Sentimental Analysis using Amazon's Electronics Review Dataset

Authors:

Vinaya Chinti

Shivani Ghatge

Course Project Professor's Name:

Dr. Liao

Course Name:

Big Data Essentials

Course Number and Section:

AIT 614

Date:

5/10/2020

Title: Sentimental Analysis using Amazon's electronics review dataset

Abstract

Sentimental analysis is the most renowned method used to analyze the demand and the quality of the product. Natural language processing is an important tool that is used to process the text and can be used in the field of sentimental analysis, text prediction, question answering system, and text summarization. Amazon encourages users to write reviews for the products, they buy. Users rated the product on the rating scale of one to five. These reviews and the rating can be used to do the sentimental analysis and this can help in improving the product quality or even analyzing the demand of the product in the market. Customer satisfaction is the key to the success of the commercial market and this can be achieved using sentimental analysis. In this project, the rating and the review text are used to do the sentimental analysis using Amazon's electronics review dataset. Amazon's electronics review dataset has enormous records and is unstructured so the data was loaded using PySpark. Supervised learning algorithms are applied to the data, as the data is labeled, a comparative analysis of the models trained by using performance metrics, and testing the model on the test dataset.

Keywords: Machine Learning, Big data, PySpark, Sentimental Analysis

Introduction

The increasing use of the internet has resulted in the generation of a huge amount of data. The data is generated through various online resources like social media platforms, online shopping websites, web browsers, and google maps. Today we live in the digital world where we want everything instantly from transferring money online to uploading an image or video. A very large data is created through the activities on the internet. This data can be used to find some meaningful patterns and trends. Nowadays, Big Data is ruling the world. Big Data is used in every field from health care, finance to entertainment industries. Technology industries, as well as digital nonnatives, can benefit by using big data technologies.

Online shopping generates a huge amount of data that can be used for the recommendation systems and sentimental analysis. In this project, a sentimental analysis is done using amazon's electronics review data. Sentimental analysis is used to analyze the sentiments of the user based on the reviews and the rating given by the user. The sentiments are classified into positive, negative, and neutral based on the ratings. Supervised machine learning models - decision tree, logistic regression, Naïve Bayes, and Random Forest are built using the training data, validated using the validation dataset, and tested on the test dataset. The file has may record, PySpark is used for real-time and parallel processing of huge data files. The review text attribute contains the reviews given by the user and the review text is extracted as the feature for training the model. Bag of words and TF-IDF are the features that are extracted using the review text and they are used as features to train the machine learning models.

Related Work

Papers related to sentimental analysis helped us in getting insights about the methodologies used to classify the text and ratings into sentiments. We analyzed various papers to study machine learning algorithms and various data processing tools that can be applied to preprocess the data. Feature extraction is an important step for building machine learning models as they affect the performance of the models to a greater extend. The paper Bo Pang, Lillian Lee, Shivakumar Vaithyanathan[7] has used the movie review dataset, they divided the reviews into positive and negative reviews by using Naïve Bayes and SVM models. They have used unigrams, bigrams, and parts of speech as features for building the model. Their results show that the Naïve Bayes algorithm performed worst in comparison to the SVM model. Callen Rain [8] has used Amazon product review dataset to perform sentimental analysis. The author used Naïve Bayes and decision list algorithms with features - bag of words, sentence length, and parts of speech as features. From the results he claimed that the Naïve Bayes algorithm performed well when compared to the decision list. Yun Xu, Xinhui Wu, Qinxia Wang [9] describes the performance of the perceptron algorithm, multiclass SVM, and Naïve Bayes on the Yelp reviews Dataset. According to the paper, the perceptron algorithm performs well on data when compared to multiclass SVM and Naïve Bayes.

Objectives

1) Use Pyspark to handle the huge data generated from reviews of amazon's electronic products.
2) Preprocess the data and perform exploratory analysis on the data.
3) Splitting the data into training, validation, and test dataset.
4) Selection of relevant features from the data.
5)Use of natural language processing principles for analyzing user reviews like a bag of words or TF-IDF.
6) Train the machine learning model using the training data.
7) Evaluation of the trained models on validation and testing data.
8) Comparative analysis of the machine learning models using performance metrics.

