Aashka Agarwal A20376307

ADVANCED DATA MINING
FINAL PROJECT REPORT
FALL 2017

Amazon Fine Food Reviews

ABSTRACT:
Sentiment Analysis is a field of information mining. In this project, we try to analyze the reviews given by the customers on various products and predict their polarity (positive, negative, and neutral) of opinions using different machine learning algorithms. As the utilization of web based business sites is expanding lavishly, users not just purchase an item on sites yet additionally give their input and recommendations that is destined to be advantageous to different clients. There is particular sort of Sentiment Analysis, for example, sentence level, record level, and viewpoint or highlight level. It helps buyers in better basic leadership. Various Machine Learning calculations like Naïve Bayes, SVM and Decision Tree are utilized for different assignments which are completed in conclusion investigation.
Also, we try to analyze the user behavior on the product to determine the main elements used by them to rate the product and recommend the similar type of product to the user.

INTRODUCTION:
The aim of this project is to extract sentiment from approximately 600K records and analyze the implications they have in the business area. The data set we used in our project is called Amazon fine food reviews.
Amazon fine food reviews dataset is capturing the reviews of the customer and analyzing them, to undertake the measures to improve the performance of the products, businesses. Analyzing the user's reviews helps businesses understand the needs of the customers.
In our project, we will analyze the reviews of the users posted on amazon for different products in different cuisines. We will be performing sentiment analysis to detect the categories for reviews on the basis of the keywords
This will help the businesses improve their products with respect to the customers reviews.
In this approach, positive or negative sentiment is distinguished from the as of now extracted features. It is a fine-grained investigation demonstrate among every single other model.
The users post their perspectives their thoughts about the product and a corpus is created and it is pre-processed Feature Extraction is performed to find the important features of the product. It is then segregated into 3 different categories positive negative or neutral by applying machine learning algorithm.

DATA:

1.1 Data Source

The primary data source is amazon fine food review data set from Kaggle.com
Link: https://www.kaggle.com/snap/amazon-fine- food-reviews/data
The size of the data is 251 MB(approx) with 568,454 records of food reviews from Amazon users It is a complete text file with different metrics and elements to analyze.
The dataset has been successfully downloaded and loaded into the system.

1.2 Data Format

1. The user reviews and all other elements are in text format.
2. The reviews.csv file consists of 10 columns including information related to product, user, reviews etc.
3. Below is the basic data table structure of the data used in the project.

Sr no	Column Name	Type
1	ID	Int
2	Product ID	String

3	User ID	String
4	Profile Name	String
5	HelpfulnessNumerator	Int
6	HelpfulnessDenominator	Int
7	Score	Int
8	Time	datetime
9	Summary	String
10	Text	String

1.3 Programming Language, packages

1. **Programming Language:** Python and R for wordcloud.
2. **Packages used:**

- Numpy: useful linear algebra, Fourier transform, and random number capabilities
- Pandas: working with numerical and tabular data
- Matplotlib: to plot the graphs
- Datetime: manipulating dates and times in both simple and complex ways.
- Nltk: natural language processing
- genism: Produce word vectors with deep learning
- Scikitlearn: Classic machine learning algorithms

EXPERIMENTS:

The main aim of the sentiment analysis to use for our project is to determine the users reviews to find the impact on the new food products and the future aspects for the products. It would be helpful in understanding the subjective reasons as to why the users are or are not buying the food product, how are they feeling about it(experience). Did the products fulfilled the user's expectation and etc. The main experiments performed are:

1. Pre-processing of data
2. Feature Extraction
3. Negation Handling
4. Model building
5. Upsampling
6. User Behavior Analysis

1.1 PreProcessing:

Preprocessing is the process of cleaning the data and preparing the text for analysis. The users while writing a review uses informal language and words which is not understood by the computer and they don't have any effect on the classification problem. Eliminating the stop words helps us to reduce the dimensionality of the data.
I followed following steps for text preprocessing:

1. Converting words to lower case: The text is converted to lower case using string package.
   Example: Right now I'm mostly just sprouting this so my cats can eat the grass. They love it → right now i'm mostly just sprouting this so my cats can eat the grass. they love it

2. Remove stop words: Stop words like I, am, have, and are removed by using the stopwords function from nltk package.

Example: Right now I'm mostly just sprouting this so my cats can eat the grass.They love it → Right I'm mostly s prouting cats can eat grass.

3. Removed punctuations: All the punctuations like (‘,”!-?) are removed from the text by using punctuation function in string package.
   Example: right i'm mostly sprouting cats eat grass → right im mostly sprouting cats eat grass.

