YouTube Video Comment Sentiment Analysis

Rohith Pattathil (SUID - 27616-8593)
Sitesh Mishra (SUID - 74289-3107)

Social Media and Data Mining (CIS - 600)
Syracuse University
New York
May 2020

Table Of Contents

Abstract

YouTube is one of the richest social media platforms, which has a huge collection of videos and videos are uploaded every second. User interaction on YouTube is observed via the likes, dislikes, comments and share features on videos uploaded by various channels. Over the last decade, YouTube has been the largest user-driven online video provider. Most videos on YouTube have significant user interaction and the work done to observe trends in these comments have been minimal due to the low quality, low information consistency and unstructured nature of the comments. In this paper we have performed sentiment analysis on YouTube video comments using three different types of algorithms. We compare the results from the algorithms used and demonstrate the best one among them. Moreover, we have predicted the future sentiment of the videos based upon the past trends.

Keywords – YouTube, Sentiment Analysis, Vader, Afinn, NRC Lexicon

1 Introduction

Millions of users all around the globe continuously update or add information to social media platforms like Twitter, YouTube or Facebook, these media and streams can affect a person's reputation in a huge way. Thus, automatically extraction sentiments and opinions that people express on social media is very important. While sentiment analysis has attracted a lot of attention from academia as well as industry for more mainstream data, the paucity of manually annotated data makes these studies very hard to use for social media and streams. Just after Facebook, YouTube is the social network which has the world's second largest population.

By the end of January 2020, YouTube has crossed 2 Billion users. Every minute 300 hours of videos are uploaded on YouTube. As per the statistics 7 out of 10 people prefer online video platforms rather than watching live TV and YouTube is the biggest online video streaming platform. Till date about 11,000 YouTube videos have gone beyond 1 Billion views. 30 Million people visit YouTube every single day. The content until February 2020 amounts up to 6.5 million hours viewing (750 years). YouTube video statistics suggest that watching YouTube on TV screens is becoming increasingly popular. As of March 2019, users were watching over 250 million hours of YouTube on TV screens–a 39 percent increase in less than one year. These numbers exclude viewing on Google's internet pay-TV service and YouTube TV [1].

As per the above statistics, it is quite clear how powerful YouTube is in today's timeline. With numerous people following, it is an absolute necessity that the content of YouTube should be suitable for every single user. The only way to verify the content is by analyzing the number of views, likes/dislikes and the comments posted by the viewers. Our goal is to pick a specific YouTube channel and extract the comments from some of its latest videos and apply sentiment analysis to figure out how the recent content is being perceived by the viewers.

With the onset of the COVID-19 pandemic and restrictions in place forcing people to stay at home, YouTube witnessed a huge spike in viewing traffic, so much so that it had to reduce the default streaming quality. Such a huge viewing time generates exponential ad revenue. Hence, motivating many people to pursue YouTube as a career option.

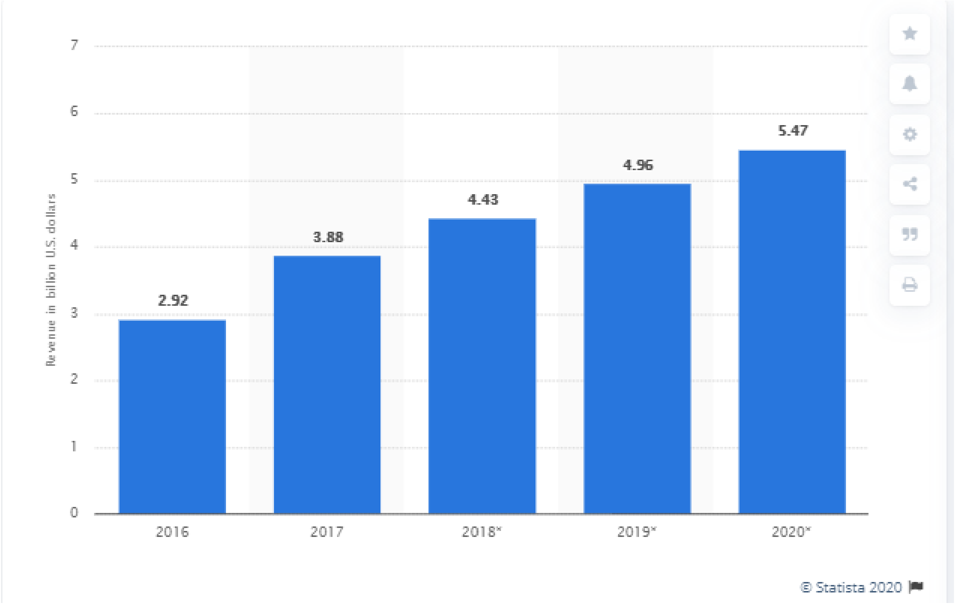

Figure I: YouTube Ad revenue from US alone

2 Prior Work

The paper "Survey on mining subjective data on the web" [14], was used to understand the different Sentiment Analysis techniques available to our disposal. Once we found techniques that are most suitable to our application, we researched on each of them in detail and we will cite the most critical papers, webpages and journals in this subsection.

1. The webpage, "Simplifying Sentiment Analysis using VADER in Python [12]. This article provided a lot of fundamental insights on how VADER performs sentiment analysis and how it can be used on Social Media Text.

2. The paper, " A new ANEW: Evaluation of a word list for sentiment analysis in microblogs" [11]. This research paper was referred to understand how AFINN-96 was developed and how it can contribute to our research. Researchers progressively use Amazon Mechanical Turk (AMT) for creating labeled language data and both VADER and AFINN used it.

3. The paper "Crowdsourcing a Word-Emotion Association Lexicon" [10]. This paper was used to understand how NRC Lexicon was built and how it can be used to classify words into emotions.
   The most crucial articles that were referred to for understanding different techniques

of Time Series prediction are mentioned below,

4. The article “Implementing Linear and Polynomial Regression From Scratch” [13] provided a comprehensive guide on how to implement Linear and Polynomial Regression

5. The webpage “Time Series Analysis, Visualization & Forecasting with LSTM” [9]. This article helped comprehend how LSTM can be used for Analysis and forecasting Channel Sentiment trend.

