NLP Group Project

Group– 3

Devanshi Shah

Hitesh Dharmadhikari

Jefil Tasna John Mohan

Nestor Romero Leon

Shrikant Kale

Professor Alaa Alslaity

COMP 262: Natural Language & Recommender System

Centennial college

Table of Contents

1. Introduction

With Amazon's Musical Instruments dataset, we have conducted a sentiment analysis model for products based on customers' textual reviews, using both a Lexicon approach and a machine learning approach.

Here, a sequence of steps such as data exploration, data cleanup and further, data pre-processing and feature engineering, text representation are done to pass it to a Lexicon model for sentiment analysis use case. In the second phase, the same steps were executed but using Machine Learning approach. Lastly, a study of how to utilize the same review data to enhance rating values for a possible use case with recommender systems is added in our report.

2. Detailed results of dataset exploration & conclusions

2.1 Basic Information of dataset

We are using sentimental analysis here as a lexicon approach. Here, we have Musical_Instruments.json as our dataset.

- **Total number of columns**: 9
- **Total number of rows**: 10261

Here is the example of each column:

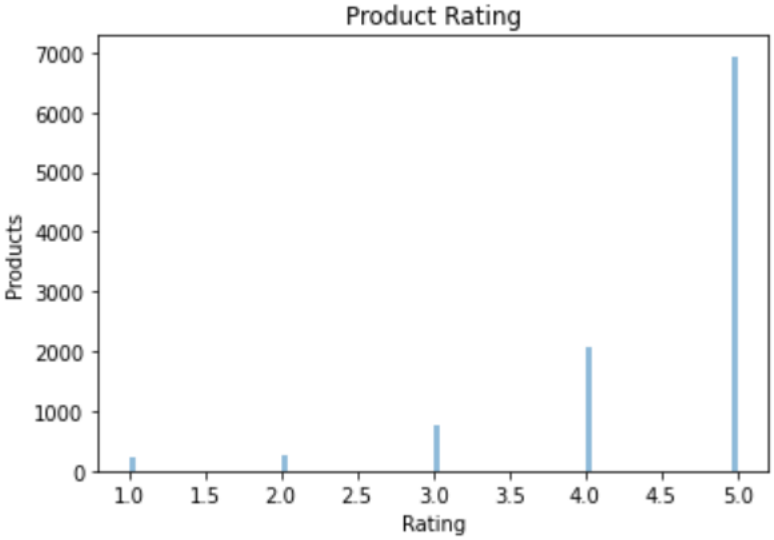
1. reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
2. asin - ID of the product, e.g. 0000013714
3. reviewerName - name of the reviewer
4. helpful - helpfulness rating of the review, e.g. 2/3
5. reviewText - text of the review
6. overall - rating of the product
7. summary - summary of the review
8. unixReviewTime - time of the review (unix time)
9. reviewTime - time of the review (raw)

- **Missing values:** Found in reviewerName column with 27 values missing
- **Type of Int Columns:** 2 (overall, unixReviewTime)
- **Type of object Columns:** 7 (reviewerID, asin, reviewerName, helpful, reviewText, summary, reviewTime)
- The overall frequency for the rating column from 1 to 5 is:

```
5   6938
4   2084
3    772
2    250
1    217
```

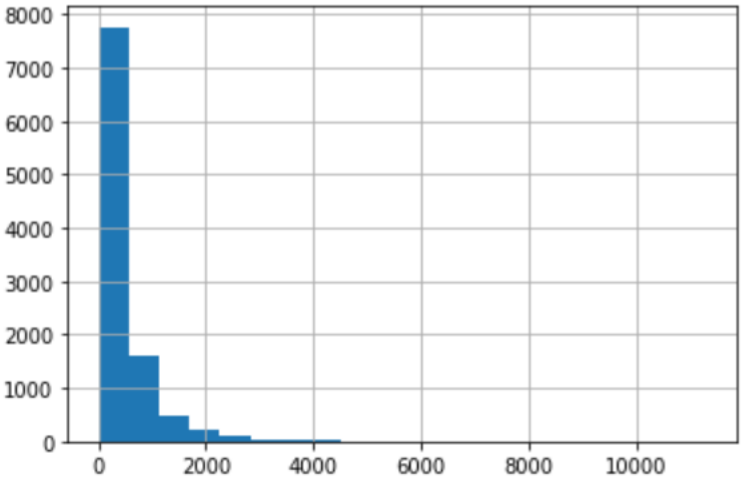
Here, we can clearly see that Positive reviews are more in number rather than negative reviews and the overall review mean is also 4.488743787155248.

2.2 Distribution of number of reviews across the products



From the above graph it is clearly visible that good ratings are given to the more products. So, 7000 products are given 5 rating, approximate 1500 products are given 4 ratings, 1000 products are given 3 ratings and rest other products are given 2 and 1 ratings.

2.3 Distribution of reviews per user



From the above graph we can see that, almost 75% of the reviews are short in length, but 25% of the reviews are of long characters.

3. Dataset pre-processing steps with explanation and justification of choices

- Here, we have used sample data set from full data set with the help of stratified sampling.
- Taking 200 samples for every instance (Ranging from 1 to 5) from overall column

5 □ 200
4 □200
3 □200
2 □200
1 □200

- After this, we have created labels on the basis of rating and created a new column named it as ratings, where the value is 'Positive' if the ratings are greater than 3 and the value is 'Negative' if the ratings are less than 3; otherwise, the value is 'Neutral' if the ratings are equal to 3.
- So, we will get three labels in ratings columns (Sentiment Analysis):

Positive	400
Neutral	200
Negative	400

3.1 Column selection

- We have removed/dropped few columns and also mentioned the reason for it as follows
  1) **reviewerName:** This column contains the name of the reviewers which will not be helpful to predict sentiment using either the lexicon approach or Machine learning approach. It also has 27 missing values.
  2) **Helpful:** It contains helpfulness ratings of review. It will not be helpful in NLP with the lexicon approach.
  3) **reviewTextLength:** It represents the length of the review text, so the length of the text is also not required here in Sentiment for prediction.
  4) **unixReviewTime:** This column contains the date and time. The lexicon approach is rule-based; that is why the lexicon approach does not have any connection with the date and time for sentiment. It also might not be useful in the Machine learning approach.
  5) **reviewTime:** The reason for dropping this column is similar as the unixReviewTime.