Selected Dataset

Data source: https://nijianmo.github.io/amazon/index.html
The dataset is taken from the source mentioned above. It has reviews of the users for the electronics on the amazon website.

```
elec_json_data.show(5)
```

```
+----------+-----+-------+--------------------+-----------+--------------+----------------+--------------------+--------------------+--------------+--------+----+
|      asin|image|overall|          reviewText| reviewTime|    reviewerID|    reviewerName|               style|             summary|unixReviewTime|verified|vote|
+----------+-----+-------+--------------------+-----------+--------------+----------------+--------------------+--------------------+--------------+--------+----+
|0151004714| null|    5.0|This is the best ...|09 18, 1999| AAP7PPBU72QFM|    D. C. Carrad|[,,,,,,,,  Hardco...|      A star is born|     937612800|    true|  67|
|0151004714| null|    3.0|Pages and pages o...|10 23, 2013|A2E168DTVGE6SV|             Evy|[,,,,,,,,  Kindle...|A stream of consc...|    1382486400|    true|   5|
|0151004714| null|    5.0|This is the kind ...| 09 2, 2008|A1ER5AYS3FQ9O3|           Kcorn|[,,,,,,,,  Paperb...|I'm a huge fan of...|    1220313600|   false|   4|
|0151004714| null|    5.0|What gorgeous lan...| 09 4, 2000|A1T17LMQABMBN5| Caf Girl Writes|[,,,,,,,,  Hardco...|The most beautifu...|     968025600|   false|  13|
|0151004714| null|    3.0|I was taken in by...| 02 4, 2000|A3QHJ0FXK33OBE|W. Shane Schmidt|[,,,,,,,,  Hardco...|A dissenting view...|     949622400|    true|   8|
+----------+-----+-------+--------------------+-----------+--------------+----------------+--------------------+--------------------+--------------+--------+----+
only showing top 5 rows
```

Fig 1: First Five rows of Data Frame

The data has 12 attributes and it has 6739590 rows.

The dataset has the following columns:

1) **reviewerID:** It is the ID of the reviewer eg: A2SUAM1J3GNN3B. The data type of the column is a string.
2) **asin:** ID of the product eg: 0000013714. The data type of the column is string.
3) **reviewerName:** It is the name of the reviewer. The data type of the column is string
4) **votes:** It represents how helpful the rating is. The data type of the column is string
5) **style:** It is the dictionary of the product metadata. The data type of the column is structure
6) **overall:** It is the rating of the product. The data type of the column is double
7) **summary:** It is the summary of the review. The data type of the column is a string
8) **unixReviewTime:** Time of the review (Unix time). The data type of the column is long
9) **reviewtime:** Time of the review(raw). The data type of the column is a string

10) **reviewText:** The review given by the user. The data type of the column is a string.
11) **image:** images that users post after they have received the product. The image is stored as an array.
12) **Verified:** This column is Boolean.

The columns which we have considered for extracting the features are the reviewText and overall