3 Process Flow

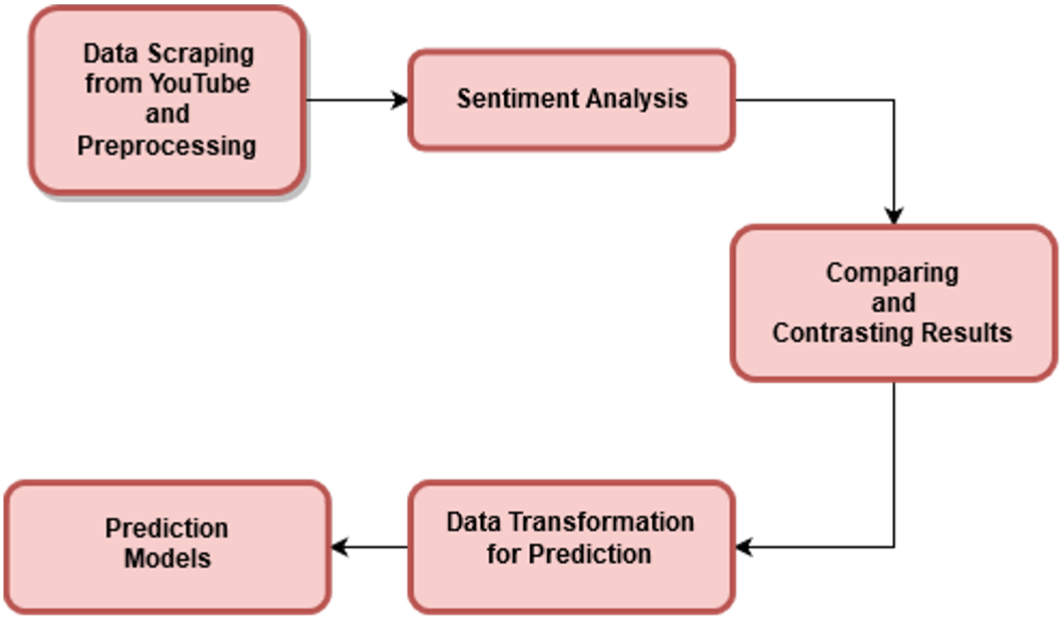
Data Scraping from YouTube and Preprocessing
Sentiment Analysis
Comparing and Contrasting Results
Data Transformation for Prediction
Prediction Models

Figure II: Project Process Flow

4 All About Data

The YouTube Data API lets you incorporate functions normally executed on the YouTube website into your own website or application. The API also supports methods to insert, update, or delete many of these resources [2]. The API makes use of OAuth authentication in order to authenticate the user upon a new session and then links to the account that has been used to authenticate it to check for access privileges and manage the daily limit quotas.

4.1 Calling the API

Below are some necessities for requesting data from this API:

1. Each request must specify an API key or provide and OAuth 2.0 token. These can be obtained from the developer console.
2. An authorization token is a must for every update, delete or insert request. The token is also a must for any request that gets the user's private data
3. The API supports the OAuth 2.0 authentication protocol

4.2 More about the API

Version 3 of the YouTube Data API has concrete quota numbers listed in the Google API Console where you register for your API Key. You can use 10,000 units per day. Projects that had enabled the YouTube Data API before April 20, 2016, have a default quota of 50M/day [2].

- A simple read operation that only retrieves the ID of each returned resource has a cost of approximately 1 unit.
- A write operation has a cost of approximately 50 units.
- A video upload has a cost of approximately 1600 units.

If you hit the limits, Google will stop returning results until your quota is reset. You can apply for more than 1M requests per day, but you will have to pay for those extra requests.

4.3 API Methods Used

4.3.1 Search

This method is used to get the channel id from the YouTube display name the user inputs. We sort the results by relevance and pick the most relevant search the query returns. After we fetch the channel id now, we perform a search on a particular channel's videos. The videos are fetched in chronological order and are limited to the number that is set in the configuration file. Now we have the videos of the channel, we need the comments for each of the videos. It Returns a collection of search results that match the query parameters specified in the API request. By default, a search result set identifies matching video, channel, and playlist resources, but you can also configure queries to only retrieve a specific type of resource [2].

4.3.2 Comment threads

After we have the video list containing the video information of the channel we selected, we now need to extract comments for each of them. For every video we call this API to return comments which are limited to a max of 100 per call. Then, we repeatedly fetch the next page until we get the desired number of comments. The number of comments that will be fetched per video can be changed in a constants file which has all the application configuration settings. From each of the items the API returns we extract the original comment text and the date when it was last updated and create a new JSON file which we will use for both sentiment analysis and for predictions.

4.3.3 Video Statistics

We use this API to fetch the current statistics for each of the video id that we pass in as a parameter. Using this we get statistics of the video like the comment count, like count, dislike count and views which come in handy when we try and find a relation in between the sentiment and the likes and dislikes a video has.

4.4 Where does the data go?

All of the data we fetch is stored in a JSON file which is unique per channel, this allows us to persist data rather than just scrape all the time. Using these JSON files we can compare and contrast the various models and algorithms we needed to use without going through the painful process of scraping comments each time and running into issues with exceeding the daily limit. In addition, once we have hit the limit, we can just continue to scrape the next day from where we stopped due to the limit and simply add to this JSON file. This makes it easier to manage the data, add to it and as a well as manipulate it. Along with the ease of updating and modifying it also comes in handy when we want to read this data into a DataFrame as it can simply be done using pandas inbuilt method readjson.

We end up creating 3 JSON files per channel

1. Video list – Contains a list of videos that we fetched initially
2. Comment Scores – Contains the score per comment that the different models give us
3. Stats – This file contains a per video stat list of all the videos in addition to the overall average sentiment for each video.

5 Sentiment Analysis

5.1 What is it?

”Sentiment analysis is the process of determining whether a piece of writing is positive, negative or neutral” as defined by the webpage Sentiment Analysis Explained [3] . It combines NLP and machine learning techniques to assign appropriate weighted sentiment scores to the themes, topics, entities as well as categories within a sentence or phrase. Data analysts across the globe in large enterprises use sentiment analysis to understand public opinion, perform market research, keep track of product as well as brand reputation, and understand customer experiences. Basic sentiment analysis of text documents follows a straightforward process [3]:

- Break each text document down into its component parts (sentences, phrases, tokens and parts of speech)

- Identify each sentiment-bearing phrase and component
- Assign a sentiment score to each phrase and component (-1 to +1 or -5 to +5 in some algorithms)
- Optional: Combine scores for multi-layered sentiment analysis

There are several uses of sentiment analysis in different industries. It allows businesses to identify customer sentiment toward products, brands or services in online conversations and feedback. Sentiment analysis models can flag any situation which is not expected and thus enable you to take action right away. The process of tagging text by sentiment is very subjective and is easily influenced by ones thoughts, beliefs and even personal experiences. Using a centralized sentiment analysis system, companies can improve their accuracy and get better insights using the same unbiased criteria to evaluate all of their data

5.2 Preprocessing

Text is the most unstructured form and thus involves a lot cleaning. These pre-processing steps help convert noise from high dimensional features to the low dimensional space to obtain as much accurate information as possible from the text [4]. We have applied following techniques before actually feeding the data to sentiment analysis models.