3.2 Data Cleanup

1) **Lowercase: -** It will decrease the number of unique words in the corpus which eventually increase efficiency.

   Before lowercasing, the same words act as different for some models. For example, ‘product’, ‘Product’ and ‘PRODUCT’ will be taken as different words in many models. However, all three should be taken as the same for many use cases of AI. Lowercasing helps to achieve this which can increase accuracy.

2) **Remove punctuation: -** In the case of sentiment analysis, many punctuations like full stops, commas are not required and those can be removed.

3) **Lemmatization: -** it decreases the size of the corpus and increases efficiency.

   Converting the same words in a different format to its base word, can lead to a better understanding of many models which helps to increase accuracy.

4. Text representation model with explanation and justification

TF- IDF (Term Frequency – Inverse Document Frequency):

TF-IDF is a text representation technique that converts text into numeric format with the help of matrix format representation where columns are unique words in the corpus and rows are the text of all instances.

Term frequency works by looking at the frequency of a *particular term* you are concerned with relative to the document.

Inverse document frequency looks at how common (or uncommon) a word is amongst the corpus. IDF is calculated as follows where ***t*** is the term (word) we are looking to measure the commonness of and ***N*** is the number of documents (d) in the corpus (D)...

Though TF-IDF does not evaluate and grasp the context of the sentence, whereas bag-of-N-grams does. Considering the weight of words may be more advantageous in the sentiment analysis task that TF-IDF is performing. As a result, TF-IDF was chosen over bag-of-N-grams. Still, it's a good idea to experiment with the bag-of-N-grams method and compare the results to the TF-IDF results.

TF-IDF will be used more for sentiment analysis as it considers the importance of each word in the corpus. On the other hand, Bag-of-words and One-hot encoding does not consider the importance of each words. But it is necessary to consider the importance of each word in the corpus for sentiment analysis as weights will increases the accuracy.

5. Models, per model clarify:

Lexicon based approaches:

Lexicon based approaches are based on rule – based approach. It has certain set of rules for their model. Whereas, in machine learning we need to train our model but lexicon approach are based on set of rules. So, it is easy to implement and cost-effective too.

Lexicon based approaches can be implemented by setting and manually deriving rules and it can be also implemented by importing the existing packages.

Here, we have used second approach to implement the lexicon-based approach for sentimental analysis.

Here we have used 2 different types of lexicon-based approaches for sentimental analysis.

i) Valence Aware Dictionary and Sentiment Reasoner (VADR)

- VADR (Valence Aware Dictionary for Sentiment Reasoning) is a text sentiment analysis model that takes into account both the polarity (positive/negative) and the intensity (strong) of emotion. This package includes understanding of punctuation, word-shape, negations, contractions as negations, slang, and emojis, all of which helps in accurately predicting many more text. VADR can be used in a variety of languages.

- Because it has been specifically adjusted to assess sentiments conveyed in social media text, it is widely utilized in analyzing sentiment on social media text (as per the linked docs). It is now included in the Natural Language Toolkit, or NLTK.

Heuristic and the way it works:

- The compound score is the sum of positive, negative & neutral scores which is then normalized between -1(most extreme negative) and +1 (most extreme positive).

- The more Compound score closer to +1, the higher the positivity of the text.

- By applying the rules mentioned above, VADR calculates positivity, negativity, neutrality, and compound of all using heuristic. The total of all 3 – pos, neg and neu is 1 (one).

ii) SentiWordNet

- SentiWordNet uses WordNet's database for its operations.

- It also includes the ability to assess positivity, negativity, or neutrality, which is essential for Sentiment Analysis.

<u>Heuristic and the way it works:</u>

- The dataset must be preprocessed, which includes the removal of stopwords and punctuation marks.

- While using SentiWordNet, it is important to find out the Parts of Speech for each word present in the dictionaries. Parts of Speech require the use of WordNet syntax.

- Noun (n) Verb (v) Adjective (a) Adverb Preposition Conjunction Pronoun Interjection The first three are the most commonly used while reviewing sentiments of a sentence

- The polarity of each word, in context with POS tagging, is found out using the sentiwordnet functions — pos_score(), neg_score() and obj_score().

Machine Learning-based approaches

Here, after using 4 different approaches of machine learning-based approaches for sentiment analysis on only a sample data set, It was noticeable that Logistic Regression and SVM algorithms were working better as compared to Naive Bayes and Gradient Boosting.

i) Support Vector Machine:-

- Support Vector Machine (SVM) is a supervised machine learning algorithm model that is used to solve Classification and Regression problems. Where the Classification problem is a label/group and the Regression problem is a continuous value.
- SVM models can categorize new text after being given sets of labeled training data for each category.
- They have two key advantages over newer algorithms like neural networks: greater speed and better performance with a limited number of samples (in the thousands).

Heuristic and the way it works:

- A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it’s simply a line) that best separates the tags. This can be done when having a 1D graph
- While having a 2D or more dimension graph and in order to solve it, one of the methods that Support Vector uses SVM kernels
- SVM kernel's main aim is to transform a low dimension into a high dimension to classify points (features) by a hyperplane
- The kernel is in need because our dataset is not linearly separable
- A support vector machine only takes care of finding the decision boundary
- A kernel is used to map a space between data points where data points are not linearly separable and also removes the allowances for erroneous classification
- SVM finds best decision boundary point for 2 dimensions (tags), it is separable by a hyperplane. For 3 dimension SVM finds best decision boundary plane with 2 dimensions which divide 3 spaces into 2 parts and thus act as a hyperplane. Similarly, for n dimensions , we have n-1 hyperplane separating into two parts

ii) Logistic Regression

Multinomial Logistic Regression is a classification algorithm that generalize Logistic Regression to multiclass problems. On this particular case the result set of classes include the values for Positive, Neutral and Negative feelings. For extending Logistic Regression, instead of a One-Vs-Rest approache, Multinomial Logistic Regression uses cross-entropy as a loss function and predicts a multinomial probability distribution to solve the problem.