4. Stemming: Stemming is a process of extracting the root word of the word used which reduces the size of the text. We used PorterStemmer to perform stemming since it is one of the most effective algorithms to perform stemming.
   Example: Right now I'm mostly just sprouting this so my cats can eat the grass. They love it → right now im mostly just sprout cat can eat grass.

1.2 Feature Extraction

A piece of information that is used to predict is termed as feature. A technique of generating features which would be used for the ML model to get the accuracy of the model.
It also help in determining the importance of word in a particular text which would help us to determine the customer thought process in selecting and purchasing the food.
I used TFIDF and count vectorizer functions from sklearn package to get the matrix.
The TFIDF vectorizer basically converts the raw text into a sparse matrix of important features.
The TFIDF and Count vectorizer matrix were formed and then were used to train and test the models further.

A very basic technique of feature generation is Bag of Words is also used to determine the frequency of words in per review. Text is transformed in a bag full of words and then we calculate various aspects to featurize(frequency, no of time the term has appeared) the text.

1.3 Negation Handling

The main problem while performing sentiment analysis is to handle the negate words. Example: not good, not bad.
There meaning changes as per the use in sentences. I used 2 different approaches to handle negation in the project.

1. Mark_negation
2. Pos_tag

Mark_negation is the function in nltk package where it includes prefix as NEG to all the words in the sentences where it finds a negative word.
Example: I don’t like to wear this dress. Now, as soon as the function encounters the don’t word it starts applying the prefix neg to all the words in the sentence. I don’t_neg like_neg to_neg wear_neg this_neg dress_neg which was finally termed as negative sentence.
In most of the records in the dataset it predicted the right polarity of the review but it was not 100% correct.
To overcome this issue, I tried pos_tag function of to handle it efficiently.
Pos_tag function is basically the parts of speech token which is executed after the string is tokenized. It separates each word with its form.
Example:
I don't know if it's the cactus or the tequila or just the unique combination of ingredients, but the flavor of this hot sauce makes it one of a kind! We picked up a bottle once on a trip we were on and brought it back home with us and were totally blown away! When we realized that we simply couldn't find it anywhere in our city we were bummed. Now, because of the magic of the internet, we have a case of the sauce and are ecstatic because of it.If you love hot sauce..I mean really love hot sauce, but don't want a sauce that tastelessly burns your throat, grab a bottle of Tequila Picante Gourmet de Inclan. Just realize that once you taste it, you will never want to use any other sauce.Thank you for the personal, incredible service!

A negation term with an adverb POS forms one of the following three possible semantic relationships. 1) The negation word RB may modify an adjective JJ which further describes a noun NN (i.e. RB, JJ, NN). 2). Similarly, the negation may modify a verb VB which further describes a noun (i.e. RB, VB, NN). 3) The negation term may modify

an adverb which further modifies an adjective or a verb. In this case we have two different relationships (i.e. (RB, RB, JJ, NN) and (RB, RB, VB, NN)). First the semantic relation that a negation forms is determined and then all opinionated words involved in such a relationship are considered as the scope of negation.

The example given above is classified as positive which is correct but the same sentence was termed as negative by mark_negation.

1.4 Test and Train size

Training data is basically the data that is trained by applying the model with specific elements. After the model is trained with the data it is then applied on the testing data to see the accuracy of the model.
Test and Train size depends on various aspects:

1. If we choose less testing data, it would have greater variance which would in turn be not satisfactory.
2. If we choose large testing data, it may classify wrong because it might not have the similar type of training data.

Hence the data should be chosen as such that neither the variance is too high or too low. Mostly, the data is splitted into ratio of 80:20, where 80 is training size and 20 is test size.

I tried the test and train size as standard 80:20 and 70:30. But after applying models, the split of 80:20 was better at giving the accuracy of model sand also classifying the sentiments into its respective bucket correctly.

The dataset we had in our project had **420651** train records and **105163** of test records.

1.5 Class labels:

Before classifying the data, it is very important to know whether the data has labels or not.
We had 5 labels for our data but we converted it to 2 labels as negative (0) and positive (1) by distributing the reviews with score >3 as positive and <3 as negative.
We eliminated the reviews whose score was 3 since they were very neutral reviews. Below is the Figure 2 where it shows the frequency of the reviews as per their scores.