5.2.1 Tokenization

For a given character sequence Tokenization is the process of transforming the text into pieces, called tokens. At this time certain other unnecessary characters such as punctuation and other special characters are removed. Some sentiment analysis algorithms do not comprehend emoticons as well so for better performance emojis are also dropped.

5.2.2 Stopwords

The most commonly occurring words that do not contribute majorly to the context of the data are called Stopwords. They generally do not add any value to the sentence meaning. These stopwords are usually removed before sentiment analysis is performed.

5.2.3 Stemming

Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language [8]. We have used NLTK python library, which is the one stop solution for tokenizing, stemming and removing the stop words from the text.

5.3 Models and Methodologies Used

5.3.1 Vader

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a sentiment analysis tool that is specifically attuned to sentiments expressed in social media using lexicon and rule-based analysis. It makes use of a combination of sentiment lexicon, which is a list of lexical features (e.g., words) which are usually labelled according to their semantic orientation as either negative or positive.

VADER has been quite successful when judging the sentiment of social media texts such as YouTube comments, Tweets, FB posts etc. The reason behind VADERs success is that even for text having slangs, punctuations, emoticons or unstructured text, it is able to judge how positive or negative a sentiment is.

Below you can see a snapshot from *VADER's* lexicon, where more positive words have higher positive ratings and more negative words have lower negative ratings [5].

Word	**Sentiment rating**
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

Most ratings in VADER are obtained from Amazon's Mechanical Turk, which is both quick and cheap. It checks each word in a sentence if it is available in the lexicon. For example, the sentence "The food is nice and the atmosphere is good", there are two words "good" and "nice" in this sentence, which is found in the lexicon with a rating of 1.9 and 1.8 respectively. As we can see below, VADER produces four sentiment metrics from these word ratings. It categorises each sentence into positive, neutral and negative. This example was rated as 55% neutral, 45% positive and 0% negative. The fourth metrics is the compound score, which represents the sum of all the ratings(1.9 and 1.8 in this example), which is normalized into a score, that is in the range -1 to 1. In the given example, the sentence has a compound rating of 0.69, which is strongly positive [5].

Word	**Sentiment rating**
Positive	0.45
Negative	0.55
Neutral	0.00
Compound	0.69

Why *VADER* outperforms?

The production of lexicons is extremely time consuming and very expensive thus they are rarely updated meaning they lack the current age slangs which might be in any dictionary.

VADER analyses sentiments primarily based on below key points, which enables it to precisely predict the sentiments of the text [4]:

- Punctuation: The use of an exclamation mark(!), increases the magnitude of the intensity without modifying the semantic orientation. For example, “The food here is good!” is more intense than “The food here is good.” and an increase in the number of (!), increases the magnitude accordingly.
- Capitalization: Using upper case letters to emphasize a sentiment-relevant word in the presence of other non-capitalized words, increases the magnitude of the sentiment intensity. For example, “The food here is GREAT!” conveys more intensity than “The food here is great!”
- Degree modifiers: Also called intensifiers, they impact the sentiment intensity by either increasing or decreasing the intensity. For example, “The service here is extremely good” is more intense than “The service here is good”, whereas “The service here is marginally good” reduces the intensity.
- Conjunctions: Use of conjunctions like “but” signals a shift in sentiment polarity, with the sentiment of the text following the conjunction being dominant. “The food here is great, but the service is horrible” has mixed sentiment, with the latter half dictating the overall rating
- Preceding Tri-gram: By examining the tri-gram preceding a sentiment-laden lexical feature, we catch nearly 90 percent of cases where negation flips the polarity of the text. A negated sentence would be “The food here isn’t really all that great”.

5.3.2 Afinn

AFINN is an English word listed developed by Finn Årup Nielsen. Words scores range from -5(most negative) to +5 (most positive). The English language dictionary in the current version of the lexicon is AFINN-en-165.txt which contains over 3,300+ words with a polarity score associated with each word. Unlike Vader, in which no pre-processing of data is required, Afinn requires data cleaning and pre-processing of data before actually running the sentiment analysis.

5.3.3 NRC Lexicon

The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) [10].The Sentiment and Emotions include different manually created and automatically created lexicons

In our project we are using NRC Word-Emotion Association Lexicon aka NRC Emotion Lexicon aka EmoLex (Ver0.92) which it the association of words with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) manually annotated on Amazon’s Mechanical Turk. Available in 40 different languages. It has 14,182 unigrams (words), 25,000-word senses [10].

5.4 Results and Findings

We executed our project on many YouTube channels of different genre and on different magnitude of data. For the result analysis we scraped approximately 500,000 comments for 3 different channels – *Dude Perfect*, *I Hate Everything* and *Unbox Therapy.*
We juxtaposed the sentiment analysis result of Vader for above channels to compare and contrast our findings.

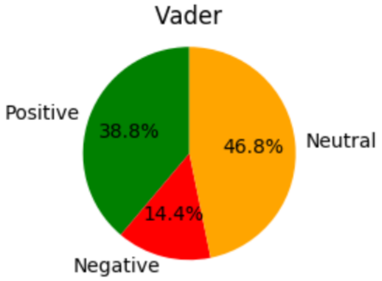

Figure III: Dude Perfect

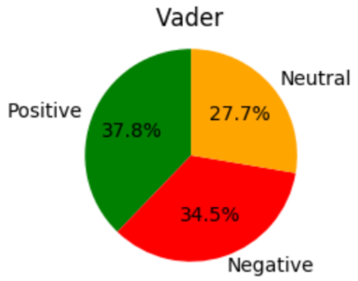

Figure IV: I hate everything

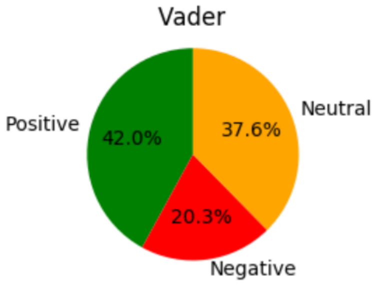

Figure V: Unbox Therapy

In the above pie charts, we can see that the *Unbox Therapy* has most positive comments. On the other side, number of positive sentiment comments are approximately same for *Dude Perfect* and *I Hate Everything*. Things gets interesting when we see negative sentiments. *I Hate Everything* has approximately 35% negative comments which is much higher as compare to 14% and 20% for *Dude Perfect* and *Unbox Therapy* respectively

Another reason for choosing these specific channels is that their video contents are completely different. This can be evidently seen through the word clouds generated with most frequent words used in the comments of these channels.

