Sentiment Analysis Using Machine Learning and Amazon Product Reviews

Statement of Work – Version 1

AIDI 1002: AI Algorithms

Class Instructor: Marcos Bittencourt

Michael Molnar

Durham College # 100806823

Submission Date: November 4th, 2020

Executive Summary

Reading and analysis customer comments and reviews is a slow and labourious endeavour for a human being, though it is an essential way by which a business learns about itself and its customers. Machine Learning allows for the acceleration and automation of this process through text classification and sentiment analysis. A learning algorithm can be trained on pre-labelled data to discover associations between the text and the labels, and then is able predict labels for new and unseen text. This removes the need for predefined and human authored rules for differentiating between the class labels.

The scope of this project will be using Natural Language Processing to develop a Sentiment Analysis model that will predict whether a selection of text represents a positive, neutral, or negative feeling about a business and its offerings.

Business Problem Statement

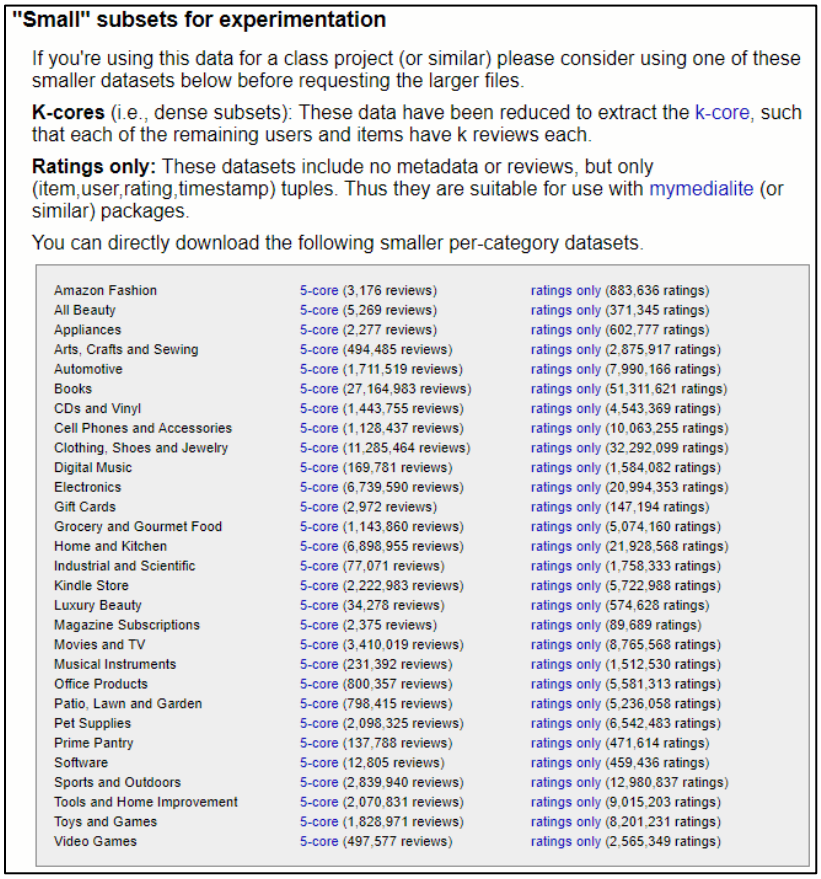
It has been estimated that 80-90% of all digital data is unstructured [1], taking the form of documents, emails, social media posts, images, and videos. In 2020 there are 3.8 billion people using social media, 43% of whom research products on social networks. 52% of online brand discovery takes place through social channels [2]. Knowing what your customers are saying about your brand, and what they are saying about your competition, is critical to building a successful business, and making proper use of the increasing amount of unstructured data is key. Every time your brand is mentioned in a review or social media post offers an opportunity to obtain an insight into your customer's feelings. Machine learning can be used to automate the process of reviewing all this data and extracting key insights that can lead to actionable steps for your business. This analysis can extend to how your brand is viewed compared to your competitors, allowing you to identify weaknesses in your businesses and possible market gaps that you can take advantage of. The automatic tagging of negative messages also enables you to prioritize issues and ensure that the most urgent communications are responded to first.