Heuristic and the way it works

- As an extension to Logistic Regression and with a change in the loss function used, the idea behind Multinomial Logistic Regression is to produce a probability for each class in a multiclass problem as if the observations were modeled as a multinomial distribution.
- Since the method calculates a probability for each class, the final selection of the prediction could be translated as the argmax value of the probability result array in which each element represents the probability of each class
- Multinomial Logistic Regression could be labeled as a statistical classification technique, that uses the idea of constructing a linear predictor function that aims at predict the probability that an observation has a particular outcome Ki in set of K classes.

iii) Grid Search auxiliary method

As part of the project we used the GridSearch method to confirm the results obtained by the previous models in under to validate the selection of hyperparameters. For the exercise we constructed a dictionary for the different models or estimators and a set of parameter ranges that the method will use in order to create all the possible combinations and finally report the best combination of hyperparameters for each model.

Below are the results obtained for our validation exercise, for the grid search parameters we use standard ranges suggested by sciklit learn documentation

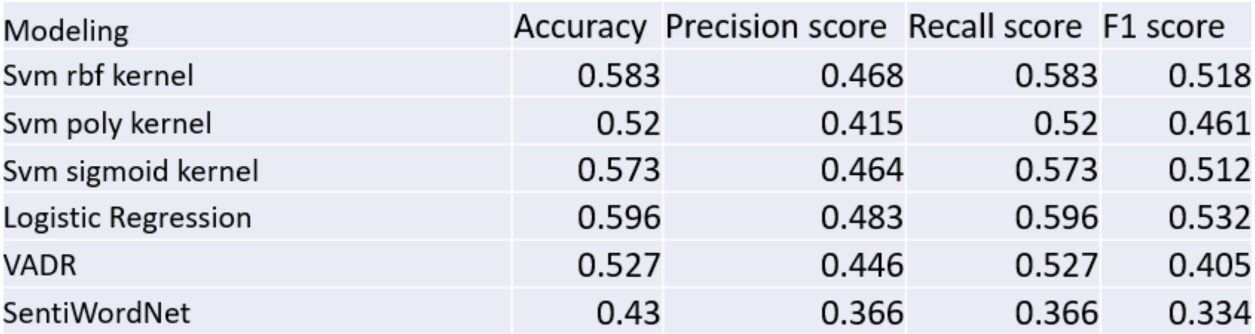
Predictor	Grid Search Parameters

LogisticRegression	lrg_params = {
	'solver' : ['lbfgs', 'sag', 'saga'],
	'C' : np.arange(0.1,1.1,0.1),
	'max_iter' : np.arange(1000,2000,200)
	}

GradientBoosting	gb_params = {
	"learning_rate" : np.arange(0.01,0.1,0.01),
	"n_estimators" : np.arange(100,200,10),
	"max_depth" : np.arange(2,5,1),
	}

Predictor	Accuracy	Precision	Recall	F1	Confusion Matrix	Best Hyper-Parameters
LogisticRegression(class_weight={0: 2, 1: 3, 2: 2}, random_state=0)	0.56	0.53	0.56	0.54	[[87 13 20] [33 7 20] [35 11 74]]	{'C': 0.5, 'max_iter': 1200, 'solver': 'saga'}
GradientBoostingClassifier(max_features=4, random_state=0)	0.51	0.41	0.51	0.46	[[70 0 50] [27 0 33] [36 0 84]]	{'learning_rate': 0.05, 'max_depth': 3, 'n_estimators': 190}

6. Testing results summary:



Modeling	Accuracy	Precision score	Recall score	F1 score
Svm rbf kernel	0.583	0.468	0.583	0.518
Svm poly kernel	0.52	0.415	0.52	0.461
Svm sigmoid kernel	0.573	0.464	0.573	0.512
Logistic Regression	0.596	0.483	0.596	0.532
VADR	0.527	0.446	0.527	0.405
SentiWordNet	0.43	0.366	0.366	0.334

Result of VADR:-

Accuracy: 0.5271428571428571

Precision score: 0.44642857142857145

Recall score: 0.5271428571428571

F1 score: 0.40524172755392707

Result of SentiWordNet:-

Accuracy: 0.43

Precision score:0.36628925646454585

Recall score:0.36666666666666664

F1 score: 0.3349675089349278

Result of SVM RBF Kernel:-

When we are using SVM with Radial Basis Function(RBF) kernel is a default kernel for svm when none is used, two parameters must be considered ‘c’ and ‘gamma’. Here, parameter ‘c’ trades off the misclassification of training examples against the simplicity of the decision surface. A low ‘c’ makes the decision surface smooth, while a high ‘c’ aims at classifying all training examples correctly. Whereas, ‘gamma’ defines how much influence a single training example has.
The best parameters are {‘C’ : 1.0, ‘gamma’ : 1.0}

```
Accuracy using rbf SVM: 0.5833333333333334

Precision using rbf SVM: 0.4685078909612626

Recall using rbf SVM: 0.5833333333333334

F1 using rbf SVM: 0.5185299295774649

Confusion Matrix using rbf SVM: [[92  0 28]
[35  0 25]
[37  0 83]]
```

Result of SVM Poly Kernel:-

The polynomial kernel represents the similarity between two vectors. Conceptually, the polynomial kernels consider not only the similarity between vectors under the same dimension but also across dimensions.
The poly kernel allows the learning of non-linear data. Here, we have set the degree = 3 which is default value and here the term Degree means a degree of the polynomial kernel function (‘poly’).

```
Accuracy using poly SVM: 0.52

Precision using poly SVM: 0.41588989677822963

Recall using poly SVM: 0.52

F1 using poly SVM: 0.461539729867132

Confusion Matrix using poly SVM: [[72  0 48]
 [31  0 29]
 [36  0 84]]
```

Result of SVM Sigmoid Kernel:-

It computes the sigmoid kernel between two vectors. The sigmoid kernel is also known as a hyperbolic tangent or Multilayer Perceptron (because, in the neural network field, it is often used as a neuron activation function).

```
Accuracy using sigmoid SVM: 0.5733333333333334

Precision using sigmoid SVM: 0.4644790812141099

Recall using sigmoid SVM: 0.5733333333333334

F1 using sigmoid SVM: 0.5125614737017589

Confusion Matrix using sigmoid SVM:[[89  3 28]
[33  0 27]
[37  0 83]]
```

Result of Logistic Regression:-

Here, For multiclass problems, only 'sag' handle multinomial loss

Maximum number of iterations taken for the solvers to converge. For Logistic regression here max_iter = 2000 is selected as a parameter.