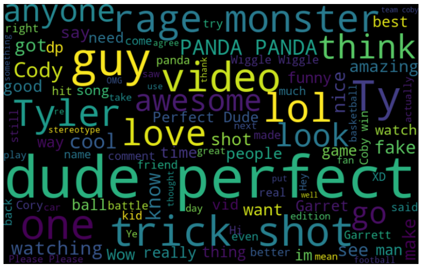

Figure VI: Dude Perfect

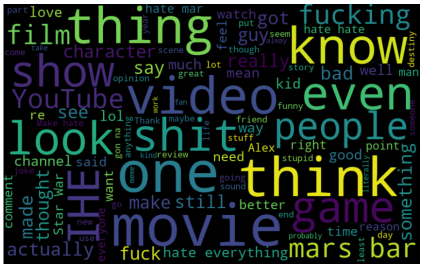

Figure VII: I Hate Everything

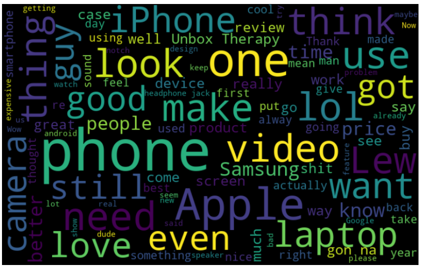

Figure VIII: Unbox Therapy

Now let's see the line plot which will project sentiments polarity against each video for a particular YouTube channel.

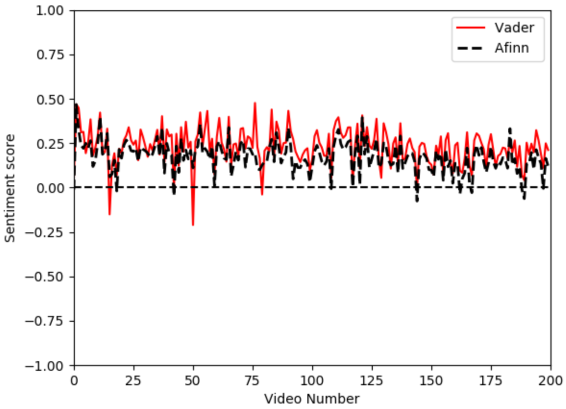

Figure IX: Dude Perfect

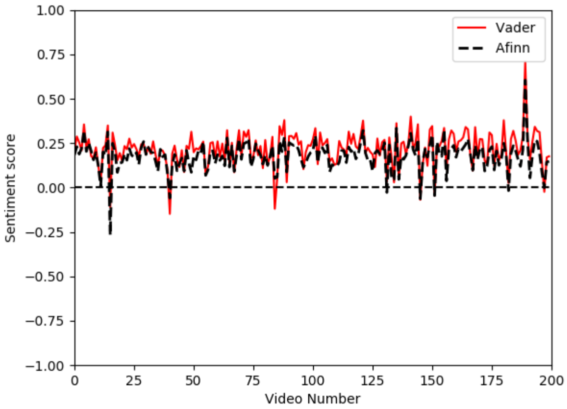

Figure X: Unbox Therapy

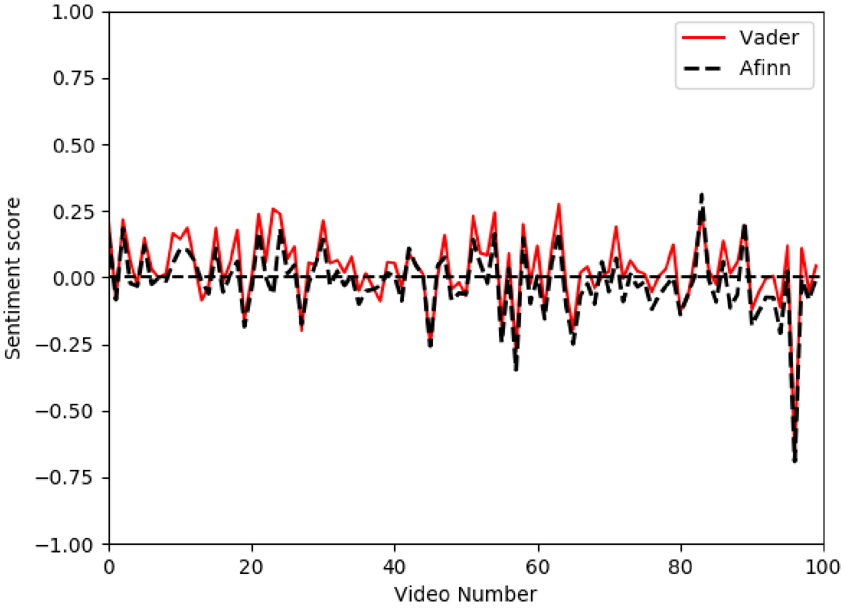

Figure XI: I hate Everything

Line graph also shows the similar result. On careful observation we can see average sentiment polarity of *Dude Perfect* and *Unbox Therapy* is approximately 0.25. However, for *I Hate Everything* its approximately 0. From the above result we can conclude that *Dude Perfect* and *Unbox Therapy* has more positive sentiment as compared to *I Hate Everything*, that's why channel *I hate Every thing* has way too less subscriber when compared to *Dude Perfect* and *Unbox Therapy*. And in future also *I Hate Everything* will grow at much slower rate in terms of subscriber count and view count as compared to *Dude Perfect* or *Unbox Therapy*. Let us now focus on comparing the various sentiment models we used. As described in above section we used 3 different sentiment models – Vader, Afinn and NRC Lexicon. Figure below shows pie chart of Vader, Afinn and NRC lexicon for Dude Perfect.