Rationale Statement

The model that will be the goal of this project will take a review or post about a product and classify it as being positive, neutral, negative. This type of analysis is currently being used in various fields; ecommerce, marketing, advertising, and politics, just to name a few. The main benefits include the ability to analyze and sort large amounts of text data in real time, in a consistent manner, without the need for a human reader.

Data Requirements and Sources

Building such a model will require a large amount of text data that has been pre-labelled with appropriate classes, such as rating scores. To satisfy this I have elected to work with the Amazon Review Data (2018) resource. This a collection of 233.1 million Amazon reviews, maintained by Jianmo Ni, and updated in 2018 [3]. Use of the dataset is permitted with a citation to the paper, *Justifying Recommendations Using Distantly-Labeled Reviews and Fine-Grained Aspects*, authored by Jianmo Ni, Jiacheng Li, and Julian McAuley [4]. The 233.1 million reviews have been reduced into 5-core and rating-only subsets, and then further divided according to product type, as shown in Figure 1.



"Small" subsets for experimentation

If you're using this data for a class project (or similar) please consider using one of these smaller datasets below before requesting the larger files.

K-cores (i.e., dense subsets): These data have been reduced to extract the k-core, such that each of the remaining users and items have k reviews each.

Ratings only: These datasets include no metadata or reviews, but only (item,user,rating,timestamp) tuples. Thus they are suitable for use with mymedialite (or similar) packages.

You can directly download the following smaller per-category datasets.

Amazon Fashion	5-core (3,176 reviews)	ratings only (883,636 ratings)
All Beauty	5-core (5,269 reviews)	ratings only (371,345 ratings)
Appliances	5-core (2,277 reviews)	ratings only (602,777 ratings)
Arts, Crafts and Sewing	5-core (494,485 reviews)	ratings only (2,875,917 ratings)
Automotive	5-core (1,711,519 reviews)	ratings only (7,990,166 ratings)
Books	5-core (27,164,983 reviews)	ratings only (51,311,621 ratings)
CDs and Vinyl	5-core (1,443,755 reviews)	ratings only (4,543,369 ratings)
Cell Phones and Accessories	5-core (1,128,437 reviews)	ratings only (10,063,255 ratings)
Clothing, Shoes and Jewelry	5-core (11,285,464 reviews)	ratings only (32,292,099 ratings)
Digital Music	5-core (169,781 reviews)	ratings only (1,584,082 ratings)
Electronics	5-core (6,739,590 reviews)	ratings only (20,994,353 ratings)
Gift Cards	5-core (2,972 reviews)	ratings only (147,194 ratings)
Grocery and Gourmet Food	5-core (1,143,860 reviews)	ratings only (5,074,160 ratings)
Home and Kitchen	5-core (6,898,955 reviews)	ratings only (21,928,568 ratings)
Industrial and Scientific	5-core (77,071 reviews)	ratings only (1,758,333 ratings)
Kindle Store	5-core (2,222,983 reviews)	ratings only (5,722,988 ratings)
Luxury Beauty	5-core (34,278 reviews)	ratings only (574,628 ratings)
Magazine Subscriptions	5-core (2,375 reviews)	ratings only (89,689 ratings)
Movies and TV	5-core (3,410,019 reviews)	ratings only (8,765,568 ratings)
Musical Instruments	5-core (231,392 reviews)	ratings only (1,512,530 ratings)
Office Products	5-core (800,357 reviews)	ratings only (5,581,313 ratings)
Patio, Lawn and Garden	5-core (798,415 reviews)	ratings only (5,236,058 ratings)
Pet Supplies	5-core (2,098,325 reviews)	ratings only (6,542,483 ratings)
Prime Pantry	5-core (137,788 reviews)	ratings only (471,614 ratings)
Software	5-core (12,805 reviews)	ratings only (459,436 ratings)
Sports and Outdoors	5-core (2,839,940 reviews)	ratings only (12,980,837 ratings)
Tools and Home Improvement	5-core (2,070,831 reviews)	ratings only (9,015,203 ratings)
Toys and Games	5-core (1,828,971 reviews)	ratings only (8,201,231 ratings)
Video Games	5-core (497,577 reviews)	ratings only (2,565,349 ratings)

