Pandya, Priyank Umesh
priyank.pandya@wsu.edu
CptS_415 Big Data

December 16, 2018
WSU ID: 11598627
Fall 2018

Project Report

Exploring Amazon Food Review

1. Introduction

Amazon (online website) is one of the biggest e-commerce stores in the world founded in 1991. It has variety of products and provides to millions of users in the world. The product falls in categories of food and groceries, electronics, clothing and beauty product and many others. In this product, Amazon ask users to give their review and experience about the products. In addition, there is rating options too that user can put and view the current rating of the product regarding the product information and its warrantee and guarantee claims. As the time goes by the review of the product increases exponentially. This helps the other users to buy product as the amazon provides critical and positive information of all the product and enhance the products customer experience.

E-commerce has changed the way of shopping. Despite lots of effort to support the e-commerce business, there are some problems that the customer is facing. [3] Few of the problem that customers are facing are quality issue, delivery and logistics, digital payment failure, Additional charges, unclear return and guarantee policies and lack of security. In quality issue, there is intentional mislead customer to increase the sale is done by fraudulent sellers. Thus, it if difficult based on review to assure the reliability of product quality. This is largely seen while selecting the size during buying the product related to clothing and footwear which is like gamble game. The other problem of e-commerce is when tracking system is given to the customer, however they are not accurate. The delivery happening when we are at work and issue of returning product when there is problem exist. In addition, some of cities the delivery service is not provided. The other main problem is transaction system that make failure in digital payment overhead. If amount is retrieved to the customer, then customer must wait for 7-10 days till amount is refunded to the bank amount. The additional charges are placed in name of shipping or something like that if order of customer is not having enough to qualify the free shipping. Some of the e-commerce companies are having improper policies and contains low-quality or damaged product. The worst part for the customer is that they are having no way to return product. The risk of banking data been sent to the companies lead to risk of cyber security. All this problem is having direct or direct related due to the review of the post or rating. This post or rating are led by fallacious website or people of the e-commerce companies who work on all this stuff to have short term gain by exploiting the customer.

1.1 Related Work

Based on the problem, various paper was read related to text analysis. This analysis is based on computational of opinions, subjective of text, which are in text mining field. There are many algorithm and applications investigated for the research of the project. From the articles and the paper, we can relate text mining from the post and rating of the product which are based on emotion detection, transfer learning and building resources. From the paper, we learned that there are two ways user can achieve better review of the product which are through sentimental analysis which is emotional analysis and opinion mining which computational study of people's opinions, attitudes and emotions for the products.

$C. Zirn$, et. al. talks about the Fine-Grained Analysis with Structural Features [4]. In this paper, authors talk about automatic framework for the fine-grained analysis based on sentiments. The authors use the Markov logic in order to integrate polarity score. This will help to score different sentiment lexicons with information. Along with that the information is related between neighboring segments and evaluate product review based on the approach. Thus, using Markov classification we get high flexibility to score from various sources. Since paper was based on supervised learning authors can work on manually corpus to learn the weights and can classify the segments into positive, negative or neutral.

G. Ganu, et.al. discusses about the important asset of user and feedback is the tool from which users buy the product or other services. Authors [5] focus is to identify the text review and using knowledge-based improve the user experience by using review. In this paper, we are using classification techniques like regression based and ad-hoc based recommendation measure for textural component of user review. The analysis was conducted between textual review and rating in which it was found that textual review is more helpful than rating as users get best answer and numerical star derived is not much useful for the user.

S. Brody and N. Elhadad explains about the unsupervised learning for extracting and determining the review text. Here unsupervised learning will classify positive and negative adjectives. They use Local LDA to prevent inference of global topics and model towards ratable point of view. Authors worked on mathematical representation on score for determining representative words. The classification is used to base on polarity of cuisine adjectives. From the experiment, authors have successfully classified based on three aspect: Food-General, Mood and Staff.

In addition, for online review analysis, author Y. Jo and A. Oh has discussed on web contain sentiments based on aspects of product and services. In the paper, authors [7] have explained problem of finding the automatically discovery that evaluates the review and sentiments for different aspect. The Authors have used SLDA for probabilistic generating model. ASUM (Aspect and Sentiment Unification Model) is used to model sentiments towards different aspects. The ASUM find pair {aspect, sentiment} based on seniti-aspect. The best advantage of ASUM from the paper was that it does not require sentiment label that are expensive to obtain for the review.