Proposed System

1) System Architecture

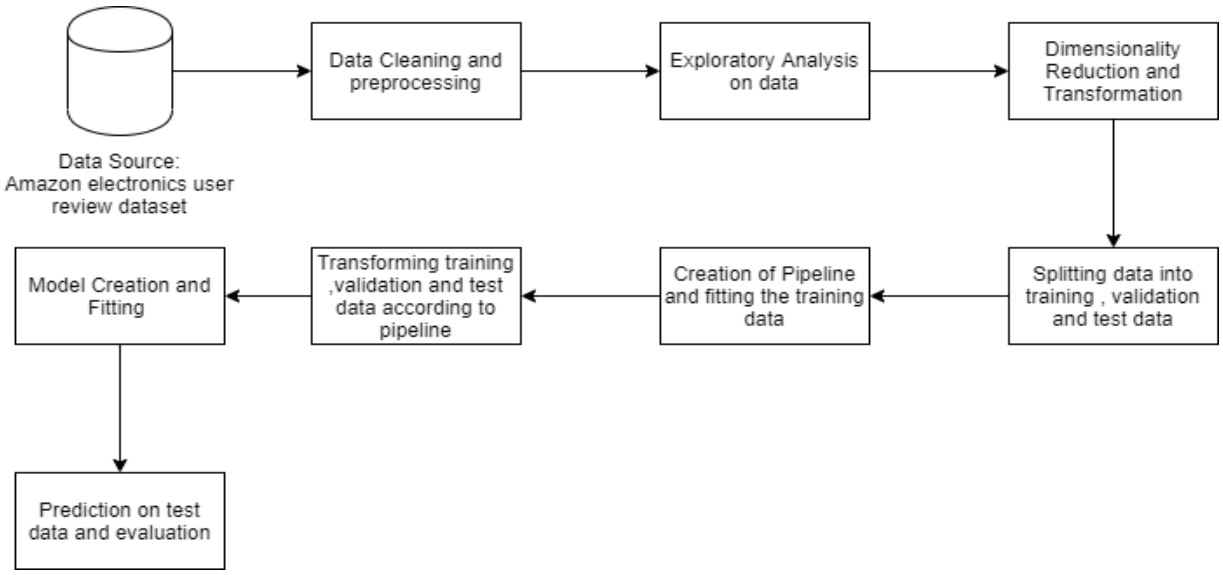


Fig 2: System Architecture

The system comprises of the following components:

1) Data Source: Amazon's electronics user review dataset is used for developing the system. The data is in JSON format and is loaded as a data frame using PySpark.

2) Data Cleaning and Preprocessing: The reviewText column contains the reviews by the user. Stopwords were removed form the reviews and all the text was converted to lower case and tokenized.

3) Exploratory Analysis: It is one of the most important steps which helps in getting the insights into data. Following graphs were developed to gain insights into the loaded dataset.

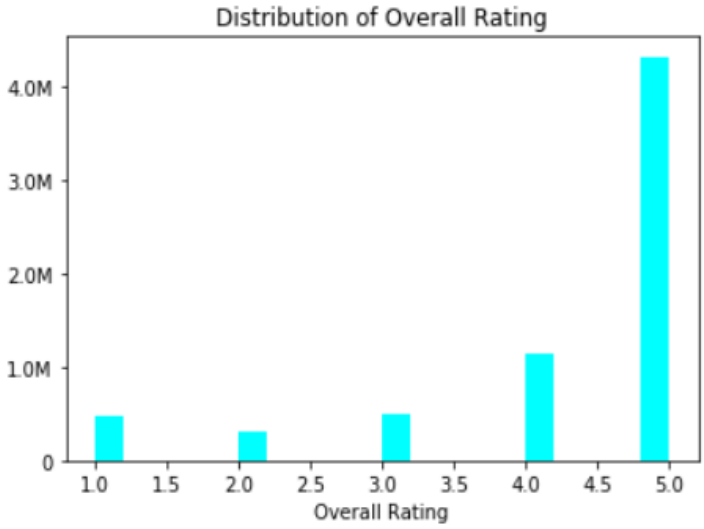


Fig 3: Distribution of Overall rating

Fig 3 shows the distribution of the ratings on the dataset. The graph shows that there are more products with ratings 5.