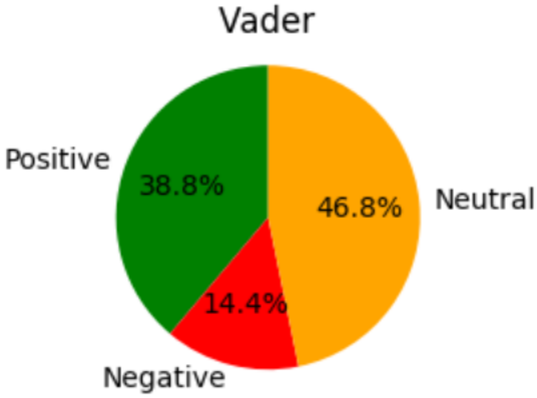

Figure XII: Dude Perfect Vader

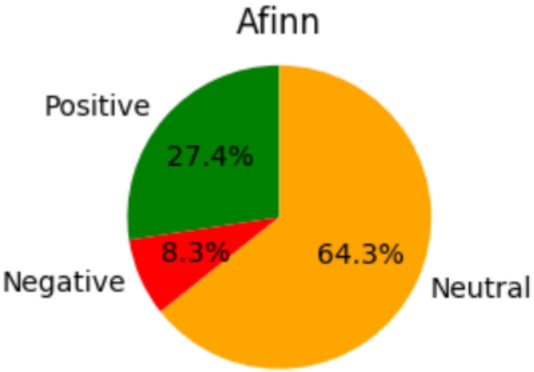

Figure XIII: Dude Perfect Afinn

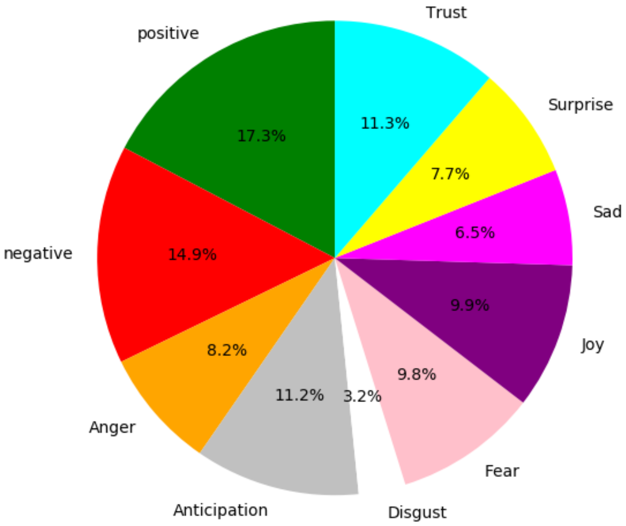

Figure XIV: Dude Perfect NRC

Note that on the same data set, Vader showed more comments as positive, while for Afinn its quite less, which is major gap in terms of analysis. For neutrality it's just the opposite. When we tested it on different dataset, we found that this is always the case with Afinn. The reason for Afinn to be biased towards neutral sentiment is that it processes a sentence word by words and ignores the punctuation and emojis, while Vader considers punctuation, emojis, CAPS character and complex combination of words with non-alphabet characters while assigning the polarity score. For example, for a comment "I love this video" it has the same sentiment score for both Vader and Afinn. But when someone comments "I love this video!!!" or "I love this video ♡♡♡" with a heart emoji or "I **LOVE** this video", the sentiment for these particular comments were different. While Vader chooses to take care of the "!!!" , "heart emojis ♡♡♡" and "LOVE", Afinn simply ignored them.

The result of NRC gave us the insight how comments are actually scored on actual human emotions rather than just Positive, negative and neutral. Note that negative comments percentage is almost same as we found with Vader or Afinn. Vader and Afinn choose to categorise comments/text as neutral if not positive or negative. NRC lexicon goes a step ahead and maps a comment/text to its actual human emotion. Below line graphs resonates same finding which we discussed in above section. Interesting point to observe it the NRC lexicon graphs. How it scores each comment on different emotion parameters.

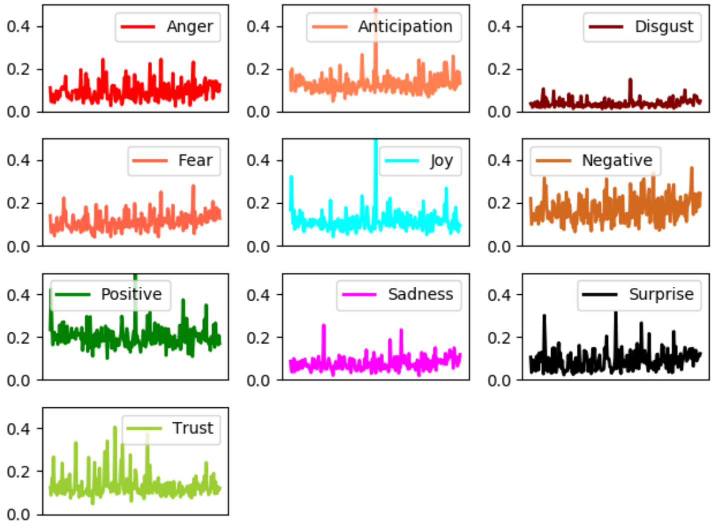

Figure XV: Dude Perfect NRC Line Graph for Different Emotions

5.4.1 Which is the Best?

By now we already had a sense that Vader is doing a better job in handing social media data like YouTube comment, which is full of punctuation and emojis. To assert our intuitions, we performed a baseline performance analysis. We labelled a total of 1200 YouTube comments from different channels as positive, negative and neutral. We kept ratio of positive, negative and neutral equal, to avoid any biased results.

We executed Vader and Afinn on our baseline data. The results of baseline analysis showed that Vader performed with an accuracy of approx. 78% which is better than 70% accuracy of Afinn. The results are as we expected.

Model	Accuracy
Vader	78.79%
Afinn	70.88%

Why not create baseline for NRC?

Note that we kept NRC out of scope for baseline analysis. We would have loved to perform such analysis for NRC lexicon. but we didn't find any Labelled Data for YouTube comments, NRC lexicon provides extensive labelled data for twitter. However, performing baseline performance analysis for YouTube comments on twitter data would not have been justified, since the context of twitters are totally different from YouTube.

We thought of labelling YouTube comments for using it for base line analysis. But labelling even 100 comments against such wide spectrum of emotions is a time-consuming process and very subjective to individuals choice.

6 Prediction Models

6.1 What we Aim?

Our goal in this project is to make a reasonable prediction on the future sentiment of a YouTube channel. This will in turn help channel owners to alter their contents to keep the viewers happy. The fundamental requirement for such a prediction is Time-Series data. We will then use this data to train the various Machine Learning and Neural Network algorithm. In order to create such a dataset, we performed a data transformation, which will discuss in great detail in the following section.

6.2 Data Transformation

The Polarity scores received from Vader and Afinn sentiment analysis are now transformed into a Time Series dataset. This transformation is achieved by grouping the dates on which comments were posted, for better illustration, let us consider the below example. The comments picked for this example are from the same date, The Vader polarity score as well Afinn score on a certain date is averaged to obtain the sentiment on that particular date. We used a lambda function and group by to create this dataset with features such as date, average polarity score, average Afinn score and number of comments. This data set is now used for the time series predictions which we will discuss in the next section.

Comment	**Date**	**Polarity Vader**	**Afinn Score**
I love the new merch it is the best cloths i have ever warn	4/25/2020	0.8402	4
Love the call for positivity when there's a lot of negativity these days	4/25/2020	0.6369	3
dude thank you so much for addressing the camaro back. Nobody i know thinks it does but i've seen thaat ever since it was visuallt announced.	4/25/2020	0.1901	2
If I win the 10 grand any chance I can spend it on a motorcycle instead? I want supercar performance for the $$:) that would be sick	4/25/2020	0.8316	6
Parker needs to take time off and focus on getting his head screwed on properly.	4/25/2020	-0.4939	0
		Mean : 0.40	**Mean : 3**

6.3 Models and Methodologies Used

6.3.1 Linear Regression

One of the most basic Machine Learning Algorithm is the Linear Regression. This model returns a condition that identifies the relationship between the independent variables and the dependent variables. The equation for linear regression can be written as follows

$$Y = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + .. + \delta_n x_n + \epsilon \tag{1}$$

where, Y is the dependent variable, $\mathrm{x_1}$,$\mathrm{x_1}$,$\mathrm{x_1}$ ..$\mathrm{x_n}$ represents the independent variables, $\delta_1, \delta_2, \delta_3..\delta_{\mathrm{n}}$ represent the weights and ϵ is the unobserved random error. Since the degree of this equation is 1, linear regression always plots a straight line. In many cases including stock market predictions linear regression may not hold, which is why we improve this condition by increasing the degree of the polynomial.