Finally, the Amazon itself has few opinions for rating review prediction. Authors [8] explains that Unigram and IGram has two models. The problem Unigram and N-Gram representation text faces is they are typically sparse in training set. So, authors have introduced bag-of opinions which has root word, set of modifier word and more than one negation words. They used ridge regression for evaluating the score assigned by numeric score. The experiment performed by authors show bag of opinions likes CLO (Cumulative Linear Offset) model. In bag-of -opinions we capture n-gram effect. However, it is in structed way which contains modifier, words, and negators. In order to avoid explosion of feature space this is used for explicit n-gram model. Authors have developed ridge regression for rated review in domain independent corpora. Thereafter, regression model is transferred to domain-dependent application. In this, we derive set of statistics over opinion score in document. It uses theses feature for standard unigram and predicting the rating review. Author have used BoO representation to model domain-independent opinions and this learned model has

potential application in sentiment summarization, opinionated information retrieval and opinions extraction.

1.2 Problem Statement

The big problem that amazon is facing is the fake review problem [1]. The news of amazon review fraud online and efforts of company to prevent from getting spoil is done on large scale [2]. This type of review problem is needing to have some solution so that customer can receive better accurate review of the product quality and other aspects. The purpose of this project is getting food review based on emotional analysis which can be derived from the post and rating of the product listed in amazon website.

1.3 Solution

In order to solve the problem, we will classification methods and so we will classify the review based on the text as positive and negative. In addition, we need to use classification techniques to deal with star rating to get the positive and negative rating.

1.4 Dataset for the Project

The Dataset is taken from the Kaggle [9]. This dataset is in the form of CSV file and it is separated by form of comma. It also contains date from October 1999 to October 2012. In addition, there include 74,258 food products, 568,454 reviews (includes star rating) and 256,059 users.

2 Data Interpretation of the dataset

Below is the current rating verses dataset distribution graph shown below:

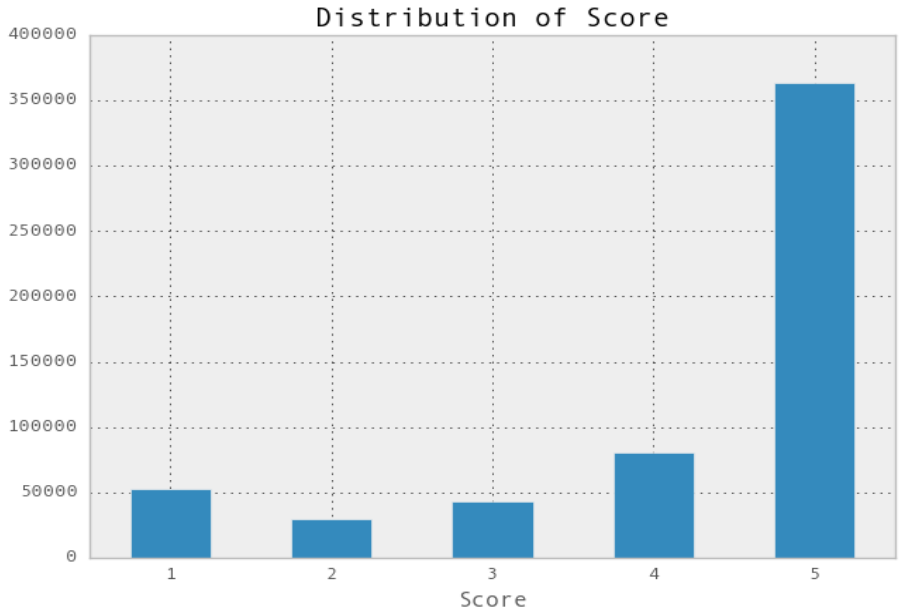

Figure 1: Data Distribution Graph

The review of product plays important role for knowing the product. In the dataset we have 568,454 reviews and it contains positive, negative and neutral reviews in it. We need to classify the review by removing and store in list the rating having 3 stars as it is considered has neutral. This is done as it does not belong to any of the category of positive and negative.

Figure 2: Positive Word (Left) and Negative Word (Right) of Unigram Word

We then build the bi-gram word cloud. This contains most frequent bi-gram from the dataset. The purpose of this is to showcase order of the word and display phrase of 2-word. Below are the Positive and Negative Bi-gram words.

Figure 3: Positive word (Left) and Negative Word of Bigram words

3 Methodology

The project is built on Spark based on *MLlib* library and *PySpark* library.

