|  |  |
| --- | --- |
| Centennial College logo | **School of Engineering Technology and Applied Science**  ***Information and Communication Engineering Technology***  **Natural Language & Recommender Systems (COMP262-401)**  **Project Report: Phase 2**  **Team:**  **Ari Cerrahyan (301156275)**  **Gordon Stevens (300864022)** |

Table of Contents

[Introduction 2](#_Toc132373486)

[Project Phase 1 2](#_Toc132373487)

[Dataset information 2](#_Toc132373488)

[Part 1: Dataset data exploration 3](#_Toc132373489)

[Part 2: Text preprocessing 9](#_Toc132373490)

[Part 3: Text representation: TF-IDF, Bag-of-Words, and spaCy 9](#_Toc132373491)

[Part 4: Modeling (sentiment analysis) lexicon approach 10](#_Toc132373492)

[Part 5: Validation and comparing results of VADR and SWN sentiment modelling 11](#_Toc132373493)

[Project Phase 2 12](#_Toc132373494)

[Part 9: Modelling (Sentiment Analysis) machine learning approach 12](#_Toc132373495)

[Part 10: Results of the training process 12](#_Toc132373496)

[Part 12: Comparison of the results of the lexicon models and the machine learning models 14](#_Toc132373497)

[Part 13: Review research paper and examine options for improvement 16](#_Toc132373498)

[References 20](#_Toc132373499)

# Introduction

This report is part of a 2-phase project, using Amazon text-based reviews on musical instruments, this project performs sentiment analysis on the dataset.

In the first phase, it is divided up into 7 parts: dataset exploration, text preprocessing, text representation, modelling using valence aware dictionary and sentiment reasoner (VADR) and SentiWordNet analysis, the two models are compared, and then this report is built.

The second phase involved machine learning approaches to modelling, and final conclusions are that using the TFIDF data with Support Vector Machine model had the best statistics with 0.59 accuracy, 0.59 precision, 0.59 recall, and 0.53 F1. This is surprising as this yielded higher statistics than VADR and SentiWordNet.

# Project Phase 1

## Dataset information

The dataset used in this project was sourced from: <http://jmcauley.ucsd.edu/data/amazon/>. It is supplied by Julian McAuley of the University of California San Diego (UCSD).

Information provided by the author:

This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014.

This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

The dataset is "Musical Instruments: 5-core" sourced directly from: http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews\_Musical\_Instruments\_5.json.gz

After ungzipping, the file *Musical\_Instruments\_5.json* is acquired. The JSON file is used directly with pandas to build the initial dataframe. The gzip file is not used in favour of speed, and is hosted remotely in a GitHub public repository for ease of use and submission.

Feature information:

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B

- asin - ID of the product, e.g. 0000013714

- reviewerName - name of the reviewer

- helpful - helpfulness rating of the review, e.g. 2/3

- reviewText - text of the review

- overall - rating of the product

- summary - summary of the review

- unixReviewTime - time of the review (unix time)

- reviewTime - time of the review (raw)

## Part 1: Dataset data exploration

After loading the dataset from a JSON file, we can view the first 9 records for clarity in figure 1.

Text

Description automatically generated

Figure 1: First 9 records of the dataset after import from JSON source file

In figure 2, the pandas dataframe is slightly over 7mb making this ideal for quick loading and analysis, contains 9 columns, 7 columns have datatype object, and 2 columns are integer datatypes. The column “reviewerName” is missing values, therefore this is considered later in the code.

A black screen with white text

Description automatically generated with low confidence

Figure 2: Pandas dataframe column information

It is found that there are textual reviews, as well as ratings with a scale of 1 to 5. The first review on the musical instruments was posted on September 18, 2004 to July 22, 2014, for a total of 3594 days, or just under a 10 year duration.

The dataset is then modified to add new columns, review text lengths (discussed later in the report), helpful rating, the review year and review month. The review year and month are created for easy access in visualisations and performance when building graphs.

The original helpful column is a mix of positive and negative votes on the actual review itself, for other visitors to decide if the review textual matter is helpful or not. This would normally be represented with a question on the webpage near the review text with thumbs-up and thumbs-down icons. In the actual dataset, this is represented by two integers in brackets, for example [13, 14], in which 13 people said the review text was helpful, but 14 people said the review was not helpful. In the helpful rating column, this is calculated into a decimal format.

In figure 3, it is shown that for this dataset, only 7 reviews were posted in 2004, however these reviews mounted over time, and there were 350, 1007, 1936, 4055, and 2679 in 2010, 2011, 2012, 2013, and 2014 respectively.