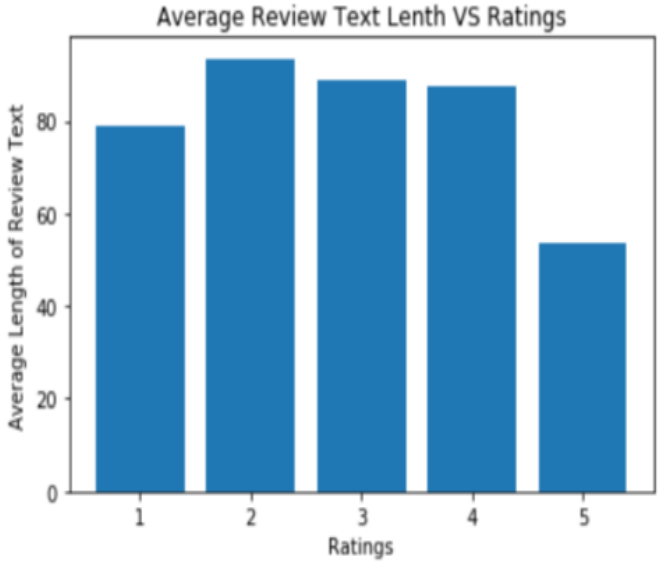


Fig 4: Average Review Text Length Vs Ratings

Fig 4 plots the relationship between the rating and the length of the review. From the graph, we understand that the length of the reviews is less when the rating is high. People tend to write more when the product is bad and they give a low rating.

Fig 5: Growth in the number of reviews over months

Fig 5 plots the relationship between the number of reviews and months. It shows that the number of reviews is more in January and December as compared to other months. To conclude, it can be said that people tend to do more shopping during Christmas and new year, this is the reason behind more reviews in January and December.

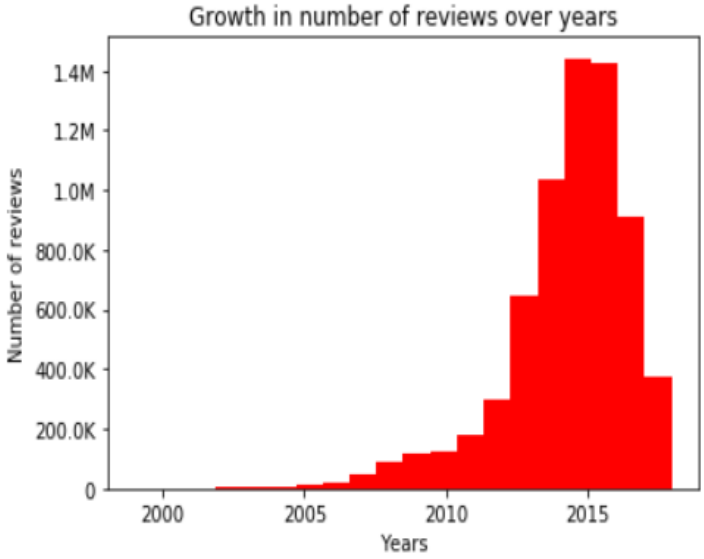


Fig 6: Growth in the number of reviews over the years

Fig 6 shows the growth in the number of reviews over the years. The importance of reviews has increased over the years.

```
from pyspark.sql.functions import col
print("Top 10 reviewed products:")
elec_json_data.groupBy("asin").count().sort(col("count").desc()).show(10)
```

```
Top 10 reviewed products:
+----------+-----+
|      asin|count|
+----------+-----+
|B003L1ZYYW| 8617|
|B0019HL8Q8| 8160|
|B0019EHU8G| 7777|
|B0015DYMVO| 7380|
|B000VS4HDM| 6802|
|B00DIF2BO2| 6226|
|B00M55C0NS| 6219|
|B00BWF5U0M| 6184|
|B00IVPU7AO| 5987|
|B000BQ7GW8| 5985|
+----------+-----+
only showing top 10 rows
```