6.3.2 Polynomial Regression

Polynomial regression is another kind of regression where the degree of the equation is greater than 1 in contrast to linear regression. Polynomial regression fits a nonlinear relationship between the value of x and corresponding value of conditional value of y, denoted by $E(y|x)$, has been used to describe nonlinear phenomena. Although polynomial regression fits a nonlinear model to the data, as an estimation problem it is linear, in the sense that the regression function $E(y|x)$ is linear in the unknown parameters that are estimated from the time series data. This the reason why polynomial regression is considered an exclusive case of multiple linear regression

In most applications of Polynomial Regression, we model the value of y, the dependent variable as an n_{th} degree polynomial, which gives as the basic equation for Polynomial regression model,

$$Y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + .. + a_n x^n + \epsilon \tag{2}$$

This equation is also considered linear from the estimation point of view since the regression function is linear considering unknown variables a_1,a_2,..a_n. We also consider x^1, x^2, ..., x^n as independent variables in this type of regression.

6.3.3 Long Short Term Memory

Long Short Term Memory is an advanced part of Artificial Neural Networks, which is also recurring. In this type of Neural Network the previous state is preserved. The main difference between recurring Neural Network and LSTM is that, RNNs have involvement with short term dependencies while LSTM is long term. This is the reason LSTM was chosen for stock market prediction in multiple researches and also is a part of this project. The stock market prediction depends upon large amount of data and is dependent on long term history of the company. Therefore, LSTM calculates error by using RNNs with long term memory which is the reason for its higher accuracy rates. Long Short Term Memory can be understood by the below illustration. LSTM has an remembering cell, an input gate, an output gate as well as an forget gate. The cell remembers the long term propagation and the gates are used to regulate the cell.

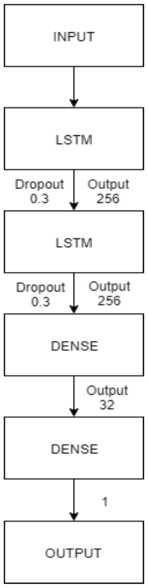

Figure XVI: Illustration on Long Short Term Memory

6.4 Results and Findings

To conclude our analysis of linear regression, polynomial regression and long short-term memory on three different type of channels. The choice of these channels was made in such a way that one received a majority of negative sentiment comments i.e. “I Hate Everything”, the second received majority of positive sentiment comments i.e. ”Unbox Therapy” and last received both positive and negatives sentiment comments i.e. “Dude Perfect”, over a period of time .This selection was done to see the performance of the Machine Learning and Neural Network Algorithms on different types of dataset (Channels). We will now look at our results for each of the algorithms used.

6.4.1 Linear Regression

On using Linear Regression in the sentiment scores for Afinn, it gave a RMSE of ***0.1590*** on the other hand for Vader, it gave a RMSE of ***0.085*** on an average over the three chosen channels. Based on these statistical figures we can conclude Vader outperformed Afinn while using Linear Regression. One of the reason Linear Regression did not meet our expectations is because it oversimplifies the problem and tries to fit all the variables in a single linear equation.

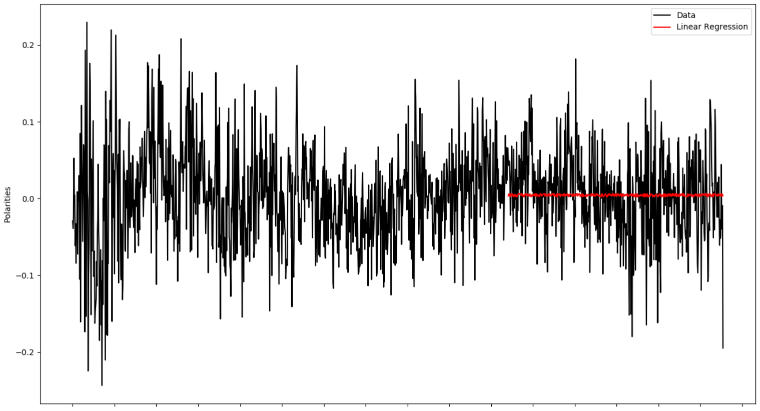

Figure XVII: Linear Regression on channel I Hate Everything (Vader)

On further analysis of the graph it was also observed that, when there are sharp variations in values, Linear regression finds it difficult to fit a straight linear line and thus we observe a huge variation in actual and predicted values. This is evident in figure 14.

On the other hand, when the graph makes slow trend changes, Linear Regression tries to fit the line, makes better predictions, and understandably gives better RMSE. This can be distinctly observed in the figure 15.

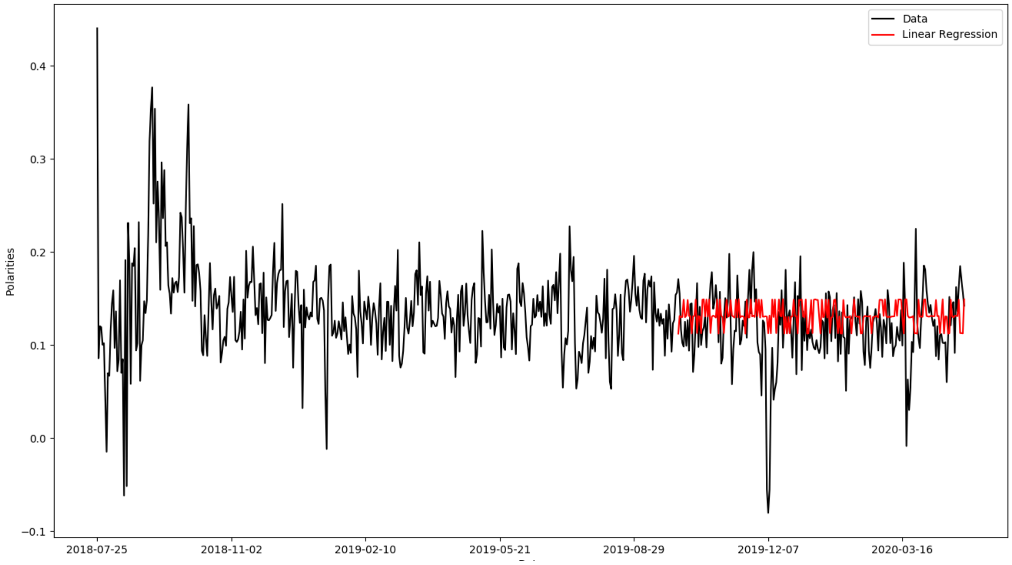

Figure XVIII: Linear Regression on channel Unbox Therapy (Vader)

Although Vader performs better on comparison to Afinn, Let us take a look at the graphs for Afinn.