Figure 1: Amazon Review Data (2018) Subsets

For the purposes of this Statement of Work document I have taken a preliminary view into the "Office Products" 5-core dataset. This is a relatively large subset of the overall dataset, offering 800,357 reviews. Along with a rating and the text and summary of the review, the dataset also provides a reviewer name and ID, the number of times the review has been voted as being helpful, and other identifying features. A sample of the data is shown in Figure 2.

```
office_reviews = pd.read_json('Office_Products_5.json.gz', orient='records', lines=True)
```

```
office_reviews.head()
```

	overall	verified	reviewTime	reviewerID	asin	style	reviewerName	reviewText	summary	unixReviewTime	vote	image
0	4	True	11 7, 2017	A2NIJTYWADLK57	0140503528	{'Format:': ' Board book'}	cotton clay	kids like story BUT while i really wanted a bo...	good story, small size book though	1510012800	NaN	NaN
1	4	True	03 7, 2017	A2827D8EEURMP4	0140503528	{'Format:': ' Hardcover'}	emankcin	Bought this used and it came in great conditio...	Good	1488844800	NaN	NaN
2	5	True	06 25, 2016	APB6087F4J09J	0140503528	{'Format:': ' Board book'}	Starbucks Fan	Every story and book about Corduroy is Fantast...	Best Books for All Children	1466812800	NaN	NaN
3	5	True	02 21, 2016	A2DHERRZIPFU7X	0140503528	{'Format:': ' Paperback'}	Caitlyn Jacobson	I purchased this book for my first grade class...	Great for Math!	1456012800	NaN	NaN
4	5	False	08 2, 2015	A2XCLJRGFANRC	0140503528	{'Format:': ' Hardcover'}	E. Ervin	Having spent numerous years in an elementary s...	Love Corduroy	1438473600	NaN	NaN

```
office_reviews.shape
```

```
(800357, 12)
```

Figure 2: Sample of the "Office Products" 5-core Dataset

Data Limitations, Assumptions, Constraints

This project will aim to analyze real Amazon reviews written by real human beings, and I make the following assumptions before looking into the text data:

- The text may or may not contain punctuation
- The text will probably contain spelling errors
- The text may contain special characters, urls, emojis
- The tone of the text will vary – there may be issues with sarcasm and comparisons, for example

I also assume that the distribution of product ratings will be extremely imbalanced. It would be a personal hypothesis that a customer is more likely to review a product that they absolutely loved or hated, rather than if it were merely adequate. A plot of the distribution of ratings reveals that 5/5 represents most of all ratings, as shown in Figure 3.

```
sns.countplot(x='overall', data=office_reviews).set_title('Distribution of Ratings')
```

```
Text(0.5, 1.0, 'Distribution of Ratings')
```

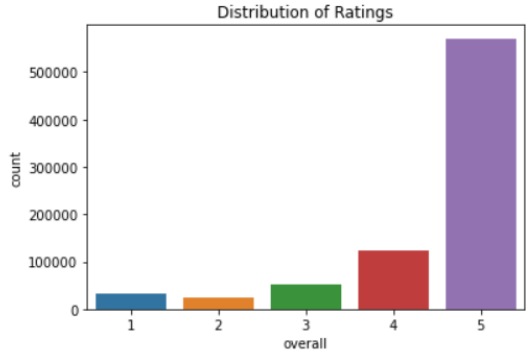


Figure 3: A Plot of the Distribution of Class Labels

Generating the counts and percentages for each class label further illustrates the imbalance this dataset contains, as shown in Figure 4.