Here, we need to add multi_class = multinomial because for our dataset we have used multi-label classification.

```
Accuracy using Logistic Regression: 0.5966666666666667

Precision using Logistic Regression: 0.4835178856231488

Recall using Logistic Regression: 0.5966666666666667

F1 using Logistic Regression: 0.5326343526801193

Confusion Matrix using Logistic Regression: [[94  2 24]
[36  0 24]
[35  0 85]]
```

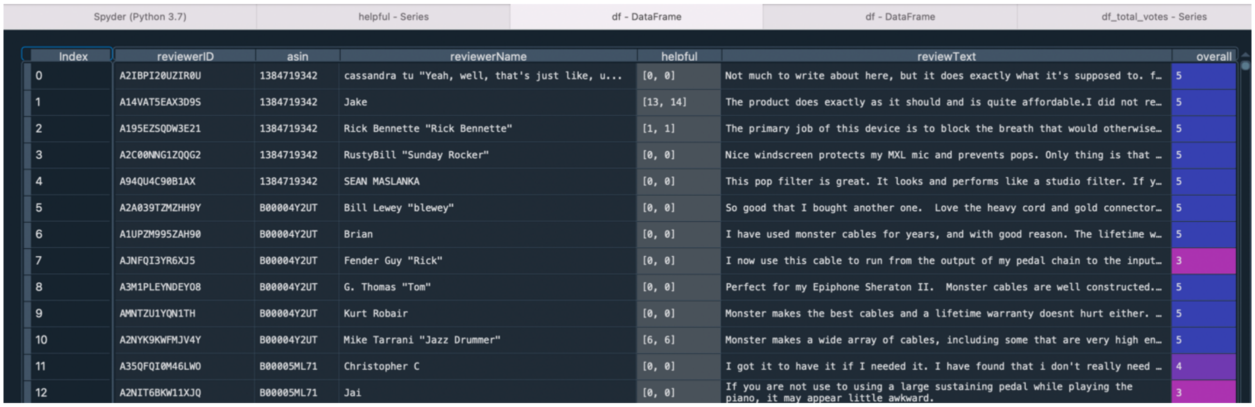
7. Future Work: Enhancing rating value using reviews data

Introduction:

As we have both reviews and ratings in the given dataset, we can use reviews as auxiliary resource to enhance ratings. In the paper, “Recommender systems based on user reviews”, several approaches to enhance the rating values have been discussed. Since the provided dataset has a helpful column available and due to its better performance with regards to RMSE, we decide to use the approach called “considering review helpfulness” to provide a solution for enhancing rating values using reviews.

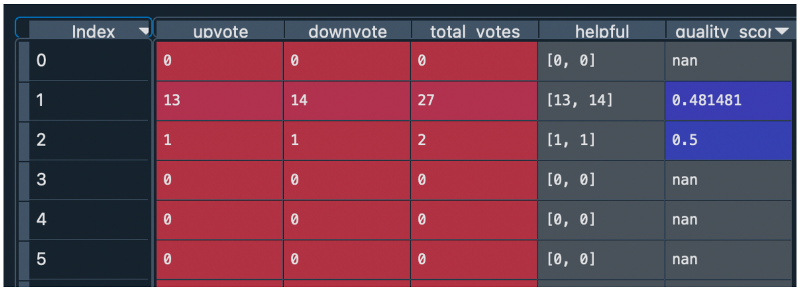
The steps followed to derive a solution for this use case are outlined below:

1. Snapshot of the dataset with features including reviewerID, asin, helpful, reviewText and overall



Index	reviewerID	asin	reviewerName	helpful	reviewText	overall
0	A2IBPI20UZIR0U	1384719342	cassandra tu "Yeah, well, that's just like, u...	[0, 0]	Not much to write about here, but it does exactly what it's supposed to. f…	5
1	A14VAT5EAX3D9S	1384719342	Jake	[13, 14]	The product does exactly as it should and is quite affordable.I did not re…	5
2	A195EZSQDW3E21	1384719342	Rick Bennette "Rick Bennette"	[1, 1]	The primary job of this device is to block the breath that would otherwise…	5
3	A2C00NNG1ZQQG2	1384719342	RustyBill "Sunday Rocker"	[0, 0]	Nice windscreen protects my MXL mic and prevents pops. Only thing is that …	5
4	A94QU4C90B1AX	1384719342	SEAN MASLANKA	[0, 0]	This pop filter is great. It looks and performs like a studio filter. If y…	5
5	A2A039TZMZHH9Y	B00004Y2UT	Bill Lewey "blewey"	[0, 0]	So good that I bought another one. Love the heavy cord and gold connector…	5
6	A1UPZM995ZAH90	B00004Y2UT	Brian	[0, 0]	I have used monster cables for years, and with good reason. The lifetime w…	5
7	AJNFQI3YR6XJ5	B00004Y2UT	Fender Guy "Rick"	[0, 0]	I now use this cable to run from the output of my pedal chain to the input…	3
8	A3M1PLEYNDEY08	B00004Y2UT	G. Thomas "Tom"	[0, 0]	Perfect for my Epiphone Sheraton II. Monster cables are well constructed.…	5
9	AMNTZU1YQN1TH	B00004Y2UT	Kurt Robair	[0, 0]	Monster makes the best cables and a lifetime warranty doesnt hurt either. …	5
10	A2NYK9KWFMJV4Y	B00004Y2UT	Mike Tarrani "Jazz Drummer"	[6, 6]	Monster makes a wide array of cables, including some that are very high en…	5
11	A35QFQI0M46LWO	B00005ML71	Christopher C	[0, 0]	I got it to have it if I needed it. I have found that i don't really need …	4
12	A2NIT6BKW11XJQ	B00005ML71	Jai	[0, 0]	If you are not use to using a large sustaining pedal while playing the piano, it may appear little awkward.	3