Chart, histogram

Description automatically generated

Figure 3: Number of ratings given by year

In figure 4, it can be seen that the reviews seem to trend towards the holiday shopping seasons, and while not a direct confirmation of this without more data, December is first month where the instrument and early reviews may be posted, with January as just after the holiday, and February and March as later reviews, after which, the reviews appear to drop in a step-wise fashion into a valley before peaking up for the new holiday season cycle.

Chart, bar chart

Description automatically generated

Figure 4: Number of ratings given by month

The Seaborn pairplot in figure 5 shows the above data again, however the review text length is now displayed.

Calendar

Description automatically generated

Figure 5: Seaborn pairplot

Upon further analysis, it is found that 3046 out of 10261 reviews (29.7%) have helpfulness ratings. Review text length generally less than 5000 characters. In figure 6, Distributions of ratings from 1 to 5 is left-skewed, no zero-star reviews present. The skewness indicates a general overall 4-to-5-star satisfaction level as 1, 2, and 3 ratings add up to 1239 ratings, while 4's are 2084 and 5's are 6938. 4-ratings are more than 1, 2, and 3 ratings combined. The lowest quantity of review ratings is 1, with 217 ratings. People are generally happy with the musical instruments.

Chart, bar chart

Description automatically generated

Figure 6: Number of ratings given

In figure 7, the most reviewed item, “B003VWJ2K8” has 163 reviews. In figure 8, the least reviewed item, “1384719342” has 5 reviews. For the most reviewed item, 134 gave it a rating of 5, and 18 a rating of 4. Ratings of 3 and 1 were given by 5 people and a rating of 2 by only one review. Therefore, this item was quite popular. There are 900 unique products reviewed.

Text

Description automatically generated

Figure 7: Most reviewed item “B003VWJ2K8”

A screen shot of a computer

Description automatically generated with low confidence

Figure 8: Least reviewed item “1384719342”

The number of reviews were 900, with an individual product mean of 11.40 reviews, standard deviation of 12.93 reviews, minimum of 5 reviews, maximum of 163 reviews, and 25% at 6.0 reviews, 50% at 8.0 reviews, and 75% at 12.0 reviews. Therefore, statistically there are significant outliers, which may influence the analysis, however due to the nature and significance of the data, the outliers will be left in the dataset for the sentiment analysis.

There were a total of 1429 users participating in posting reviews within the dataset. In figure 9, it shows that most reviews are under 2000 characters. The shortest length review is 0 characters. The longest length review is 11310 characters. The mean number of characters in review text length is 485.9 characters. The standard deviation of number of characters in reviews is 613.43, there are reviews with zero characters (just number-value ratings), the longest review has a length of 11310 characters, 25% at 162 characters, 50% at 284 characters and 75% at 552 characters. There are 3 reviews with over 9000 characters. This indicates that reviews are generally around 500 characters, however there are statistical outliers once again.

Chart, histogram

Description automatically generated

Figure 9: Number of characters in reviews

## Part 2: Text preprocessing

Next a sample of 1000 reviews is taken, 200 from each rating level (1 to 5). Importantly, these samples are taken in repeatable way, and will sample the same 1000 reviews each time the notebook is run, as a random seed is set. Next, a new column “overall” in the dataset is created in which reviews with ratings of 4 and 5 are considered positive, 3 is considered neutral, and ratings of 1 or 2 are considered as negative. This 1000 reviews dataset contains 400 positive, 200 neutral, and 400 negative reviews.

This dataset has the following columns dropped:

Removed the following columns:

- reviewerName: this column has 27 missing values, as well the names of reviewers is not helpful in predicting sentiment or in machine learning.

- helpful: not needed in NLP analysis, contains the helpfulness of the review.

- reviewTextLength: not needed for sentiment analysis, was created for other analysis in step 1.

- unixReviewTime, reviewTime, reviewYear, reviewMonth: contains time information, and while useful in dataset exploration, but is not required for lexicon approaches as it is rule-based and does not have connections with data and time for sentiment analysis. Not useful in the machine learning approach either.

This leaves the dataset with reviewerID, asin, reviewText, overall, summary, and overallRating.

Next, the reviewText is edited by lowercasing the text, and the punctuation is removed. Next the reviewText is lemmatized. Lemmatisation is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatisation depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. In short, lemmatization converts conjugated or other forms of a word into the base word, increases model understanding and decreases size of corpus, therefore increases efficiency.

Next, training and testing datasets are created in a 70%/30% split, stratified upon the rating value, with a repeatable random seed set. This represents 280 positive and negative reviews, and 140 neutral reviews within the training dataset. The testing dataset contains 120 positive, and negative reviews, with 60 neutral reviews. This is very respectable number of reviews and serve the analysis well.