	Label Counts	Percentage
5	570993	0.713423
4	122889	0.153543
3	50702	0.063349
1	31697	0.039604
2	24076	0.030082

Figure 4: Percentages of Class Labels

This imbalance will have to be addressed in the algorithm testing and evaluation stage of the project. The reason for this, as seen in Figure 4, is that over 86% of all reviews are given a positive rating of 4 or 5. This means that an algorithm that simply predicts positive for every review will achieve an accuracy score of 86%. There are several approaches that can be taken to address this. Two common techniques are over-sampling and under-sampling. In over-sampling instances of the smaller classes are duplicated in the training dataset. This can overcome imbalance but can lead to overfitting. Under-sampling is removing instances of the larger classes, but this creates the opportunity to lose potentially useful information contained in the removed rows. A third solution would be data augmentation [5]. Since the Amazon Review dataset consists of numerous product categories, I can work with one, split it into training and testing data, and then augment the training set with reviews from a secondary dataset, thereby balancing the distribution of the classes. Appropriately evaluation metrics can also be used to take this imbalance into account. This will be discussed later in this Statement of Work.

The model I create will be limited by its ability to process aspects of human speech and text that would be easy for a human interpreter to understand. The model will be constrained by its ability to recognize tone and sarcasm, for example. The model must also be able to examine

word combinations when assigning labels. For example, looking just at the keyword “bad” might indicate a negative sentiment, but if the review actually said “not bad” then the score would be different. A way to deal with this is by using n-grams, which will split the text into word combinations rather than single words. This will allow a phrase like “not bad” to be considered as a single entity. Context and comparisons will also be difficulties. A phrase such as “this is better than other products” is certainly positive, but “this is better than nothing” is negative. Another constraint will come in how misspelled words are dealt with.

These assumptions and limitations will be explored in the Exploratory Data Analysis and Model Testing portions of the project that will follow.

To get a sense of the text data I look at the first 5/5 review and the first 1/5 found in the dataset. Figure 5 below shows the review text for the first 5/5 review.

```
office_reviews['reviewText'].iloc[2]
```

```
'Every story and book about Corduroy is Fantastic. This book is great and I bought all the Corduroy books for my 2 boys and now for their total of 5 children. You have to buy a Corduroy bear for everyone who has the books. Love to hold them while the stories are read.'
```

```
office_reviews['overall'].iloc[2]
```

```
5
```

Figure 5: A Sample 5/5 review.

This is a positive review and contains several obviously positive words, such as “fantastic”, “great”, and “love”. The phrase “have to buy” would also be very indicative of a positive sentiment.

Figure 6 shows the first 1/5 review, and right away I can identify numerous factors that will be problematic and difficult for a classification model to handle.

```
office_reviews['reviewText'].iloc[150]
```

```
"If you're looking for something high quality don't get this. It's a piece of paper. It wasn't well wrapped in the packing and was just shoved in the box. The box was wet upon arrival and the paper was mostly ruined. I'm unable to use it in my classroom. Wish I had gone with something better quality. It looks better in the picture than it does in person, especially when wet and wrinkled and doesn't come with the necessary items to attach the arrow (that I have to cut out on my own). For what I paid for it I definitely expected more. My kids like it and like the look of it. It's just not sturdy enough for a classroom. I'm going to have to attach cardboard to the back and hope I can iron it flat now that it's dry. Not sure how long it will last. Also have to go shopping to get something to somehow attach the arrow so that it will turn when the kids change the weather each day. Uggh. NOT my best buy. Don't recommend it. It looks great, just not high enough quality for a classroom, especially one with younger kids (K - 2nd grade)."
```

```
office_reviews['overall'].iloc[150]
```

```
1
```

Figure 6: A Sample 1/5 Review.

