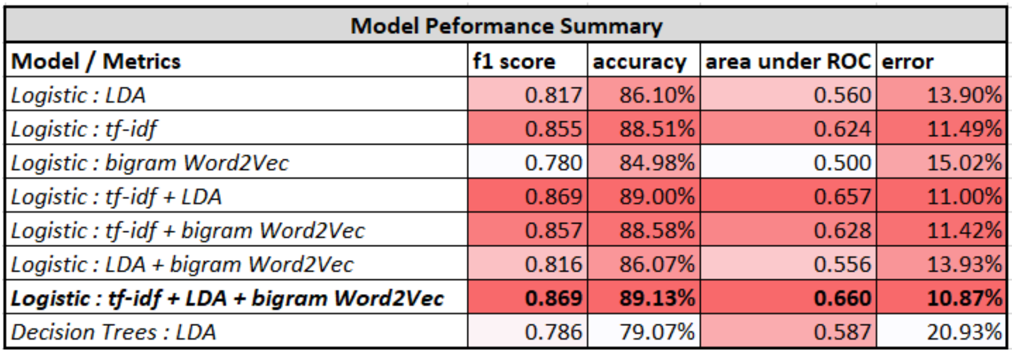
  $$p(d \mid \theta, \emptyset, Z) = \prod j{=}1\ length\ of d\ \theta_{zd,j} \emptyset_{zd,j}\ , w_{d,j}$$

  Every word in all the text reviews is assigned a topic at random, iterating through each word, weights are constructed for each topic that depend on the current distribution of words and topics in the document.

Modeling



We used 2 classifiers in this project – Decision Trees Classifier and Logistic Regression. Since the number of samples in the training set is huge, we abstained from running classification algorithms like K-Nearest Neighbors or Random Forests etc. which would be inefficient in this case particularly. We trained our models on the training set implementing k-fold validation and then tested the performance of the classifier on the test set. The models were implemented using combination of the feature vectors produced through wrd2vec, tf-idf and bi-grams along with latent dirichlet allocation. Below is a summary of the metrics across various models used to assess performance.



Model Peformance Summary				
Model / Metrics	**f1 score**	**accuracy**	**area under ROC**	**error**
Logistic : LDA	0.817	86.10%	0.560	13.90%
Logistic : tf-idf	0.855	88.51%	0.624	11.49%
Logistic : bigram Word2Vec	0.780	84.98%	0.500	15.02%
Logistic : tf-idf + LDA	0.869	89.00%	0.657	11.00%
Logistic : tf-idf + bigram Word2Vec	0.857	88.58%	0.628	11.42%
Logistic : LDA + bigram Word2Vec	0.816	86.07%	0.556	13.93%
Logistic : tf-idf + LDA + bigram Word2Vec	**0.869**	**89.13%**	**0.660**	**10.87%**
Decision Trees : LDA	0.786	79.07%	0.587	20.93%

Evaluation and Results

Generally, the results of this experiment were very successful. The classifiers managed to accurately tag a great amount of user-generated data much further past the random baseline of 50%. Our takeaway was that the model trained on the Logistic Classifier using all the features together – tf-idf, LDA and bi-gram with Word2Vec gave us the best performance in terms of accuracy and other metrics mentioned above. The Logistic classifier trained solely on tf-idf also gave us similarly good results. Compared to the logistic model, decision trees fared poorly.

We tested certain hypothesis that we had formulated at the beginning of our analysis and were able to evaluate it through our exploratory analysis and model building approach.

- Adding regressor 'helpfulness_score' to model M which in this case was Logistic with tf-idf **did not** improve the accuracy performance of the model. The previous accuracy was 88.51% and it marginally went upto 88.7% which wasn't a considerable improvement.
- We did not find that the verbosity of a review is related to its helpfulness. In fact, through the EDA we concluded that reviews that are voted helpful tend to be concise and shorter in length hence invalidating our hypothesis.

References -

[1] 6 Reasons Why Amazon Product Reviews Matter to Merchantshttps://www.entrepreneur.com/article/253361

[2] Amazon Fine Foods reviews dataset- https://snap.stanford.edu/data/web-FineFoods.html

[3] Bo, P. (2008) Opinion Mining and Sentiment Analysis, Foundations and trends in information retrieval, 2(1–2): pp. 1–135. DOI:10.1561/1500000011

[4]Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the conference on empirical methods in natural language processing (EMNLP), volume 1631, page 1642. Citeseer, 2013.