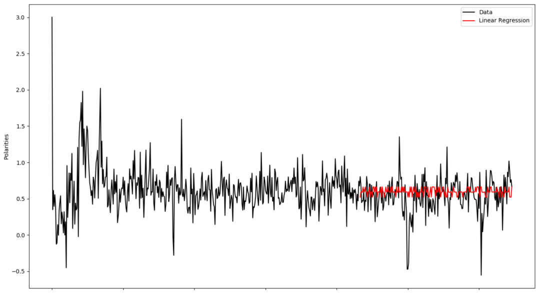

Figure XIX: Linear Regression On Unbox Therapy (Afinn)

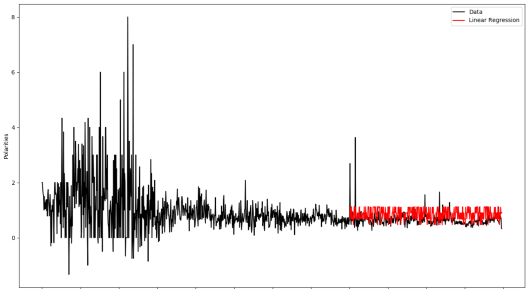

Figure XX: Linear Regression on Dude Perfect (Afinn)

6.4.2 Polynomial Regression

On using Polynomial Regression in the sentiment scores for Afinn, it gave a RMSE of ***0.1497*** on the other hand for Vader gave a RMSE of ***0.0823*** on an average over the three chosen channels. Even in the case of Polynomial Regression, based on the statistical figures Vader outperforms Afinn. Although Polynomial Regression has a higher degree of the equation (4 for this project) in comparison to Linear Regression, It still finds it difficult to fit a line when there are sharp variations in values. The figure (fig no) illustrates this observation.

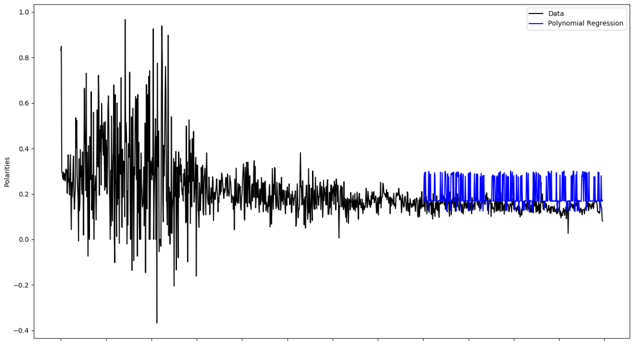

Figure XXI: Polynomial Regression On Channel Dude Perfect (Vader)

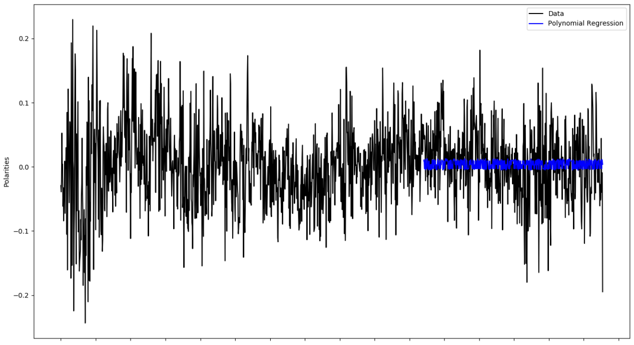

Figure XXII: Polynomial Regression on Channel I hate everything (Vader)

6.4.3 Long Short-Term Memory

On using LSTM in the sentiment scores for Afinn, it gave a RMSE of ***0.0780*** on the other hand for Vader gave a RMSE of ***0.0477*** on an average over the three chosen channels. Consistent the outcomes of Linear and Polynomial Regression, Vader performs better than Afinn in the case of LSTM as well.

LSTM being a Recurring Neural Network Algorithm, is far more superior on comparison with Linear and Polynomial Regression and performs exceptionally well in case of both slow and fast trend changes. This can be observed evidently in the below graphs.

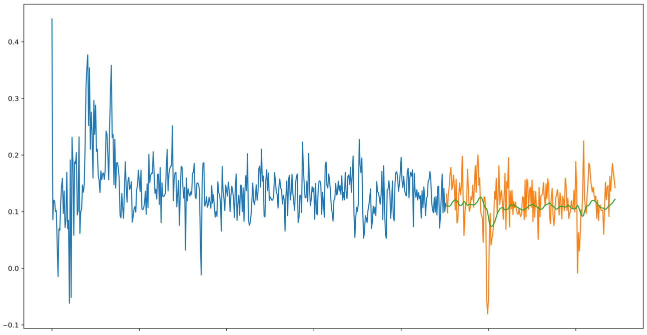

Figure XXIII: LSTM (Vader) on channel Unbox Therapy

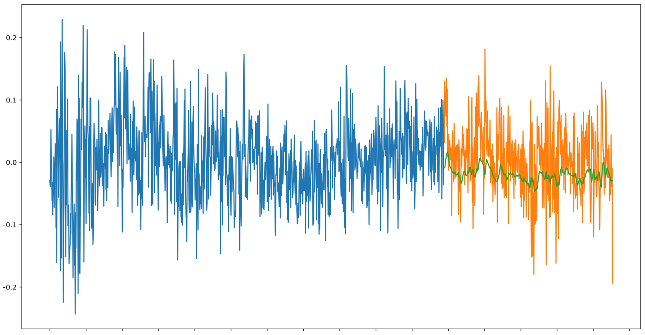

Figure XXIV: LSTM (Vader) on channel I hate everything

Based on the above mentioned statistical and visual facts, we can conclude Vader outperforms Afinn in all three algorithms. We can also conclude LSTM achieves better results on comparison with Linear and Polynomial Regression.

7 Additional Analysis

7.1 Are comments representative of the likes and dislikes?

The common belief is that the like/dislike ratio can project how the sentiment of the comments is. We were curious about this too, so we decided to see if they are actually related. We used the sentiment analysis in conjunction with the view count and comment count of a video to try and predict the likes/dislikes ratio. After the data was attached along with the video statistics, we tried to use machine learning algorithms to accurately predict the like/dislike ratio based on the comment sentiments and other key video stats. We used complex multi-layer neural networks to try to predict the ratio with reasonable accuracy. Our results were very interesting as will be discussed in further sections.