The review contains the phrase “high quality” in its opening sentence, though obviously the surrounding words make it clear to a human reader that it is being used in a negative context here. There are some clearly negative phrases, “don’t get this”, “expected more”, “don’t recommend it”, for example, but again phrases that will be confusing if taken out of context –

"it looks great", in the last sentence of the review, being a prime example. These are some of the issues that will limit the accuracy of the classification model.

Model and Architecture Approach

In order to create a sentiment analysis model I will first combine the labels into three classes – "negative" for ratings of 1 and 2, "neutral" for a rating of 3, and "positive" for a rating of 4 and 5. I will then determine the most appropriate way of dealing with the imbalance in the classes, as detailed above. Next, before any learning algorithms are tested, data cleaning and pre-processing will be required.

There are several common techniques used for cleaning text data for machine learning projects, including:

- Lowercasing all text
- Removing special characters, punctuation, numbers
- Removing stop words (common English words such as "and", "the", "in", and so on that will not add value to the learning algorithms)
- Normalization through Stemming or Lemmatizing (reducing words to their roots)

The Natural Language Toolkit, *NLTK*, is a Python platform that offers many useful libraries to automate several of these processes [6]. One consideration I will make is examining the use of emojis before simply removing punctuation and special characters. I have taken a preliminary look at the dataset and examined the number of reviews that contain a ":)" within their text. These total 5,338 rows of the Office Supplies dataset. Almost all of the ratings for these rows are 5 and 4, as would be expected, but there are some lower scores as well. The rows in which the review is simply ":)", without any other text, total 267. Further analysis into other emojis will be completed as part of the data cleaning and preparation process.

The next stage of the project will be feature extraction. Vectorization, the process of transforming words into numeric vectors or matrices based on their frequencies, is part of this process. There are numerous choices for this – Bag of Words and Count Vectorizer (counts the number of times each word appears in the collection), TF-IDF (measures importance by infrequency) [7], or pre-trained vector embeddings such as Word2Vec [8] and GloVe [9]. Word2Vec and GloVe aim to capture the semantic meaning of words and cluster together words that are similar.

Once data cleaning is complete and a vectorizer is selected, a preprocessing pipeline will be created, whereby the text can be transformed into a form suitable to be fed into a classification algorithm.

Modelling Techniques and Evaluation Metrics

There are numerous approaches to algorithm selection for this project and many will be tested. Here we are dealing with three classes, positive, neutral, and negative, and therefore a multiclass classifier will be required. Multiclass Logistic Regression, Naïve Bayes, Support Vector Machines, Random Forest, k-Nearest Neighbours are examples. The problem can also be addressed using the OneVsRest algorithm, which uses a classifier such as Logistic Regression or SVM to transform the problem into a binary classification for each label [10]. There are also deep learning techniques that will be explored. The theory behind these algorithms will be detailed in the modelling reports to follow later in the project.

In terms of evaluation, cross-validation and metrics such Accuracy score, F1 Score, Precision and Recall, and ROC/AUC Curves will be considered. If the dataset remains imbalanced, Scikit-Learn's Balanced Accuracy Score [11] can be used as an alternative to the traditional accuracy score to account for this. Scikit-Learn's predict_proba() function [12] will allow me to generate the probabilities the label selections are based on when using probability based algorithms such as Logistic Regression and Random Forests.

Prototyping and Development

The final deployment of this project will be in the form of a Cloud based application. It will have an interface for the user to enter their own review text and a sentiment score will be generated, along with the determining probability for the label. The application could automatically flag reviews that are determined to be negative with a high degree of certainty so that these customers could have their complaints dealt with as a priority. Figure 7 is what a potential endpoint might look like.