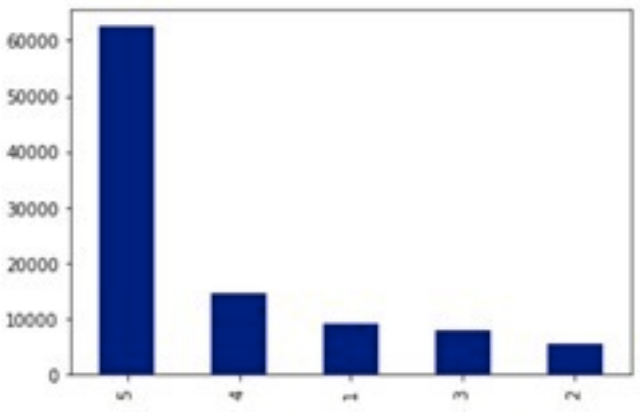

Figure 2

Analyzing the Figure 2 we see that most of the reviews have a score of 5 concluding that the dataset Is positive skewed.

1.6 Modeling:

After the data is trained and labels are being interpreted, it's time to test the model and check the accuracy of the models. I tried with various models to check the accuracy of the model and also the effectiveness of the words

that are classified as positive and negative. I selected the F1 score as the measure of accuracy in our dataset. F1 takes the harmonic mean of both the precision and recall. I tried Logistic regression, Linear SVC and Naïve's Bayes to check for the accuracy.

1. Logistic regression: After testing the model with Logistic regression, I found that is working fine but the words classified as positive and negative resembled to a great extent.
2. Linear SVC: It did not work well with the original data. The words were out of the scope.
3. Naïve Bayes: I also tried NB (Multinomial and Bernoulli) since it works best for the text classification data but in our case, it failed miserably. The words classified as positive were making sense but negative words were a waste. The words were of no importance and it had the least F1 value.

Below is the table for comparison for the models without applying the pos_tag or mark_negation:

Model	Features	F1score	Confusion Matrix	Positive Words	Negative Words
Logistic Regression	191891	0.93	[11693 4756] [2037 86677]	Blowout,amazig	Disappoint,decept
Linear SVC	191891	0.92	[12098 4351] [3606 85108]	Certaini,hamper	Terrible,smarter
Multinomial Naïve Bayes	191891	0.91	[10922 5527] [4145 84569]	Love,flavor	Zoster,zzzzz
Bernoulli	191891	0.88	[8799 7650] [4960 83754	Like,love	Zz,zthis

Table 1A

After such poor result, we tried with negation and N-gram technique to check whether the model works better or not.

Model	Technique	Features	F1score	Confusion Matrix	Positive Words	Negative Words
Logistic Regression	Negation+unigram Negation + bigram Only bigram	240330 4314881 3486618	0.93 0.94 0.94	[11104 5345] [1768 86946 [11366 5083] [882 87832 [11649 4800] [1011 87703]	Great, best, delici great, best Perfect,high recommend	Money_NEG,wors Worse,disappoint Wont buy,bland
Linear SVC	Negation+unigram **Negation + bigram** Only bigram	240330 **4314881** 3486618	0.93 **0.96** 0.96	[12581 3868] [3163 85551] **[13462 2987]** **[1327 87387]** [13532 2917] [1380 87334	Blowout,delus **Great,delici** Great,high recommend	Starsand,wor **Return,worst** Don't recommend,wont buy

Multinomial Naïve Bayes	Negation+unigram Negation + bigram Only bigram	240330 4314881 3486618	0.91 0.78 0.85	[10922 5527] [4145 84569] [449 16000] [2 88712] [505 15944] [2 88712	Love,flavor Love,great Love,great	Zoster,zzzzz Zyllius, zziga Zyllius,zziga
Bernoulli	Negation+unigram	240330	0.88	[8799 7650] [4960 83754	Like,love	Zz,zthis

Table 1B

After observing the values in the above table 1B, we concluded that the Linear SVC with bigram and negation works best for the dataset. If we see the confusion matrix for the same, we see that only 2987 were wrongly classified as negative i.e label 0 and 1327 were wrongly classified under positive i.e label 1.

I also tried the trigram but the results were unsatisfactory and hence did not mentioned in the table. If we compare all the aspects with the other models executed in table 1A and table 1B, Linear SVC with bigram and negation has the most accurate result in all the aspects (accuracy, confusion matrix and the words classified).

Suggested by professor, in the progress reports that we should handle negation very carefully, so we tried pos_tag function of nltk package to see if the results are more better than the mark_negation and ngram techniques.

After the analysis, we observed that pos_tag function returned similar accuracy but varies in the confusion matrix. The positive and negative sentiments were more misclassified as compared to mark_negation. We are still working on the pos_tag function to understand it better. Till then we conclude, that mark_negation works better in this case.