## Part 3: Text representation: TF-IDF, Bag-of-Words, and spaCy

TF-IDF (Term-Frequency-Inverse-Document-Frequency) represents the text corpus in numerical, matrix format, columns are unique words, rows are text of all instances. TF looks at the number of occurrences of a term within the document, IDF looks at the commonness within the corpus. TF-IDF does not understand context, but bag-of-words does. The package spaCy uses a pretrained package, in this case the code opts for the medium package. The reviewText is processed by lowercasing, removing punctuation, removing digits, and striping whitespace, the text is tokenized, stop words removed, and then run through the spaCy model. In figure 10, we can see the reviewText transfomed into reviewVectorized. For phase 2, we will choose TF-IDF as it considers the importance of each word in the corpus, where bag-of-words and spaCy do not.

Text

Description automatically generated

Figure 10: reviewText to reviewVectorized

## Part 4: Modeling (sentiment analysis) lexicon approach

In part 4, modeling, Valence Aware Dictionary and Sentiment Reasoner (VADR) is used to create scores, the compound score and the overall sentiment.

Text

Description automatically generated with medium confidence

Figure 11: Results of VADR processing in the dataset

Results for this model:

Valence Aware Dictionary and Sentiment Reasoner (“VADR”) modelling

Accuracy: 0.52857

Precision: 0.44286

Recall: 0.52857

F1: 0.39187

These numbers are low. VADR is a sentiment analysis model which balances the positive or negative nature of the text, as well as the perceived strength of the text. This includes analysis on negations, contractions, informal language, emoticons, and punctuation.

Next, SentiWordNet (“SWN”) is used for sentiment analysis modelling, which produced a sentiment score in decimal form, and a description if it is positive, negative, or neutral.

A picture containing chart

Description automatically generated

Figure 12: Results of SentiWordNet processing in the dataset

The model produced undesirable results:

Accuracy: 0.25714

Precision: 0.31548

Recall: 0.29909

F1: 0.24551

SWN uses the WordNet database as a structure or basis for its words and assesses sentimentality. The dataset is preprocessed removing stopwords and punctuation, and parts of speech (“POS”) are determined, using WordNet. Using the POS tagging, pos\_score() and neg\_score() methods are used to determine sentiment and is totaled up.

## Part 5: Validation and comparing results of VADR and SWN sentiment modelling

A black screen with white text

Description automatically generated with medium confidence

Figure 13: Results of VADR and SentiWordNet

As displayed in figure 13, the clear winner was VADR, likely because it has a more rigorous analysis model, as described in the report. Phase 2 will introduce machine learning methods for modelling, which will give a more compelling and interesting set of results.

# Project Phase 2

## Part 9: Modelling (Sentiment Analysis) machine learning approach

In the documentation for this project, it asks to complete two approaches, however all mentioned approaches were taken for good measure, as not to exclude any interesting results which may arise. From phase 2 as mentioned previously, TF-IDF was chosen as it considers the importance of each word in the corpus, where bag-of-words and spaCy do not. The following models were selected: Logistic regression, Support Vector Machines (SVM/SVC), Naïve Bayes, and Gradient Boosting.

## Part 10: Results of the training process

Results for Model: Logistic Regression

--------------------------------------

Accuracy: 0.58

Precision: 0.5175433789954338

Recall: 0.58

F1: 0.5226298384851017

Confusion Matrix:

[[88 2 30]

[28 1 31]

[34 1 85]]

Results for Model: SVM - RBF kernel

-----------------------------------

Accuracy: 0.59

Precision: 0.590486616917646

Recall: 0.59

F1: 0.5239863147336458

Confusion Matrix:

[[94 0 26]

[30 0 30]

[37 0 83]]

Results for Model: SVM - Poly kernel

-----------------------------

Accuracy: 0.5466666666666666

Precision: 0.5485775630703167

Recall: 0.5466666666666666

F1: 0.48609599208312715

Confusion Matrix:

[[79 0 41]

[24 0 36]

[35 0 85]]

Results for Model: SVM - Sigmoid kernel

---------------------------------------

Accuracy: 0.5766666666666667

Precision: 0.4660608814175375

Recall: 0.5766666666666667

F1: 0.5152532963219987

Confusion Matrix:

[[90 2 28]

[29 0 31]

[36 1 83]]

Results for Model: Naive Bayes

------------------------------

Accuracy: 0.5866666666666667

Precision: 0.5888556235756736

Recall: 0.5866666666666667

F1: 0.521864823644183

Confusion Matrix:

[[86 0 34]

[23 0 37]

[30 0 90]]

Results for Model: Gradient Boosting

------------------------------------

Accuracy: 0.49666666666666665

Precision: 0.4975485826350508

Recall: 0.49666666666666665

F1: 0.4416057993300016

Confusion Matrix:

[[73 0 47]

[25 0 35]

[44 0 76]]

A picture containing table

Description automatically generated

Figure 14: Results of training the machine learning models

## Part 12: Comparison of the results of the lexicon models and the machine learning models

Graphical user interface, text

Description automatically generated

Figure 15: Results of VADR and SentiWordNet (Phase 1 Results)

For review, in figure 15 the results from the results from the two sentiment analysis models were not sufficient. The machine learning approach displayed in figure 14 shows much better results. Even the lowest performing machine learning model, Gradient Boosting had better overall statistics than the top performer in the sentiment analysis models, VADR. To further test the results, a grid search was performed with the following results:

Grid Search Analysis: SVC(random\_state=22)

Fitting 5 folds for each of 180 candidates, totalling 900 fits

Best paramters

{'C': 0.9, 'decision\_function\_shape': 'ovo', 'degree': 1, 'kernel': 'poly'}

Grid Search Analysis: LogisticRegression(class\_weight={0: 2, 1: 3, 2: 2}, random\_state=22)

Fitting 5 folds for each of 150 candidates, totalling 750 fits

Best paramters

{'C': 0.4, 'max\_iter': 1000, 'solver': 'lbfgs'}

Grid Search Analysis: GradientBoostingClassifier(max\_features=4, random\_state=22)

Fitting 5 folds for each of 270 candidates, totalling 1350 fits

Best paramters

{'learning\_rate': 0.060000000000000005, 'max\_depth': 4, 'n\_estimators': 190}

And each best-model produced:

Results for Model: SVM - BEST Poly kernel

-----------------------------

Accuracy: 0.5833333333333334

Precision: 0.5204761904761904

Recall: 0.5833333333333334

F1: 0.5254027313266444

Confusion Matrix:

[[91 2 27]

[29 1 30]

[36 1 83]]

Results for Model: Logistic Regression BEST

--------------------------------------

Accuracy: 0.5666666666666667

Precision: 0.5317141950401233

Recall: 0.5666666666666667

F1: 0.5382186272777432

Confusion Matrix:

[[81 12 27]

[25 7 28]

[34 4 82]]

Results for Model: Gradient Boosting BEST

------------------------------------

Accuracy: 0.49666666666666665

Precision: 0.4969711816845575

Recall: 0.49666666666666665

F1: 0.4413940783242509

Confusion Matrix:

[[72 0 48]

[28 0 32]

[43 0 77]]

Graphical user interface, text, application

Description automatically generated

Figure 16: Results of final training

As displayed in figure 16, SVC\_RBF still performed the best out of all models tested within the project.

## Part 13: Review research paper and examine options for improvement

As instructed in the suggested research paper, “…Because a review is written in natural language, the most obvious way of analyzing it is to identify frequently used terms. A weighting measure such as TF-IDF (as mentioned in Sect. 2.1) can be applied to determine how representative each term is in the review…” This project already uses TF-IDF effectively. The paper all suggests understanding a user’s sentiment orientation, positive or negative, towards a product, otherwise known as overall opinion.

It makes sense to analyse the text, both the summary text and the review text using TF-IDF and the overall score to determine a more accurate score based upon the sentiment.

First of all the dataframe is prepared, and non-useful columns are dropped. The helpful column is dropped in favour of upvote and downvote and totalvote columns. The raw date column is also dropped as day, month, year are already parsed out.

Text

Description automatically generated

Now that the dataframe is more properly prepared:

Text

Description automatically generated

Text

Description automatically generated

Results of XGBboost classifier predictions:

Table

Description automatically generated

Text

Description automatically generated

## References

Baccianella, S., Esuli, A., Sebastiani, F. (2010) “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.” *Proceedings of the 7th Conference on Language Resources and Evaluation (LREC 2010)*, pp. 2200-2204.

Beri, A. (2020) “Sentimental analysis using vader.” Towards Data Science. Retrieved March 19, 2023 from <https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664>

Hutto, C. J. (2023) “VADER-Sentiment-Analysis.” Retrieved March 19, 2023 from <https://github.com/cjhutto/vaderSentiment>

McAuley, J. (n.d.) “Amazon Product Data.” Amazon Review Data. Retrieved March 19, 2023 from <http://jmcauley.ucsd.edu/data/amazon/>