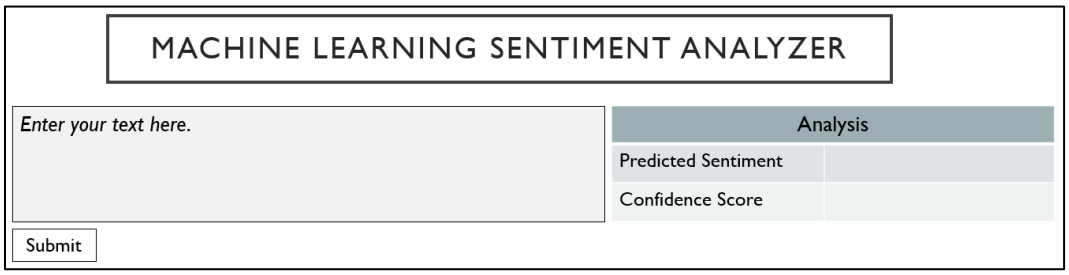


Figure 7: Mockup of a Potential Final Application

Testing Process

Beyond the evaluation metrics described above I will study the words and phrases the model most heavily relies on in making its label determinations. This will allow me to ensure the model is performing as accurately as possible by reviewing if it gives weight to words it should

not. This process will also allow me to evaluate how well the assumptions, limitations, and constraints described previously are being handled.

Project Timeline

The final model and application will be delivered in approximately six weeks from today's date. To ensure that this deadline is met I list the project milestones and planned completion dates in Figure 8.

Task	Planned Completion By
Phase 1: Data Analysis	
Exploratory Data Analysis	November 8
Data Cleaning and Manipulation	November 15
Statistical Analysis	November 15
Feature Extraction	November 15
Update SOW	November 15
Phase 2: Modelling	
Test and Evaluate Learning Algorithms and Frameworks	November 15
Research Software Tools and Pipeline	November 22
Assess Data Assumptions, Limitations, and Constraints	November 22
Phase 3: Prototyping	
Prototype Proposed Model Architecture	November 22
Model Tweaking and Evaluation	November 22
Update SOW and Prepare Modelling Report	November 22
Phase 4: Delivery	
Develop Software Pipeline to Host Model	December 18
Develop Solution Endpoint for Final User	December 18
Prepare Final Report and Documentation	December 18

Figure 8: Project Milestones and Timelines

References

[1] Davis D. AI Unleashes the Power of Unstructured Data. CIO. https://www.cio.com/article/3406806/ai-unleashes-the-power-of-unstructured-data.html. Accessed October 28, 2020.

[2] Cooper P. 140+ Social Media Statistics that Matter to Marketers in 2020. Hootsuite. https://blog.hootsuite.com/social-media-statistics-for-social-media-managers/. Accessed October 30, 2020.

[3] Ni J. Amazon Review Data (2018). https://nijianmo.github.io/amazon/index.html. Accessed October 28, 2020.

[4] Ni J, Li J, McAuley J. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. *Empirical Methods in Natural Language Processing (EMNLP)*. 2019. http://cseweb.ucsd.edu/~jmcauley/pdfs/emnlp19a.pdf. Accessed October 28, 2020.

[5] Brownlee J. 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset. Machine Learning Mastery. https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/. Accessed October 30, 2020.

[6] Natural Language Toolkit. NLTK 3.5 Documentation. https://www.nltk.org/. Accessed October 30, 2020.

[7] Feature Extraction. Scikit-Learn. https://scikit-learn.org/stable/modules/feature_extraction.html. Accessed October 30, 2020.

[8] Word2vec. Google Code. https://code.google.com/archive/p/word2vec/. Accessed October 30, 2020.

[9] Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/. Accessed October 30, 2020.

[10] Sklearn.multiclass.OneVsRestClassifier. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html. Accessed October 30, 202.

[11] Sklearn.metrics.balanced_accuracy_score. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html. Accessed October 30, 2020.

[12] Sklearn.Linear_model.LogisticRegression. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed October 30, 2020.