1.7 Parameter Tuning

1. Changing parameters for LinearSVC
Changing the value of C from 30 to 40. C is the penalty parameter of the error term.
svm.LinearSVC(X,y,x_train,C=40)

- As the value for c increases, accuracy increases initially. After reaching a certain optimal value the accuracy starts falling.
- The regularization value at the start prevents the model from overfitting, hence the accuracy goes on increasing upto a certain value of regularization.
- Then after certain value the model starts getting underfitted and accuracy starts decreasing as the algorithm starts finding local minimum.

2.Changing Train_Test_Split value and plotting the effects of that on accuracy
For train-test split = 0.8, the algorithm gives very good accuracy.
Since there is not much of difference in accuracy, I selected LinearSVC() with C = 30 and max_iter =1000

1.8 Upsampling

As mentioned earlier in the report, the dataset we are using is highly skewed towards positive, we tried upsampling of data to see the results. We again sampled the data in equal numbers for positive and negative labels but with the score of 5. All the positive reviews are now resampled as positive and negative in equal

number and performed all the experiments but that didn't help either. No model gave us the words that were fitted into its correct sentiment.

1.9 User Behavior Analysis

User Behavior Analysis is the study to understand the pattern of the user through which they tend to give score. What aspects they in their mind while rating the product, what features do they use to rate the products, do they really mean the score and the words they use while describing the products(contradiction). What other similar products do the user like on the basis of the rating.

For the first part of experiment, I considered userid, profile name, score, total count(calculated) and mean score(calculated).

To start this experiment, we started with 5 users at random and calculated the most frequent words used by the user for every score to understand the frequency of the words used. We wrote a function to calculate the words and their total count per user on the basis of the score.

User	**A2M9D9BDHONV3Y**
Total score	448
Average	4.5
Score 1	Never, buy,dry,tasteless
Score 2	Try,like,wrong,buy
Score 3	Taste, like,notlike
Score 4	Highly recommend
Score 5	Must buy

Table 2A

By analyzing the table 2A user A2M9D9BDHONV3Y A. Bennett who gave approximately 448 reviews with an average score of 4.5. We calculated the total number of review on the basis of score also including the negation features. We observed that. Bennett uses the words like give it a try for score 2 reviews and must buy for score 5 reviews. After getting the results, I expanded it with the whole dataset and recorded the results. For every user, the total sum and average of the scores they have provided irrespective of the product was calculated.

The main issue to test this experiment was the selection of the user. To have this fixed we tried hit and trail method and plotted graphs of random users with the scores they have given to the products. But this method seemed to be very tedious. One user A30XHLG6DIBRW8 C.F Hill gave mostly the score 5 reviews. Hence, we discarded this user for testing.

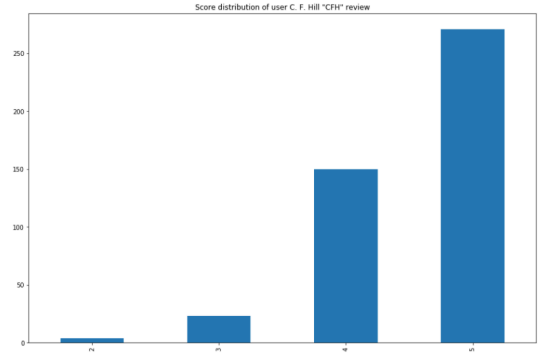

Figure 3A

We came up with another approach of extracting the users whose mean score was >3.5 and <2.5 for analysis. We found a list of users with the above condition and took 1 user for testing. A2M9D9BDHONV3Y was a very neutral u ser means, he gave reviews in all the categories. Hence, we chose her further for analysis.

Below Figure3B is the graph for the user A Bennett(A2M9D9BDHONV3Y):

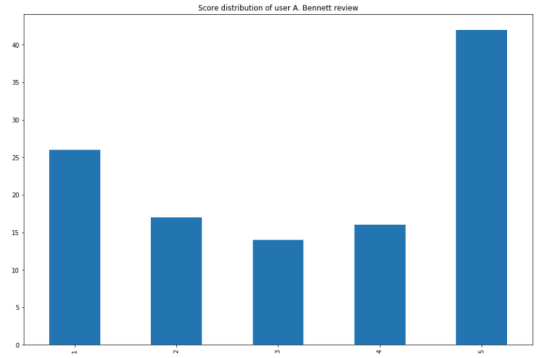

Figure 3B

Not much useful discoveries except positive / negative tones, since people are very different in texting style.
As we have used the, concept of pos_tag, we understood the importance of part of speech and know its effect on the orientation of sentence
We no focus only on the adjective words given my users to get more analysis on the user trend.