2. Computing upvotes, down_votes and total votes using the helpful column of the dataset



Index	upvote	downvote	total_votes	helpful	quality_scor
0	0	0	0	[0, 0]	nan
1	13	14	27	[13, 14]	0.481481
2	1	1	2	[1, 1]	0.5
3	0	0	0	[0, 0]	nan
4	0	0	0	[0, 0]	nan
5	0	0	0	[0, 0]	nan

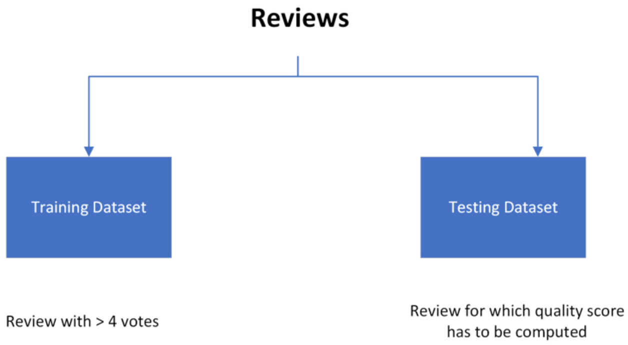
3. For reviews that have helpfulness votes, we compute a quality score using the following formula:

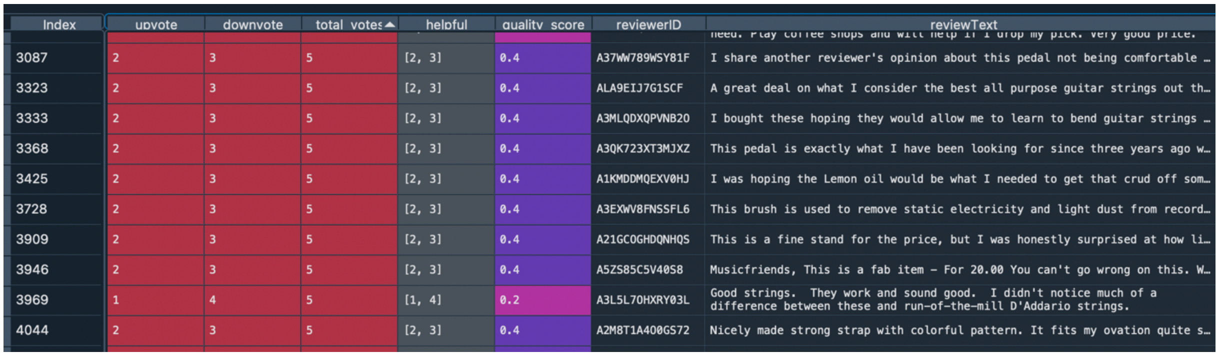
> Quality score = Number of helpful votes / Total number of votes

From the screenshot it is visible that we have a lot of reviews that do not have votes and thus quality score cannot be computed using this approach.

4. To compute the quality score for the reviews that do not have quality scores we use employ multi-step approach that uses reviews:

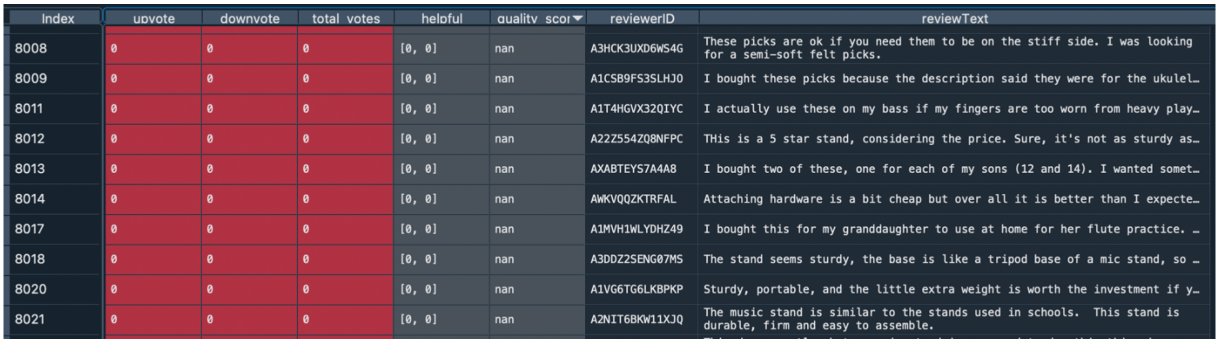
   A. We will first split our dataset into training and testing. Reviews that have more than 4 votes will be filtered and used as training set for the Logistic Regression Model. The model will then be used to predict the quality score for the testing set. Testing set consists of reviews for which we need to calculate the quality score.





Index	upvote	downvote	total votes	helpful	quality score	reviewerID	reviewText
							need. Play coffee shops and will help if I drop my pick. Very good price.
3087	2	3	5	[2, 3]	0.4	A37WW789WSY81F	I share another reviewer's opinion about this pedal not being comfortable …
3323	2	3	5	[2, 3]	0.4	ALA9EIJ7G1SCF	A great deal on what I consider the best all purpose guitar strings out th…
3333	2	3	5	[2, 3]	0.4	A3MLQDXQPVNB2O	I bought these hoping they would allow me to learn to bend guitar strings …
3368	2	3	5	[2, 3]	0.4	A3QK723XT3MJXZ	This pedal is exactly what I have been looking for since three years ago w…
3425	2	3	5	[2, 3]	0.4	A1KMDDMQEXV0HJ	I was hoping the Lemon oil would be what I needed to get that crud off som…
3728	2	3	5	[2, 3]	0.4	A3EXWV8FNSSFL6	This brush is used to remove static electricity and light dust from record…
3909	2	3	5	[2, 3]	0.4	A21GCOGHDQNHQS	This is a fine stand for the price, but I was honestly surprised at how li…
3946	2	3	5	[2, 3]	0.4	A5ZS85C5V40S8	Musicfriends, This is a fab item – For 20.00 You can't go wrong on this. W…
3969	1	4	5	[1, 4]	0.2	A3L5L7OHXRY03L	Good strings. They work and sound good. I didn't notice much of a difference between these and run-of-the-mill D'Addario strings.
4044	2	3	5	[2, 3]	0.4	A2M8T1A4O0GS72	Nicely made strong strap with colorful pattern. It fits my ovation quite s…