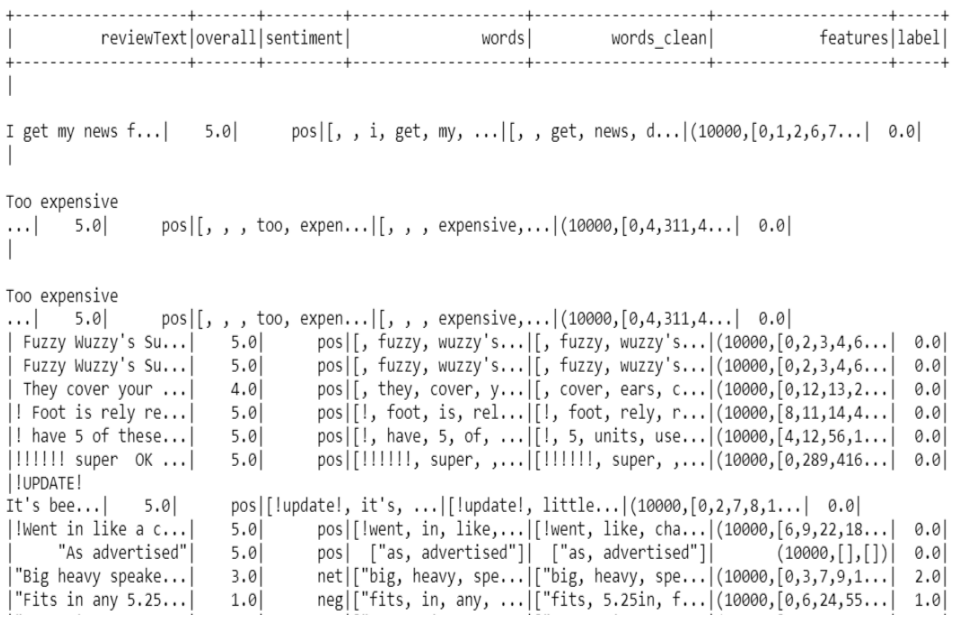
Fig 7: Top 10 products having maximum reviews

Fig 7 shows the top 10 electronics which has received the maximum rating

4) Dimensionality Reduction and Transformation: The overall column of the dataset is used to form a new column called “Sentiment”. If the value of the overall column is greater than three then the sentiment is positive and the value of the sentiment column is pos. If the value of the overall column is less than 3 then the sentiment is negative and the value of the sentiment column is neg and if it is equal to three then the value of sentiment column is net which is a neutral sentiment.

5) Splitting the dataset into training, validation, and testing dataset: The entire dataset is divided into training, validation, and testing datasets. 70% of the data is used for training, 20% for validation, and 10% for testing. The model is trained using the training dataset, validated using the validation dataset, and tested using the test dataset.

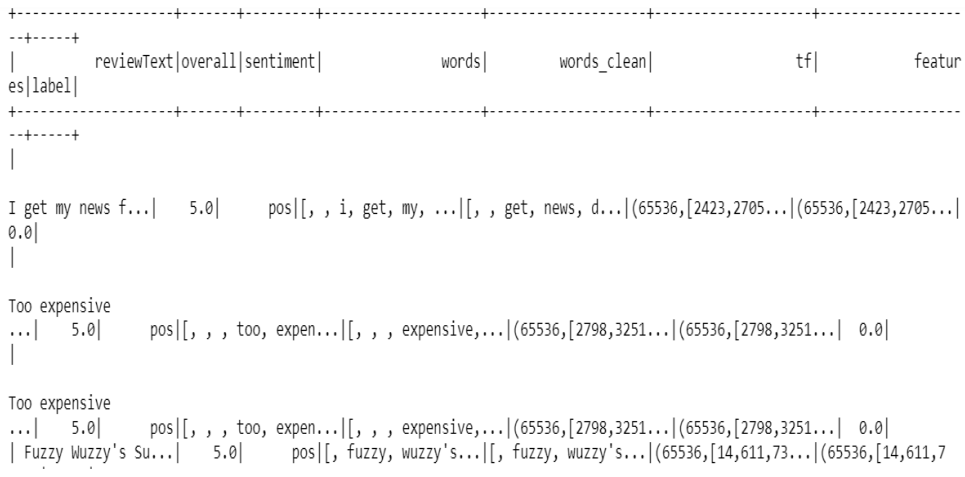
6) Creation of pipeline and fitting the training data: From the dataset, only two columns namely the reviewText and overall are used for sentimental analysis. Features are extracted from the reviewText column. The features that we have used is a bag of words and TF-IDF. words_clean contains the tokens and tf contains the value of term frequency. When the pipeline is applied to the data we get a new column called features that contains the bag of words. Two pipelines are used on the dataset one with a bag of words as the feature and second with TF-IDF as the feature. In the case of TF-IDF, the feature column contains the TF-IDF values of the text.