7.2 Models and Features Used

7.2.1 Models

1. MLP Regressor - As defined by Wikipedia, A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training [6]. MLPs are useful in research for their ability to solve problems stochastically, which often allows approximate solutions for extremely complex problems like fitness approximation.

2. Keras - There are two types of built-in models available in Keras: sequential models and models created with the functional API. In addition, you can also create custom models that define their own forward-pass logic [7]. We have used the sequential class of keras models which are created using keras_model_sequential() which compromises

of a set of linear layers. We used 10 rounds of K-fold cross validation to help improve how well the algorithm would perform.

7.2.2 Feature Selection

We are trying to find out a relation in between the comment sentiment and the likes/ dislikes a video has. To do this we fetched the statistics of every video we performed sentiment analysis on. After we used the 3 different models mentioned above, the results of each of them were combined with other important video features like view count, comment count and upload date. Then for each of the sentiment analysis models we took the analysis performed and combined it with stats of each of the videos.

The fore mentioned prediction models were used for each of the 3 different sentiment analysis results, with a goal to predict the ration of likes/dislikes using the comment count, view count and the sentiment of the comments. We split the data into train and test with 70 percent being used to train the data and then fed in the test set to check how well our models performed. The results to say the least were quite interesting, these will be discussed in the next section.

7.3 Findings

Taking the sentiment score output by each of the 3 algorithms we used in addition with the video view count and comment count, we tried to predict the like/dislike ratio of the video. Below were the results we observed.

Sentiment Analysis Algorithm	**Neural Network**	**RMSE**
Vader	MLP Regressor	44.31
Vader	Keras Sequential Regressor	24
Afinn	MLP Regressor	43.90
Afinn	Keras Sequential Regressor	20
NRC	MLP Regressor	42.13
NRC	Keras Sequential Regressor	18

As we can see the comment sentiments do not do a good job of representing the like dislike ratio. A RMSE of 20 even after we use a complex neural network when predicting the ratio of likes and dislikes is a huge difference. Thus, we can conclude that the likes of a video does not represent any obvious trends for comment sentiment. The monetary, subscriber gains a YouTube channel has depends on both these aspects, thus if we were able to get our hands on historical like dislike data we could use a combination of all the features to predict how a channel would grow in popularity in the upcoming times.

8 Conclusion

This paper illustrates the Sentiment analysis of a YouTube channel, which is provided as input by the user. Along with the Sentiment analysis, it discusses about reasonable prediction of viewer sentiments in the upcoming days. We have analyzed a three different channel in terms of their content, scraping about 500,000 comments for each of them. We have used Vader, Afinn and NRC Lexicon for the sentiment analysis. After analyzing these methods on baseline data as well as live YouTube comments, we observed that Vader constantly performs the best. We have represented our results using different visualization techniques such as line plots, and pie charts as well as statistical figures.

After performing sentiment analysis on comments of these channels, the data is then grouped based on date to form a time-series dataset. Then we leveraged Linear Regression, Polynomial Regression and an Recurring Neural Network Algorithm - LSTM to predict the future sentiment of the channel. As per the analysis, We can conclude that LSTM performs the best with an error percentage of 4.7%, this was followed by Polynomial Regression with an error percentage of 8.2% . Linear regression did not match our expectations as it oversimplifies the problem and the best results achieved on an average gave an error percentage of 8.5%. Apart from these findings, we also determined there is minimal co relation between like/dislike ratio and the sentiment of the comment, as even when we tried to use a complex multi-layer neural network the best RMSE we could achieve was 18, which is huge gap when trying to predict a ratio.

From all the above mentioned findings and observation, we can conclude that Vader performs the best sentiment analysis which resonates with our baseline analysis, This is backed up by the fact that when we use Polynomial Regression or LSTM in conjunction with the sentiments predicted by Vader we get the best predictions.

Though we have completed our work here and concluded with some interesting predictions and results, the following section clearly shows that there are much more things to work for in this area. We will continue work in this area and hopefully find more interesting and surprising yet valuable results.

9 Future Scope

In this paper, we have used YouTube API v3 to retrieve data from YouTube. The data API has limited scope in the data it can scrap. However, if you want to dive deeper into analyzing the sentiments or predict the future response of a YouTube channel, the data API isn't good enough. YouTube Analytics API comes to picture in this case. The YouTube Reporting and Analytics APIs let you retrieve historical YouTube data to automate complex reporting tasks, build custom dashboards, and much more.

The future scope for this paper is solely based on the access to YouTube Analytics API. Through the Analytics API, you can retrieve the data from vault for a particular channel. Data such as change in subscriber count per day, like/dislike count, and average playback time of videos can be fetched for a particular channel. Based on the past trends, we can train Machine Learning models in order to figure out what kind of response the channel is going to get in the future from the viewers, in terms of subscriber gains or a sudden increase in popularity. This can help the channel owner realize how the content is perceived by the viewers and look at the future trends they are likely to observe. With average playback time at our disposal in addition to comments, likes and dislikes, we can try our hands on predicting the monetary gains a channel may have.

Bibliography

[1] 40 YouTube stats and facts to power your 2020 marketing strategy : https://sproutsocial.com/insights/youtube-stats/.

[2] YouTube Data API Overview — Google Developers: https://developers.google.com/youtube/v3/getting-started.

[3] Sentiment Analysis Explained: https://www.lexalytics.com/technology/sentiment-analysis.

[4] Sentiment analysis of reviews: Text Pre-processing : https://medium.com/@annabiancajones/sentiment-analysis-of-reviews-text-pre-processing-6359343784fb.

[5] Using VADER to handle sentiment analysis with social media text : http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html.

[6] Multilayer perceptron : https://en.m.wikipedia.org/wiki/Multilayer_perceptron.

[7] About Keras Models : https://keras.rstudio.com/articles/about_keras_models.html.

[8] Bart Jongejan and Hercules Dalianis. Automatic training of lemmatization rules that handle morphological changes in pre-, in-and suffixes alike. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 145–153. Association for Computational Linguistics, 2009.

[9] Susan Li. Time series analysis, visualization forecasting with lstm, May 2019.

[10] Saif M. Mohammad and Peter D. Turney. Crowdsourcing a word-emotion association lexicon. 29(3):436–465, 2013.

[11] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, and Mariann Hardey, editors, *MSM*, volume 718 of *CEUR Workshop Proceedings*, pages 93–98. CEUR-WS.org, 2011.

[12] Parul Pandey. Simplifying sentiment analysis using vader in python (on social media text), Nov 2019.

[13] Chris Tam. Implementing linear and polynomial regression from scratch, Apr 2020.

[14] Mikalai Tsytsarau and Themis Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514, 2012.