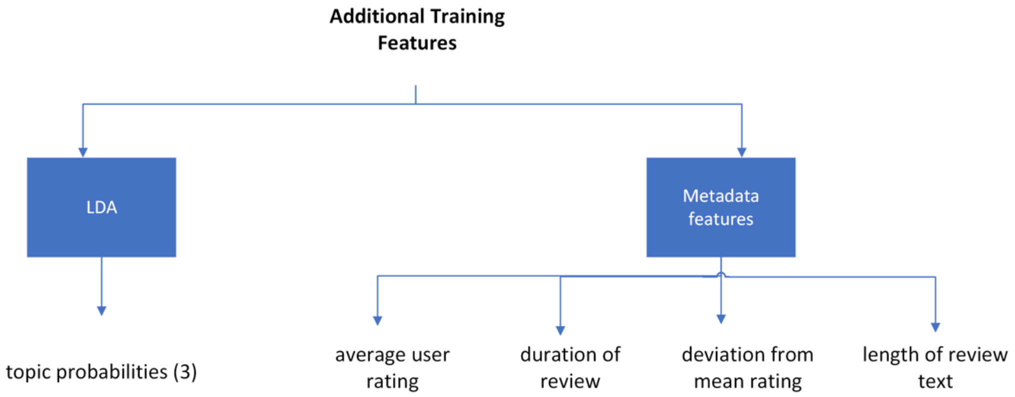
Initial Training dataset



Index	upvote	downvote	total_votes	helpful	quality_scor	reviewerID	reviewText
8008	0	0	0	[0, 0]	nan	A3HCK3UXD6WS4G	These picks are ok if you need them to be on the stiff side. I was looking for a semi-soft felt picks.
8009	0	0	0	[0, 0]	nan	A1CSB9FS3SLHJO	I bought these picks because the description said they were for the ukulel…
8011	0	0	0	[0, 0]	nan	A1T4HGVX32QIYC	I actually use these on my bass if my fingers are too worn from heavy play…
8012	0	0	0	[0, 0]	nan	A22Z554ZQ8NFPC	THis is a 5 star stand, considering the price. Sure, it's not as sturdy as…
8013	0	0	0	[0, 0]	nan	AXABTEYS7A4A8	I bought two of these, one for each of my sons (12 and 14). I wanted somet…
8014	0	0	0	[0, 0]	nan	AWKVQQZKTRFAL	Attaching hardware is a bit cheap but over all it is better than I expecte…
8017	0	0	0	[0, 0]	nan	A1MVH1WLYDHZ49	I bought this for my granddaughter to use at home for her flute practice. …
8018	0	0	0	[0, 0]	nan	A3DDZ2SENG07MS	The stand seems sturdy, the base is like a tripod base of a mic stand, so …
8020	0	0	0	[0, 0]	nan	A1VG6TG6LKBPKP	Sturdy, portable, and the little extra weight is worth the investment if y…
8021	0	0	0	[0, 0]	nan	A2NIT6BKW11XJQ	The music stand is similar to the stands used in schools. This stand is durable, firm and easy to assemble.

Initial Testing dataset

B. Additional features required for training set are extracted using LDA and metadata features



a. Extract topic probabilities using LDA (Latent Dirichlet Allocation)

Python Pseudo code for computing probabilities using LDA:

```
corpus = preprocessed_reviews  #remove stop_words and punctuations, lemmatization
#creating term dictionary of corpus
dictionary = corpora.Dictionary(corpus)
#vectorize dictionary using bag of words
corpus_matrix = [dictionay.doc2bow(i) for i in corpus]
#implementing LDA for topic modeling using gensim
Lda_model = Lda( corpus_matrix, num_topics = 3, id2word = dictionary)
```

After implementing the above code on our reviews corpus, we will get following features for our model:

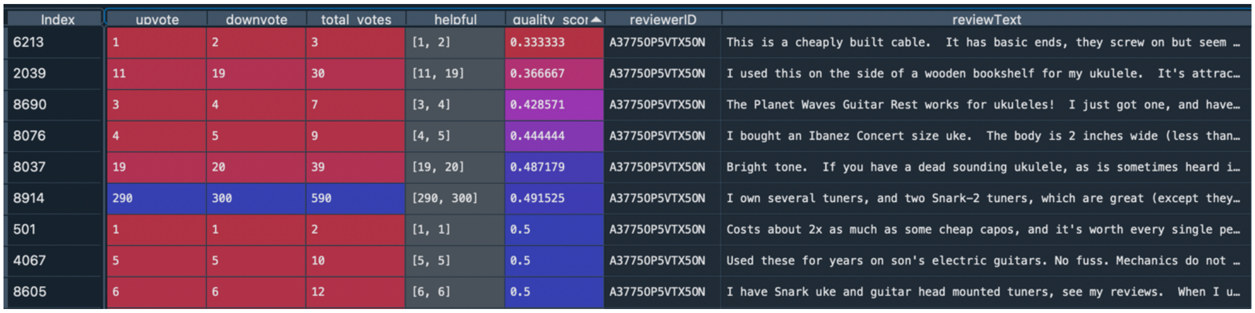
reviewID	topic_1_prob	topic_2_prob	topic_3_prob
A3L0BF1DJ22V9R	0.583456	0.333456	0.234456
A1LHKVHP38M046	0.288933	0.349034	0.589292

b. Extract MetaData features

Following metadata features will be extracted:

i. average user rating

- For the current review, find other reviews by similar reviewer



Index	upvote	downvote	total_votes	helpful	quality_scor▲	reviewerID	reviewText
6213	1	2	3	[1, 2]	0.333333	A37750P5VTX50N	This is a cheaply built cable. It has basic ends, they screw on but seem …
2039	11	19	30	[11, 19]	0.366667	A37750P5VTX50N	I used this on the side of a wooden bookshelf for my ukulele. It's attrac…
8690	3	4	7	[3, 4]	0.428571	A37750P5VTX50N	The Planet Waves Guitar Rest works for ukuleles! I just got one, and have…
8076	4	5	9	[4, 5]	0.444444	A37750P5VTX50N	I bought an Ibanez Concert size uke. The body is 2 inches wide (less than…
8037	19	20	39	[19, 20]	0.487179	A37750P5VTX50N	Bright tone. If you have a dead sounding ukulele, as is sometimes heard i…
8914	290	300	590	[290, 300]	0.491525	A37750P5VTX50N	I own several tuners, and two Snark-2 tuners, which are great (except they…
501	1	1	2	[1, 1]	0.5	A37750P5VTX50N	Costs about 2x as much as some cheap capos, and it's worth every single pe…
4067	5	5	10	[5, 5]	0.5	A37750P5VTX50N	Used these for years on son's electric guitars. No fuss. Mechanics do not …
8605	6	6	12	[6, 6]	0.5	A37750P5VTX50N	I have Snark uke and guitar head mounted tuners, see my reviews. When I u…

- Find average of all the upvotes the user has received on their reviews

  average = 4.5454

ii. duration for which the review has been around

This can be computed using 'reviewTime' column in dataset. Calculated in days.

iii. deviation of rating from the mean rating of the product

- compute mean rating for the product
- calculate deviation of current rating from mean rating

reviewID	asin	rating	mean_product_rating	deviation
A3775OP5VTX5ON	B004Z17008	5	4.8333	0.1667

iv. length of review text

- computed using len(reviewText) in characters

5. The computed features and target variable ‘quality score’ (earlier computed using formula) are used to train a Logistic Regression Model to obtain ‘quality score’ for reviews.

Example of **complete feature space** and **target variable** for training the Logistic Regression Model:

reviewerID	asin	reviewText	topic_1_prob	topic_2_prob	topic_3_prob	Avg_user_rating	reviewDuration	deviation_mean_rating	len_review	quality_score
A3L0BF1DJ22V9R	B004Z17008	This is a cheaply built cable ………	0.583456	0.333456	0.234456	4.5454	220	0.1667	23459	0.491525

Python Pseudo Code for training Logistic Regression Model:

```
x_test = df_without_quality_score

x_train = df_with_quality_score.drop[‘quality_score’]

y_train = df[‘quality_score’]

#defining a Logistic Regression Model

logreg = LogisticRegression()

#fitting the model to training data

logreg.fit(x_train, y_train)

#prediction using testing data

y_pred = logreg.predict(xtest)
```

Using this we would be able to obtain quality score for all the products in the dataset using reviews dataset.

6. The quality score computed are taken as weight and assigned to rating in probabilistic matrix factorization (PMF) framework to predict enhanced rating values.

- PMF is used to infer latent (hidden) factors of users and products from ratings

We start with a dataset similar to following which has user rating and calculated quality score for each product:

	Product 1 Rating	Product 1 Quality Score	Product 2 Rating	Product 2 Quality Score	Product 3 Rating	Product 3 Precision Score
User 1	1	0.4915	4	0.8978	3	0.8978
User 2	4	0.2378	3	0.4915	3	0.5687
User 3	5	0.8978	4	0.3389	1	0.8978
User 4	2	0.3345	1	0.5678	5	0.4915

Python Pseudo Code for Matrix Factorization:

```
#loading the dataset with users, products, ratings and quality score

df = user_products_dataset

#number of users

U = len(df['users'])

#number of products

P = len(df[0])

#number of features (hyperparameter)

K = 3

#creating the user matrix

U = numpy.random.rand(U, K)

#creating the products matrix

P = numpy.random.rand(P, K)

nU, nP = matrix_factorization(df, U, P, K)

#dot product of user and product matrix to obtain matrix with enhanced ratings

 nR = numpy.dot(nU, nP)
```

The result of matrix factorization is a matrix that gives enhanced ratings as follows:

	Product 1	Product 2	Product 3
User 1	2.9989	3.8933	2.2928

User 2	4.2782	4.6722	4.6734
User 3	5	2.1234	1.2293
User 4	3.92020	3.2673	4.9990

Following this multi-step approach including feature extraction, computing quality score and using matrix factorization, we obtain enhanced values for the ratings. This approach can also be used to predict missing user ratings which is useful to build a recommender system using collaborative filtering approach.

8. Final conclusion

Throughout the exercise the sensitivity of all the models towards unbalanced data was quite noticeable. The results, even if not high values, were obtained with a more balanced dataset that allows to further trust the predictors with the best performance

In general, the Machine Learning approaches outperformed the Lexicon approaches with better scores across the board. The Precision and Recall levels seem quite similar but still the results are low values just as with the accuracy achieved. The amount of data used for training the model was not large and this limits the vector representations similarity measures for both Lexicon and Machine Learning approaches.

Specialized libraries and resources such as Spacy and Wordnet allow for rapid development of an entry level sentiment analysis model. With improved data preprocessing and a pipeline to orchestrate the entire process an average model could be initially deployed and be further improved by feedback loops to achieve better results with a larger corpus and more accurate text representations aligned with the business case.

9. Assumptions

- It is assumed that labels in the dataset will always be in English language
- The scores are accurate and correspond to actual comments and ratings provided by real people without the intervention of manipulations of bots for artificial reviews
- Training models either under the Lexicon or Machine Learning approach with the full dataset will cause the models to developed a bias towards positive sentiment given the distribution of the ratings for the input dataset

References:

6.8. pairwise metrics, affinities and kernels. scikit. (n.d.). Retrieved April 19, 2022, from https://scikit-learn.org/stable/modules/metrics.html

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, *25*(2), 99–154. https://doi.org/10.1007/s11257-015-9155-5

Pupale, R. (2019, February 11). *Support vector machines(svm) - an overview*. Medium. Retrieved April 19, 2022, from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

Raghavan, S., Gunasekar, S., Ghosh, J.: Review quality aware collaborative filtering. In: Proceedings of the 6th ACM Conference on Recommender systems, Dublin, Ireland, ACM, RecSys'12, pp 123–130 (2012)

Support Vector Machines (SVM) algorithm explained. MonkeyLearn Blog. (2017, June 22). Retrieved April 19, 2022, from https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

Appendix 1: Project plan.