reviewText	overall	sentiment	words	words_clean	features	label
I get my news f...	5.0	pos	[, , i, get, my, ...	[, , get, news, d...	(10000,[0,1,2,6,7...	0.0
Too expensive ...	5.0	pos	[, , , too, expen...	[, , , expensive,...	(10000,[0,4,311,4...	0.0
Too expensive ...	5.0	pos	[, , , too, expen...	[, , , expensive,...	(10000,[0,4,311,4...	0.0
Fuzzy Wuzzy's Su...	5.0	pos	[, fuzzy, wuzzy's...	[, fuzzy, wuzzy's...	(10000,[0,2,3,4,6...	0.0
Fuzzy Wuzzy's Su...	5.0	pos	[, fuzzy, wuzzy's...	[, fuzzy, wuzzy's...	(10000,[0,2,3,4,6...	0.0
They cover your ...	4.0	pos	[, they, cover, y...	[, cover, ears, c...	(10000,[0,12,13,2...	0.0
! Foot is rely re...	5.0	pos	[!, foot, is, rel...	[!, foot, rely, r...	(10000,[8,11,14,4...	0.0
! have 5 of these...	5.0	pos	[!, have, 5, of, ...	[!, 5, units, use...	(10000,[4,12,56,1...	0.0
!!!!!! super OK ...	5.0	pos	[!!!!!!, super, ,...	[!!!!!!, super, ,...	(10000,[0,289,416...	0.0
!UPDATE! It's bee...	5.0	pos	[!update!, it's, ...	[!update!, little...	(10000,[0,2,7,8,1...	0.0
!Went in like a c...	5.0	pos	[!went, in, like,...	[!went, like, cha...	(10000,[6,9,22,18...	0.0
"As advertised"	5.0	pos	["as, advertised"]	["as, advertised"]	(10000,[],[])	0.0
"Big heavy speake...	3.0	net	["big, heavy, spe...	["big, heavy, spe...	(10000,[0,3,7,9,1...	2.0
"Fits in any 5.25...	1.0	neg	["fits, in, any, ...	["fits, 5.25in, f...	(10000,[0,6,24,55...	1.0

Fig 8: Dataframe for the bag of words

Fig 8 shows the words_clean column which has the tokens and the features column which has the bag of word feature.



reviewText	overall	sentiment	words	words_clean	tf	features	label
I get my news f...	5.0	pos	[, , i, get, my, ...	[, , get, news, d...	(65536,[2423,2705...	(65536,[2423,2705...	0.0
Too expensive ...	5.0	pos	[, , , too, expen...	[, , , expensive,...	(65536,[2798,3251...	(65536,[2798,3251...	0.0
Too expensive ...	5.0	pos	[, , , too, expen...	[, , , expensive,...	(65536,[2798,3251...	(65536,[2798,3251...	0.0
Fuzzy Wuzzy's Su...	5.0	pos	[, fuzzy, wuzzy's...	[, fuzzy, wuzzy's...	(65536,[14,611,73...	(65536,[14,611,7	

Fig 9: Dataframe for TF-IDF

Fig 9 shows the data frame which has tf and feature columns. The tf column contains the term frequency whereas the features column contains the TF-IDF values.

7) Transforming training, validation, and testing data according to the pipeline: The pipeline formed is applied to the training, validation, and testing datasets, so that the model can be trained and tested using the features formed.

8) Model Creation and fitting: Supervised algorithms -logistic regression, Naïve Bayes, Decision Tree, and Random Forest are developed using various hyperparameters. The developed models are then trained using the training dataset.

9) Prediction on the test data and evaluation: The models built are tested on the test dataset and are evaluated by using performance metrics.