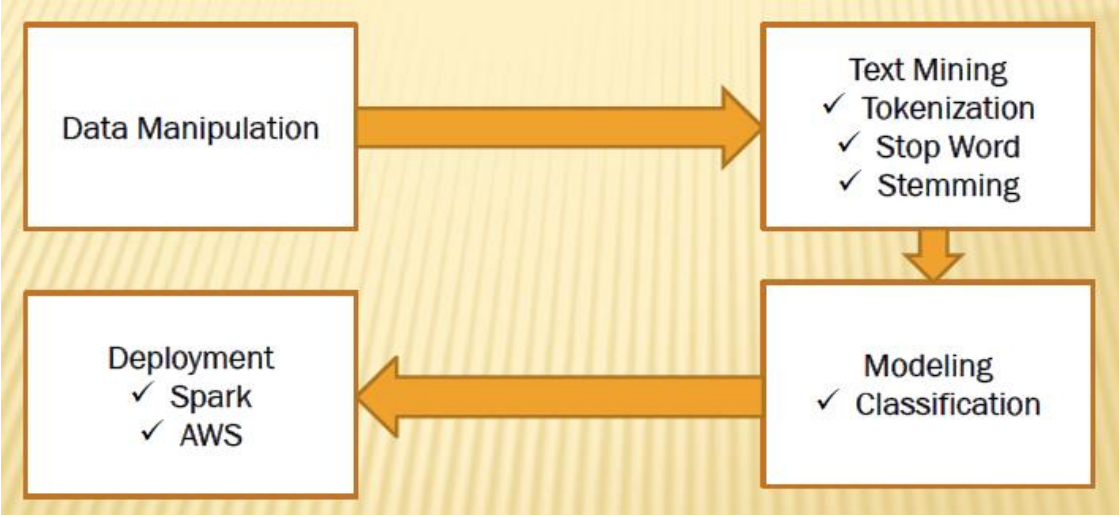

Figure 4: Methodology

3.1 Data Manipulation and Data Mining

As earlier talked, we are filtering the rating having 3 stars so that we can classify the 1 star and 2-star rating as zero and consider it as negative rating and likewise for 3-star and 4-star to be 1 so that we get positive rating.

Text	Target
I have bought sev...	1
"Product arrived ...	0

Figure 5: Removing the record having rating=3 ad converting score to positive and negative feature

The output is showing Top 2 rows of the above figure.

After classifying the rating based on star, now we are classifying the review that is been posted on the product. For this, we will be using text mining. Below the flow diagram of the text mining.