PROJECT PLAN

PROJECT NAME	Sentiment Analysis Model - Musical Instruments Review	PROJECT MANAGER	Devanshi Shah
PROJECT DELIVERABLE	· Working Code with better accuracy for both Lexicon and Machine Learning-based approach · Report with Design and Research Requirements · Power Point Presentation		
SCOPE STATEMENT	we construct "a sentiment analysis model for products based on customers' textual reviews," using both a Lexicon approach and a machine learning approach		

START DATE	08/03/2022	END DATE	19/04/2022	OVERALL PROGRESS	100 %

AT RISK	TASK NAME	ASSIGNED TO	START DATE	END DATE	DURATION in days	STATUS
X	Read Scope of the Project and carry out initial research	All members	08/03/2022	09/03/2022	2	Competed
X	Dataset Exploration	Shrikant Kale, Jefil Tasna John Mohan	10/03/2022	14/03/2022	5	Completed
X	Text basic pre-processing: A & B	Devanshi Shah, Nestor Romero	10/03/2022	14/03/2022	5	Completed

X	Text basic pre-processing: C & D	Jefil Tasna John Mohan, Shrikant Kale, Hitesh Dharmadhikari	10/03/2022	14/03/2022	5	Completed
X	Text representation	Nestor Romero, Devanshi Shah	10/03/2022	14/03/2022	5	Completed
X	Modeling (Sentiment Analysis) Lexicon approach i) VADR	Shrikant Kale	10/03/2022	14/03/2022	5	Completed
X	Modeling (Sentiment Analysis) Lexicon approach ii) TextBlob	Jefil Tasna John Mohan	15/03/2022	15/03/2022	1	Completed
X	Modeling (Sentiment Analysis) Lexicon approach iii) SentiWordNet	Nestor Romero	16/03/2022	18/03/2022	3	Completed
X	Testing: Test out the model using the 30% test data note the accuracy, precision, recall and F1 score.	All members	22/03/2022	01/04/2022	10	Completed
X	Presentation	All members	22/03/2022	01/04/2022	10	Completed
X	Project report	All members	22/03/2022	01/04/2022	10	Completed
X	Modeling (Sentiment Analysis) Machine Learning approach: i. Logistic Regression	Hitesh Dharmadhikari	22/03/2022	01/04/2022	10	Completed
X	Modeling (Sentiment Analysis) Machine Learning approach: ii. SVM	Shrikant Kale	05/04/2022	05/04/2022	1	Completed
X	Modeling (Sentiment Analysis) Machine Learning approach: iii. Naïve Bayes	Jefil Tasna John Mohan	06/04/2022	08/04/2022	3	Completed
X	Modeling (Sentiment Analysis) Machine Learning approach: iv. Gradient Boostin	Devanshi Shah	11/04/2022	14/04/2022	4	Completed

X	Testing: Test out the two models using the 30% test data note the accuracy, precision, recall, and F1 score	Nestor Romero, Devanshi Shah	11/04/2022	14/04/2022	4	Completed
X	Compare the test results of the Lexicon model versus the two machine learning models.	Jefil Tasna John Mohan, Shrikant Kale	15/04/2022	16/04/2022	2	Completed
X	Research Paper	Shrikant Kale	16/04/2022	19/04/2022	3	Completed
X	Final Report	All Members	16/04/2022	18/03/2022	2	Completed
X	Power Point Presentation	Shrikant Kale, Jefil Tasna John Mohan	18/04/2022	19/04/2022	2	Completed

Appendix 2: Meeting Register

Meeting Date	**Attendees**	**Subjects of Discussion**	**Work Assignment**
08/03/2022	All members	Initial discussion about project scope and work division for Research and Design aspects	Nestor: Text basic pre-processing: A & B Shrikant: Dataset Exploration Devanshi: Text basic pre-processing: A & B Hitesh: Text basic pre-processing: C & D Jefil: Dataset Exploration
14/03/2022	All members	Review work for checkpoint and prepare for presentation	NA
16/03/2022	All members	Discuss technical requirements and divide work	Shrikant: Modeling (Sentiment Analysis) Lexicon approach

			i) VADR Nestor: Modeling (Sentiment Analysis) Lexicon approach iii)SentiWordNet, Text representation Jefil : Modeling (Sentiment Analysis) Lexicon approach ii) TextBlob Devanshi : Testing: Test out the model using the 30% test data note the accuracy, precision, recall and F1 score. Hitesh : Text representation
25/04/2022	All members	Review progress on assigned tasks and address any blockers or changes	NA
04/04/2022	All members	Review the completed work and plan for integration. Discuss flow for checkpoint presentation	NA
08/04/2022	All members	Divide work for Machine Learning approach and other remaining work and enhancements	Hitesh: Modeling (Sentiment Analysis) Machine Learning approach: i. Logistic Regression Shrikant: Modeling (Sentiment Analysis) Machine Learning approach: ii. SVM Nestor:Testing: Test out the two models using the 30% test data note the accuracy, precision, recall, and F1 score Jefil : Modeling (Sentiment Analysis) Machine Learning approach: iii. Naïve Bayes

			Devanshi: Modeling (Sentiment Analysis) Machine Learning approach: iv. Gradient Boosting
14/04/2022	All members	Review progress on assigned tasks and address any issues/Blockers	Jefil Tasna John Mohan, Shrikant Kale: Compare the test results of the Lexicon model versus the two machine learning models. Research Paper : Shrikant Kale Final Report : all member
17/04/2022	All members	Main Project Integration	NA
18/04/2022	All members	Reviewing all the requirements, code, final testing and dividing work for presentation	NA