2) NLP and Data Analytic Approaches

The text is processed to form tokens and the stop words from the text are removed. Following existing algorithms are used for developing the system:

1) Logistic Regression: It is used when the dependent variable is categorical. It predicts the class of a test sample by using the probabilities. It does not perform well when the independent variables are not correlated to the dependent variables. Therefore, feature extraction plays a very important role in developing a logistic regression model. The correlation of the dependent and the independent variables needs to be checked so that the model fits on the data well. It requires fewer computations and does not require the data to be scaled. There are the following three types of logistic regression models:

- **Binary Logistic Regression:** The dependent categorical variable is binary.
- **Multinomial Logistic Regression:** The dependent categorical variable can have multiple values.
- **Ordinal Logistic Regression:** The dependent categorical variable can have multiple values and the order of the values also matter. For example the ratings of the movie.

2) Decision Tree: It has a flowchart like structure with a root node, internal nodes, and the leaf nodes. The classification is done by sorting the instances down the trees starting from the root node and the leaf nodes provide the classification labels. The path down the tree is chosen by using various splitting criteria - entropy, Gini index, or maximum error. They are computationally expensive, as we need to compute the value at each split to decide which path to take down the tree.

3) Random Forest: Random forest is a variant of the decision tree algorithm. It is an ensemble learning method. It consists of multiple individual decision trees that have a low correlation between them. Each tree predicts the class label and the class with the most votes becomes the model’s prediction. Random forest gives better accuracy as compared to the decision tree because even if some trees are wrong, the majority of them will be correct and as a group, they will predict the class label correctly.

4) Naïve Bayes: It is widely used for classification tasks and is a probabilistic algorithm. It is based on the Bayes algorithm. Naïve Bayes has the following 3 types:

- **Multinomial Naïve Bayes:** It is most commonly used for document classification. It uses the frequency of the words in the document as features.
- **Bernoulli Naïve Bayes:** The depenent variable is binary in case of Bernoulli naives bayes.
- **Gaussian Naïve Bayes:** It is used when the dependent variable is continuous and not discrete.

Following features are used for training the models:

1) Bag of words: To use a bag of words as features, the sentence is tokenized. It gives the vector representation of the sentences so that it can be given as features for training the algorithm. We calculate the frequency of all the words in the document and the words are then replaced by their frequencies which results in the vector representation of the documents.

2) TF-IDF: It is a technique used to give weights to the words in the document which specifies the importance of the word in the document. Term frequency calculates the frequency of the words in the document. IDF is the inverse of the document frequency and it measures the frequency of a word in a set of documents.

3) New Algorithm

Following steps are used in developing the sentimental analysis
1) Data is loaded into the data frame from JSON using PySpark.
2) reviewText and overall columns are used for feature extraction.
3) Bag of words and TF-IDF features are extracted from the text.
4) Pipeline is built and the models are trained using first by using Bag of words features and then by using TF-IDF features.
5) The trained model is tested on the testing data and comparative analysis is done using the performance metrics obtained from various models built.

4) Hardware and Software Development Platforms:

- **Hardware:** Laptops with configuration Intel I7 8th gen processor, 512 SSD and 16 GB RAM.
- **Software:** Jupyter Notebook
- **Libraries:** PySpark, Numpy, NLTK, MlLib, Matplotlib

Experimental Results and Analysis

The following table shows the accuracy, F1-score, precision, and recall obtained after validating and testing the trained models on the validation and the testing data.

	Accuracy	F1-Score	Precision	Recall
Logistic Regression Using TF-IDF				
Validation Data	86.45%	84.28%	83.54%	86.45%
Test Data	86.49%	84.35%	83.63%	86.49%
Logistic Regression Using Bag of words				
Validation Data	86.59%	84.01%	83.54%	86.59%
Test Data	86.64%	84.08%	83.63%	86.64%
Naïve Bayes using TF-IDF				
Validation Data	78.25%	80.48%	83.92%	78.25%
Test Data	78.24%	80.48%	83.93%	78.24%
Naïve Bayes using Bag of Words				
Validation Data	83.22%	82.69%	82.22%	83.22 %
Test Data	83.24%	82.71%	82.25%	83.24%
Random Forest using TF-IDF				
Validation Data	81.06%	72.58%	65.71%	81.06%
Test Data	81.10%	72.64%	65.78%	81.10%
Random Forest using Bag of Words				
Validation Data	81.06%	72.58%	65.71%	81.06%
Test Data	81.10%	72.64%	65.78%	81.10%
Decision Tree using TF-IDF				
Validation Data	81.77%	74.94%	73.58%	81.77%
Test Data	81.78%	74.95%	73.50%	81.78%
Decision Tree using Bag of Words				
Validation Data	81.77%	74.95%	73.59%	81.77%
Test Data	81.78%	74.96%	73.52%	81.78%