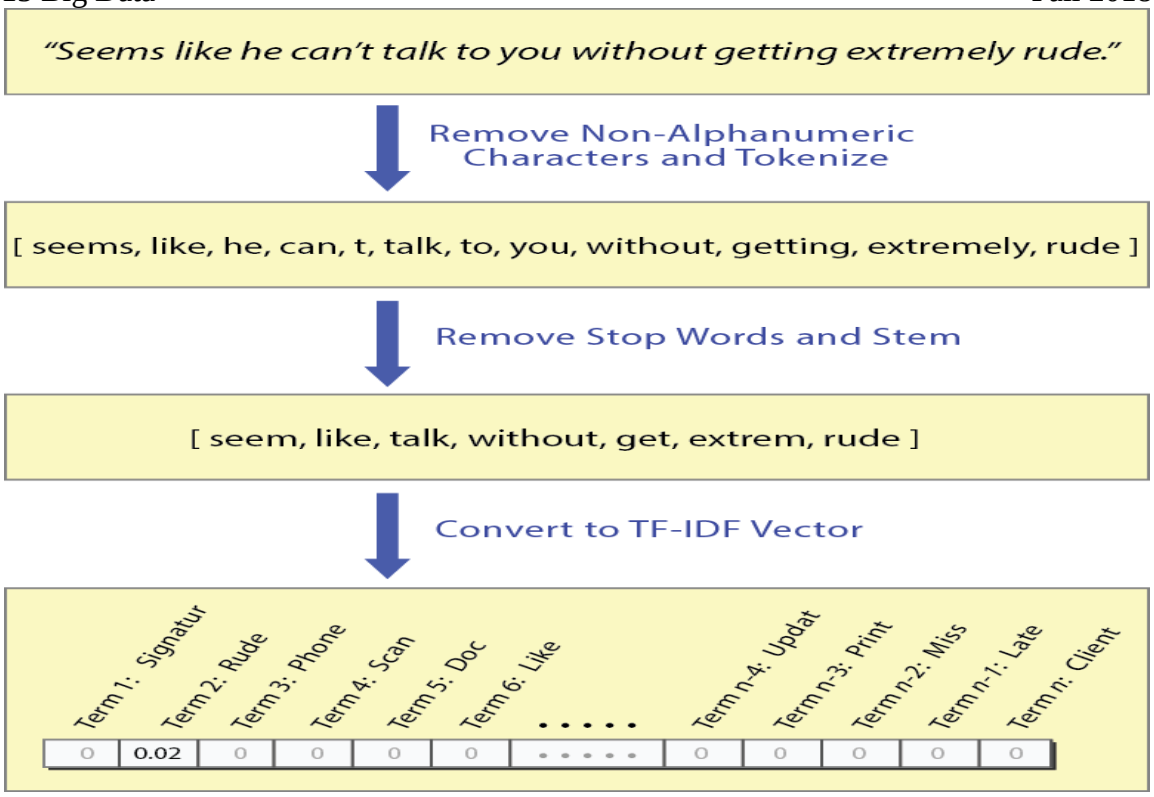

Figure 6: Text mining process

Step 1: Case Normalization

In this step, we make sure that all the words are in lowercase in the review. This can be said as normalizing the case of the review.

```
+--------------------+------+
|          text_lower|Target|
+--------------------+------+
|i have bought sev...|     1|
|"product arrived ...|     0|
+--------------------+------+
only showing top 2 rows
```

Figure 7: Case Normalization is performed on the filtered data

From the above figure the text will be converted into lowercase.

Step 2: Tokenization

In this step, from the word which is in lowercase is been tokenized. So now we have list of tokens having individual words.

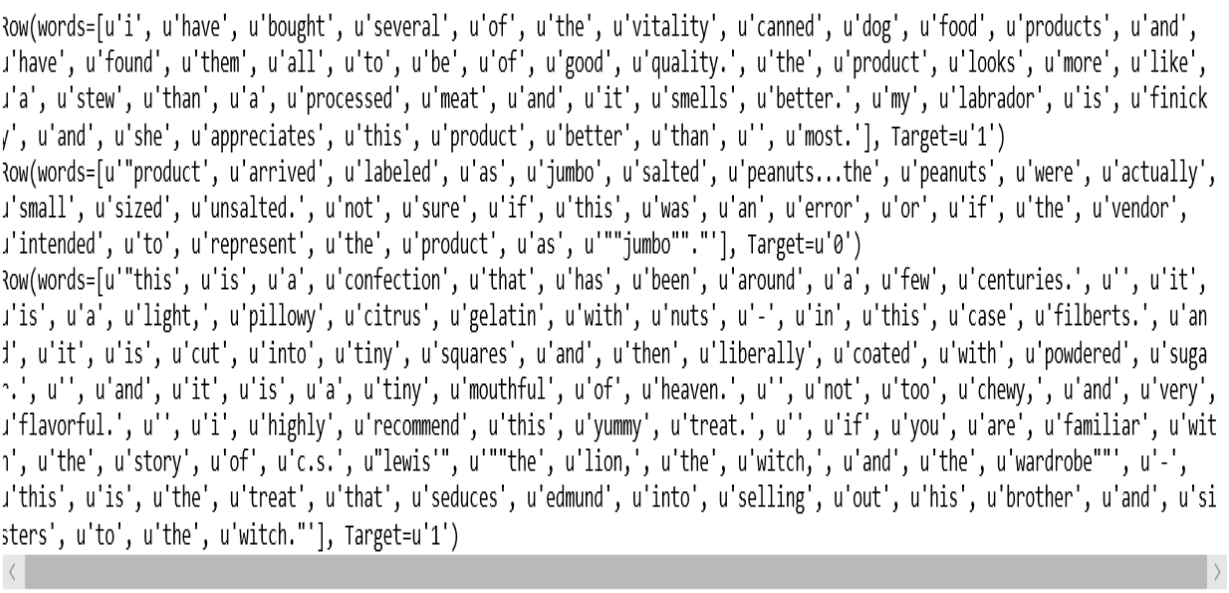

Figure 8: Tokenization

The above figure shows the lower-case words are been tokenized. This is done using "Tokenizer" keyword of python for this project.

Step 3: Removing the Stop words and Stemming

The stop words are the words that are commonly used. Now in this step, in order to get important words from the list made from tokenization, we will use the stop word list and remove the commonly used word. So now we get list of important words.

```
+------+--------------------+
|Target|      words_filtered|
+------+--------------------+
|     1|[bought, several,...|
|     0|["product, arrive...|
+------+--------------------+
only showing top 2 rows
```

Figure 9: stop words removal

In the above figure, we can have shown or printed the top 2 rows that have used "$StopWordsRemoval$".

Stemming

After the removal of stop words, we use Stemming algorithm. This is used to reduce the irregular form of word to common form.

In Stemming, we have $PortStemmer$ method for stemming the text.