We again wrote a new function to handle only adjective words form the whole review sentence and grab them on the basis of the score per user. After analyzing further with the adjectives words, we also got to know about the features used by A Bennett for type of the food. Bennett hates the food products which are dry and expensive but loves the product which are organic and fresh.
We expanded this analysis on all the users of the dataset and found out various features used by the customers. On observing the words and the scores of some of the users, I found that there is contradiction in their reviews. One such user was A2GDBBZMMBX1L M.Barron gave a review of score 2 but used very positive words like must try, great product.
By observing such type of records, I got to know that some users do not mean to say what they are actually penning down. We trained the model and the accuracy(F1) was 95.6%.
We are still in progress of improving this further and getting to analyze more about the product and the users.

ERROR ANALYSIS:
Initially, we started with the normal sentiment analysis without handling negation but after a while when the results were not accurate and we analyzed further.
Also, we got a review from professor to tackle the negation. We learned more about the process to handle negation.
I tried 2 approaches for handling negation as stated earlier in the report.
Mark_negation and pos_tag functions of nltk package.
Pos_tag function was basically tokenizing the sentence into part of speech tag with its algorithm to identify words in different scenario.
In our dataset, pos_tag didn't mark a significant progress in identifying the words into its orientation.
Mark_negation did a better job with bigram technique in classifying the words into its orientation.
As our data was more skewed towards positive bucket, we tried to upsampling the data and apply the models to test the accuracy and the words.
During the upsampling of the data, we broke the records into equal buckets and tried our algorithms. But again, it failed and there were no good results with upsampling.
Below Table4A shows us the results from mark_negation and pos_tag:

Original Reviews	Mark_negation	Pos_tag	Manually Result
I recently tried this flavor/brand and was	positive	Neutral	positive

surprised at how delicious these chips are. The best thing was that there were a lot of "brown" chips in the bsg (my favorite), so I bought some more through amazon and shared with family and friends. I am a little disappointed that there are not, so far, very many brown chips in these bags, but the flavor is still very good.			
fantastic!you can not beat this taste and you can not resist to it only one if you are spice lover	Positive	Negative	positive
They changed the Chips now they taste horrible	Negative	Negative	negative

Table 4A

The above table shows some of the examples of the mark_negation and pos_tag results. We observed that the first 2 records were wrongly classified as neutral and negative by pos_tag but were correctly classified as positive by mark_negation. The 3rd record was classified as negative by both the approach.
This are few examples illustrated in the report but when we observe the whole dataset we see that mark_negation play a better approach in our dataset.

One solution to this issue is to include more negative reviews form the secondary source and try with our analysis. We would definitely try this and extend the project scope.

Conclusion:
The outcomes delivered through machine learning strategies are very great in contrast with the human created baselines. In terms of relative performance, Naive Bayes tends to do the worst and SVMs tend to do the best, although the unsupervised algorithm uses bigrams containing an adjective.
Bigram presence information turned out to be the most effective; in fact, none of the alternative features we employed provided consistently better performance
A human would easily detect the true sentiment of the review, but bag-of-features classifiers would presumably find these instances difficult, since there are many words indicative of the opposite sentiment to that of the entire review.
Below are some of the top words that were correctly classified by our best classifier Linear SVC with its classification report.

Positive	Negative	Confusion matrix	Classification report
great delici best highrecommend love perfect excel awesom wife enjoy good fantast	high hope bland return unfortun don't recommen d unfortun threw tasteless stale disgust	[13462 2987] [1327 87387]	Classification report for precision recall f1-score support 0 0.91 0.82 0.86 16449 1 0.97 0.99 0.98 88714 avg / total 0.96 0.96 0.96 105163 Model Accuracy: 0.9589779675361106

nice amazing	raw return_NEG horribl worst_NEG terribl disappoint worst		

Table5A

As we can see, the classification report of the linear SVC with bigram and negation shows us the scores for all the positive and negative class labels. It is the only model which is misclassifying the elements in a very small number with a good accuracy.

Learning scenario:

It was a great experience to work on this project. Since I and Hiral were a team of only 2 people we executed the whole project together. Individually, I learned more about the sentiment analysis and negation handling.
Since sentiment analysis and negation handling was a new topic to work on, I learned many aspects to handle the words accurately. Behavior analysis was an interesting topic to work upon. Getting functions to populate different stuff from the dataset and then analyzing the results was worth the course.
I hope to work on more on recommending the products to the user and label the similar type of users in 1 category through the use of KNN approach. I would also like to have deeper dive in more features that can be analyzed through the user perspective. We look forward to addressing this challenge in future work.

References:

1. Negation Identification and Calculation in Sentiment Analysis by Amna Asmi and Tanko Ishava
2. Thumbs up? Sentiment Classification using Machine Learning Techniques Bo Pang and Lillian Lee and Shivakumar Vaithyanathan