Table 1: Evaluation metrics for the trained models

Lessons Learned

With the increasing use of online services and the internet enormous amount of data is produced. The data is of no use unless it is cleaned and insights are drawn from the data. According to statistics about 80% of the data produced is unstructured and requires tools other than relational databases to handle it. The key takeaways for this project are natural language processing, use of Pyspark for handling big data, and machine learning. Natural language processing taught us basic preprocessing steps such as tokenization and stop word removal. We learned the process of extracting features from the data. The concepts of the bag of words, TF-IDF, Naïve Bayes, Decision tree, Random forest, and logistic regression were studied and implemented using the MlLib library. MlLib is the machine learning library used for data frame loaded using PySpark.

Future Work

The models were built using only Amazon's electronics review dataset. To train the algorithm on a large corpus of data, all the amazon reviews for all the categories can be merged. This will help in improving the accuracy of the models as well. For the handling of big data, cloud technologies can be used -AWS and Microsoft Azure. Features such as ngrams, parts of speech, the sentence length can be used to train the model to increase the accuracy further. Neural network models like Long Short Term Memory can be built and skip-gram or glove embedding can be used to convert the sentences to vector representations.

Conclusion

From the accuracies calculated it can be observed that logistic regression has performed well on the validation and test data with both bag of words and TF-IDF features. Random forst and decision trees have almost the same accuracies on test and validation data irrespective of the features. Naïve Bayes algorithm has performed better on the data when the bag of words are used as features as compared to TF-IDF. Decision trees and random forest take longer time for training. Logistic Regression takes least time for training and is the most efficient.

References

[1] Decision Tree. (2019, April 17). Retrieved May 4, 2020, from https://www.geeksforgeeks.org/decision-tree/

[2]Gandhi, R. (2018, May 17). Naive Bayes Classifier. Retrieved May 4, 2020, from https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[3]Koehrsen, W. (2017, December 27). Random Forest Simple Explanation. Retrieved May 4, 2020, from https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d

[4] Chauhan, N. S. (2020, April 18). Real world implementation of Logistic Regression. Retrieved May 4, 2020, from https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125

[5] Sentiment Analysis: How Does It Work? Why Should We Use It? (n.d.). Retrieved May 4, 2020, from https://www.brandwatch.com/blog/understanding-sentiment-analysis/

[6] Dumbleton, R. (2020, March 9). Sentiment Analysis: Definition, Uses, Examples Pros /Cons. Retrieved May 4, 2020, from https://getthematic.com/insights/sentiment-analysis/

[7] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, 2002. Thumbs Up? Sentiment Classification using machine learning techniques. Available http://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

[8] Callen Rain. Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning. Available https://www.sccs.swarthmore.edu/users/15/crain1/files/NLP_Final_Project.pdf

[9] Yun Xu, Xinhui Wu, Qinxia Wang. Sentiment Analysis of Yelp‘s Ratings Based on Text Reviews. Available http://cs229.stanford.edu/proj2014/Yun%20Xu,%20Xinhui%20Wu,%20Qinxia%20Wang,%20Sentiment%20Analysis%20of%20Yelp's%20Ratings%20Based%20on%20Text%20Reviews.pdf

[10] Scott, W. (2019, May 21). TF-IDF for Document Ranking from scratch in python on real world dataset. Retrieved May 9, 2020, from https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089