Step 4: TF-IDF

In this step, we use TF-IDF in $pyspark$. This is used using sparse metrics. This is used because in the dataset there are null values. This is stored in RDD data sparsity sparse metrics which is efficient place for storage.

$$IDF(t, D) = \log \frac{|D| + 1}{DF(t, D) + 1}$$

$D \rightarrow$ In the dataset no. of review

$DF(t, D) \rightarrow$ Occurrence of words in no. of document

```
root
 |-- text: string (nullable = true)
 |-- label: double (nullable = true)
 |-- words: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- hashing: vector (nullable = true)
 |-- features: vector (nullable = true)
 |-- rawPrediction: vector (nullable = true)
 |-- probability: vector (nullable = true)
 |-- prediction: double (nullable = true)
```

Figure 10: Schema of Prediction Model

In project side, we created the pipeline for tokenization, TF-IDF and Logistic Regression Model and then we transform the model after training it. Thereafter, we printed the schema of prediction dataset as show above. Finally, we received the **Error Rate on Training set** of $\mathbf{0.0218858036078}$.

While working, we came across the python library for sentiment analysis also known as Sentiment Analyzer Library. We used 20000 instances from the data and evaluate sentiment score. This includes sentiment score based on 4 parameters for every parameter which is been evaluated. The 4 parameter includes Compound, Negative, Positive and Negative sentiment score for the text having range of 0 to 1. Below is the output for the Vedar Sentiment Analysis.

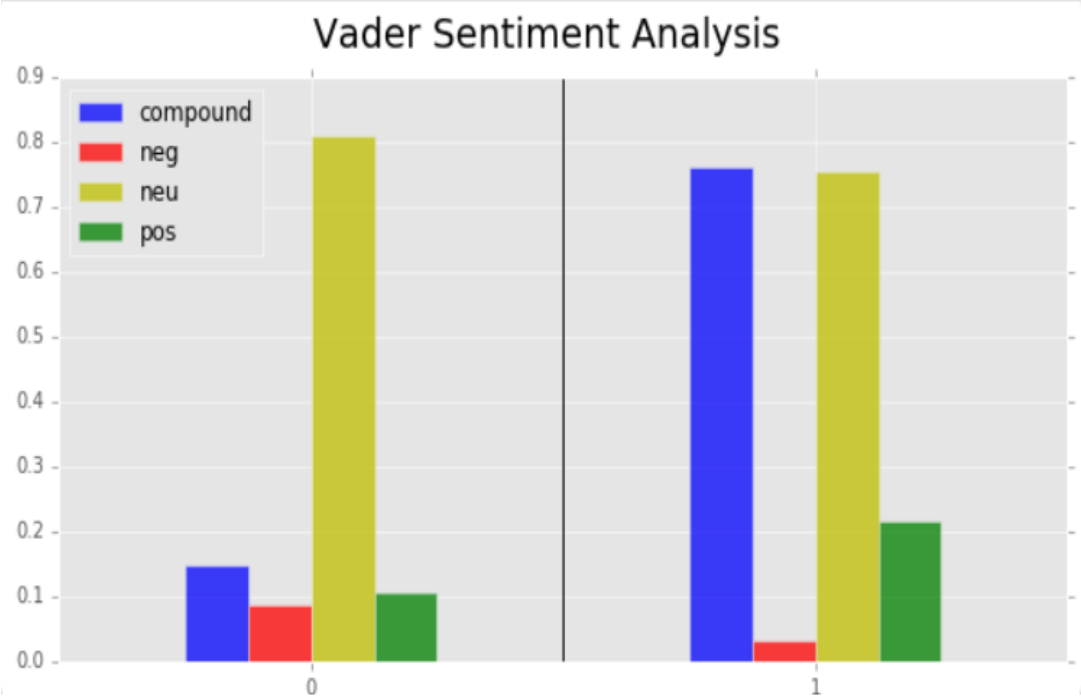

Figure 11: For Positive and Negative comment on different component using Vader aggregate score

The figure 11 above shown is of Vader aggregate score. This contains positive and negative class reviews of different component. Here X-axis is Target.

3.2 Modeling

Model are based on positive and negative prediction. This is based on TF-IDF features and includes algorithm like logistic regression, Naïve Bayes and Gradient Boost Tree. This include accuracy on average basis with 5-fold cross-validation.

3.2.1 Logistic Regression

To optimize the average accuracy 5-fold cross validation, we used grid parameter search to find best value of regularization parameter.

```
Out[69]: [0.9561484487899199,
          0.9471773117678483,
          0.9401087396786587,
          0.9519581876966348,
          0.9062738138953563,
          0.9016377225429577]
```

Figure 12: Accuracy Result of Logistic Regression

For better understanding, we have shown the result in tabular form.

Regularization Parameter	Accuracy
0.001	94.01%
0.01	94.72%
0.1	95.61%
1	95.20%
100	90.63%
200	90.16%

Table 1: Accuracy result of Logistic Regression

Result and Observation:

From the above data or output, we can say that $\mathbf{0.1}$ Regularization parameter of Logistic Regression is having highest accuracy result of $\mathbf{95.61}\%$.

It was observed that when we were succeeding in removing the irrelevant data from the dataset, we achieve better results. This has helped us to know how important the feature engineering is during the classification.

3.2.2 Naïve Bayes

Again using 5-fold cross validation, we calculated the accuracy of Naïve Bayes and the result found was that accuracy is in between $\mathbf{57}\% - \mathbf{58}\%$ which is not better than Logistic Regression.

```
Out[11]: [0.5750640256303141, 0.57489283866258]
```

Figure 13: Screenshot of Accuracy rate of Naïve Bayes

Result and Observation:

From the result, we observation we get the advantage for Naïve Bayes Classification for this dataset:

- The computation of the Naïve Bayes Algorithm is very efficient for the classification purpose.
- Naive Bayes can predict for most of the classification and prediction problem with accurate results.

However, there were few disadvantages that observed for the Naïve Bayes for this dataset are as follow:

- We the dataset was having less amount of data the precision was good. However, as the amount of data increased the precision of the Naïve Bayes Algorithm has deceased gradually.
- Furthermore, we observed for acquiring better result we need proper large amount of data.

3.2.3 Gradient Boost Tree

In this, we again used 5-fold cross validation and found that from the number of trees we did **not received expected accuracy** as it had **memory problem** as **number of trees is increased**.

Result and Observation:

While performing on dataset using the Gradient Boost Tree, we observed that it takes longer time for training.

However, for the given dataset the gradient boost tree was observed that it was prone to overfitting and was not able to extrapolate.

3.3 Deployment

After Modeling, the important step is to deploy the *PySpark* application in Amazon Web Service. This will test the large amount data and will be analyzed. Amazon Web Service (AWS) contains two main components: $S3$ in which data and code will be uploaded and EMR will processed for Spark cluster. The following are the steps for deployment:

Step 1: (Filtering or removing unnecessary library and other data)

Convert iPython notebook to ". py" file. We can remove the python library as it is already existing in AWS. PySpark shell and normal cluster shell are created to run the application to provide two version of application.

Step 2: (Configure AWS Environment)

In AWS, Dataset and other program files are uploaded to $S3$ bucket. The Spark Cluster is created by the AWS with the help of EMR service. The Cluster code name is the used or connected by Putty SSH. Thereafter the data is downloaded on cluster from $S3$ and we get url for all the files which can be used using command "wget" and URL is provided to download the file onto cluster.

Below is the screenshot for Deployment of Spark cluster on AWS:

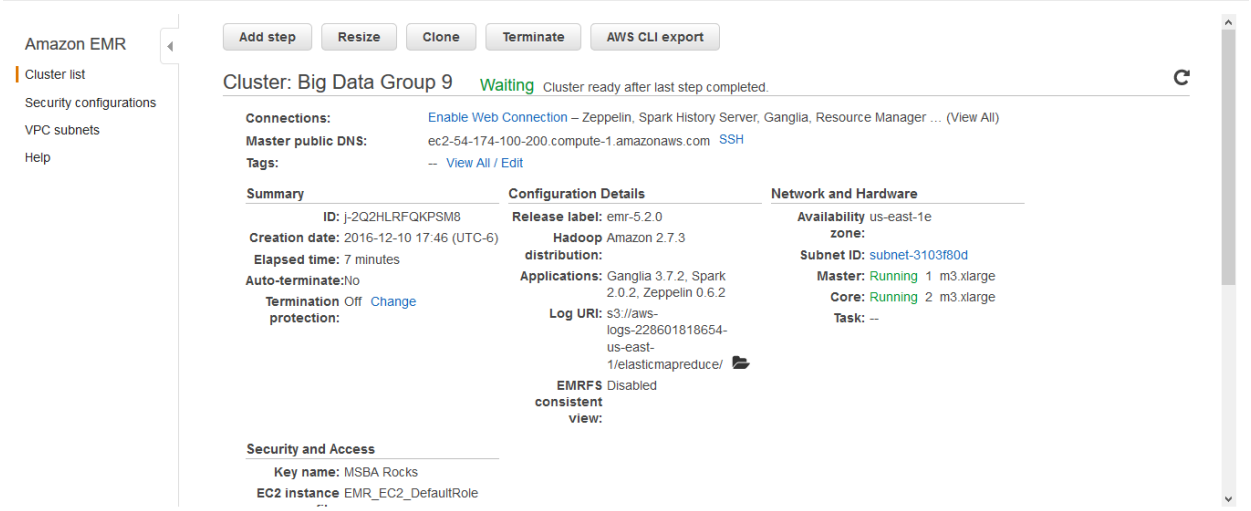

Figure 14: Spark Cluster

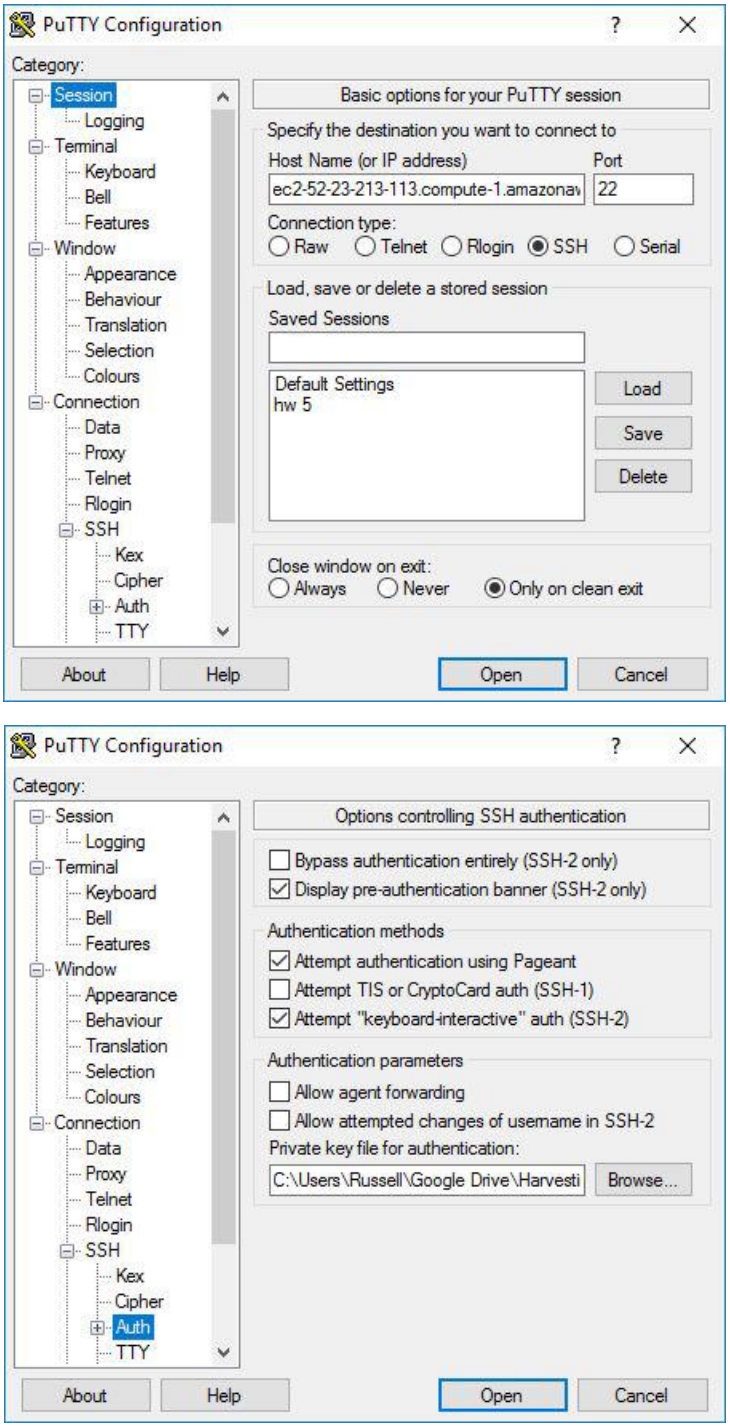

Figure 15: Connecting to Putty SSH

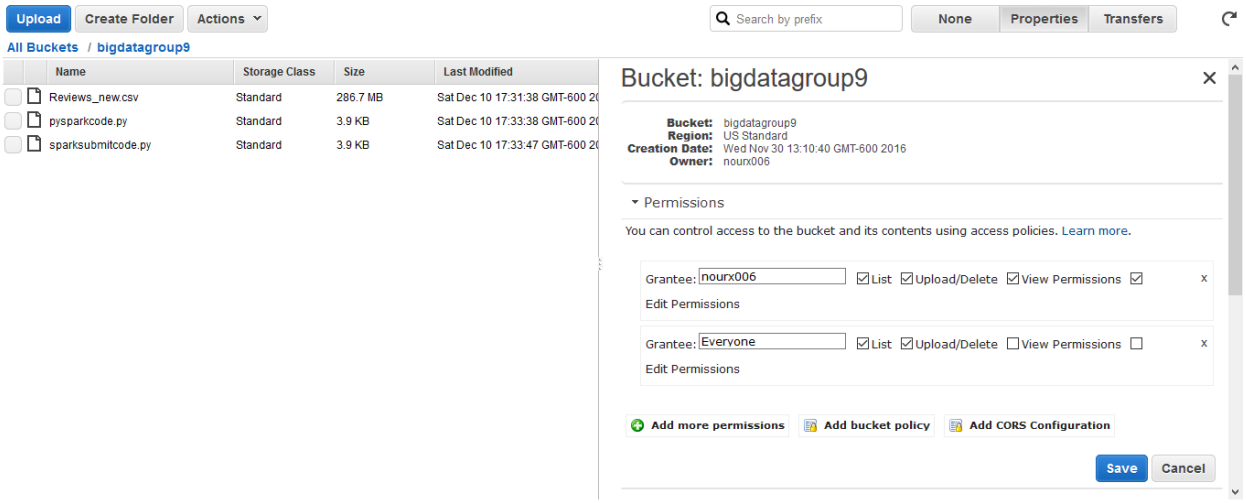

Figure 16: AWS S3

Figure 17: S3 file URL using "wget" in command shell

Figure 18: from cluster node name Downloading the dataset

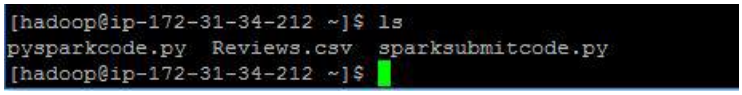

Figure 19: Files on cluster node name

Figure 20: Using Hadoop cluster node for the data

Figure 21: Running the python file in cluster terminal using spark submit

The last part is the executing the python file within the *PySpark* shell.

4 Conclusion

From the experimental analysis we can say that from regularization parameter of 0.1 we logistic regression accuracy of 95.61% and using Naïve Bayes we get accuracy way lesser than logistic regression that is in between $57\% - 58\%$. From the Gradient Boost Tree, we suffered from the memory issue when number of trees increased. It is observed that Logistic Regression is good baseline for measuring the performance of other algorithms.

All the predictive modelling was showcased using Spark and we were able to use it in Data manipulation and preprocessing, data mining i.e. Rating based and Text mining, Predictive model and Deployment in distributed system. Thus, we can say that Spark is useful tool for "Big Data" analysis. This is because we do not have to use any additional package or libraries in the project.

5 Future work

In machine learning classification algorithm, we are having many more classification methods like SVM, Random Forest, etc. that we can use and compare it with Logistic Regression, Naïve Bayes and Gradient Boost Tree. Also, we can use the Apache Storm,Flink, SAS, TIBCO Stream Base, etc. as deployment is tricky, and sometimes may suffer from memory issue.

6 Reference

[1] https://boingboing.net/2018/05/10/caveat-emptor.html

[2] https://www.buzzfeednews.com/article/nicolenguyen/amazon-fake-review-problem

[3] https://yourstory.com/2017/04/common-problems-online-shopping/

[4]C. Zirn, M. Niepert, H. Stuckenschmidt, M. Strube, "Fine Grained Sentimental Analysis with Structural Features", in proceeding of 5th International Joint Conference of National Language Processing, pp. 336-344, 2011.

[5]G. Ganu, N. Elhadad, A. Marian, "Beyond the stars: Improving Rating Predictions using Review Text Context", Twelfth International Workshop on the Web and Database, 2009.

[6]S. Brody, N. Elhadad, "An Unsupervised Aspect-Sentiment Model for Online Reviews", Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL", pp 804-812, 2010.

[7]Y. Jo, A. Oh, "Aspect and Sentiment Unification Model for Review Analysis", WSDM, ACM 978-1-4503-0493-1/11/02, 2011.

[8]L. Qu, G. Ifrim, G. Weikum, "The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns", in Proceeding of the 23rd International Conference on Computational Linguistics, pp. 913-921, 2010.

[9] https://www.kaggle.com/snap/amazon-fine-food-reviews/home, This link contains the dataset that is been used in project.