CSE3001: Software Engineering Sentiment Analysis Tool

PROJECT REPORT

GROUP 12
B.ADARSH REDDY (18BCI0196)
SUMEET ROY KURIAN (18BCI0188)
RIYA SHRESTHA (17BCE2346)
TARUN CHAND (17BCE2365)

SUBMITTED TO: PROF. AKILA VICTOR

ABSTRACT

Sentiment Analysis is a text classification method that analyses an incoming message/review and tells whether the underlying sentiment is positive, negative our neutral. Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organizations across the world. It can also be an essential part of your market research and customer service approach. Not only can you see what people think of your own products or services, you can see what they think about your competitors too. The overall customer experience of your users can be revealed quickly with sentiment analysis, but it can get far more granular too. In this project we build a model in python to analyze product reviews and predict whether it is a positive or negative review. With the recent advances in deep learning, the ability of algorithms to analyze text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool for doing in-depth research. We use logistical regression, word vectorization and data pre-processing to build a ml model to categorize reviews as positive or negative.

The model is trained over Andrew Maas' IMDb Reviews dataset. First the reviews are cleaned and preprocessed after which we vectorize the reviews. Then a logistical regression model is trained over the data to make the classifier.

AIM

The purpose of the project is to help people provide a suggestion by considering and analyzing a vast amount of data. Because of taking a big data base into consideration the reviews will be

accurate and trustable. In the recent years' people have problem in choosing the right products and services. there has been a lot of progression in machine learning and deep learning platforms, the ability of Artificial intelligence to analyze text has improved considerably. Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of different brand, products or services while monitoring online conversations. With the recent advances in deep learning, the ability of algorithms to analyze text has improved considerably. Use of advanced artificial intelligence , mathematical and machine learning concepts to make an effective tool for doing in-depth research for brand/product management and planning.

SCOPE OF THE PROJECT

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. Social media monitoring tools like Brand-watch Analytics make that process quicker and easier than ever before, thanks to real-time monitoring capabilities. The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organizations across the world.Shifts in sentiment on social media have been shown to correlate with shifts in the stock market. The Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages ahead of 2012 presidential election. Being able to quickly see the sentiment behind everything from forum posts to news articles means being better able to strategize and plan for the future. It can also be an essential part of your market research and customer service approach. Not only can you see what people think of your own products or services, you can see what they think about your competitors too. The overall customer experience of your users can be revealed quickly with sentiment analysis, but it can get far more granular too. The ability to quickly understand consumer attitudes and react accordingly is something that Expedia Canada took advantage of when they noticed that there was a steady increase in negative feedback to the music used in one of their television adverts. Companies are increasingly using sentiment analysis technology to monitor internal communications in order to better understand employees' moods and assess any potential risks. "Sentiment analysis has become a form of risk management and is emerging as a useful risk control tool for a variety of businesses," said Vasant Dhar, a data scientist and professor at New York University's Stern School of Business and the Center for Data Science. Firms in highly regulated, compliance-oriented or risk-focused industries, such as financial services, health care and insurance, are starting to use the technology to identify and address regulatory risk issues, compliance problems and potential

PROPOSED METHODOLOGY

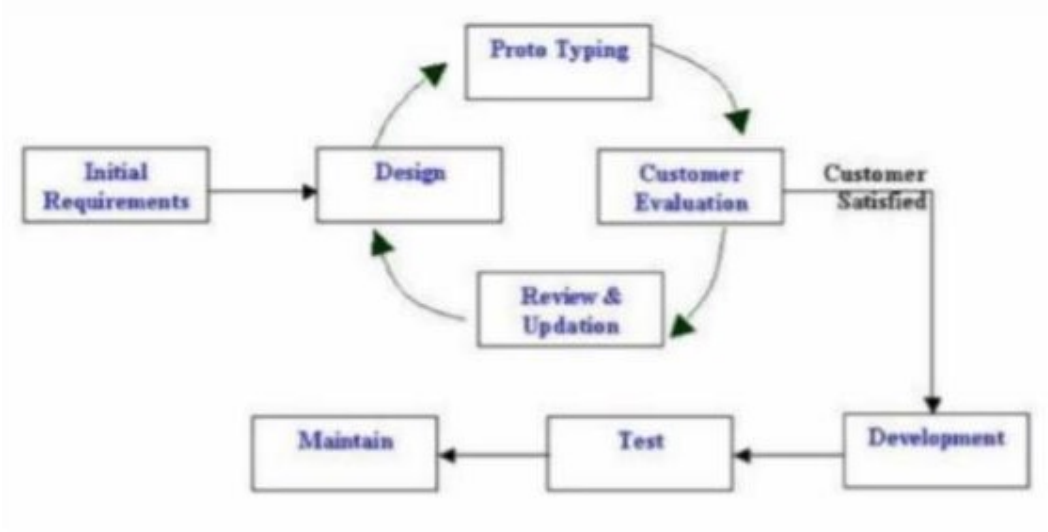
Training classifier model: We use Andrew Maas' Large Movie Review Dataset v1.0 to train the model. The dataset 50,000 movie reviews which are marked as positive or negative. • The

reviews are cleaned ,this includes removing breaks, symbols and stop words(words like if', 'but', 'we', 'he', 'she' etc which do not influence the final sentiment of the review much).After which reviews are pre-processed by applying normalization in the form of word stemming. • Now we vectorize the dataset, create a large matrix with one column for every unique word in the dataset. Then we transform each review into one row containing 0s and 1s, where 1 means that the word in the dataset corresponding to that column appears in that review. We use n grams vectorization where the dataset also combines 2,3 words together

to make additional features (this is to make the model more accurate. For example, if a review had the three word sequence "didn't love movie" if these words are considered individually with a unigram-only model the model will not capture that this is actually a negative sentiment because the word 'love' by itself is will be highly correlated with a positive sentiment.) • split data into test set and train set. • train logistical regression model on the train set and then test its accuracy on the train set • After the model is trained it can be used to analyze the sentiment behind reviews. Analyzing sentiment of a review set: 1)Get review set and vectorize reviews 2)Calculate a sentiment score for each vectorized review using the trained classifier. 3)Calculate a mean score for the review set and use it to give a overall sentiment (positive, negative, neutral)

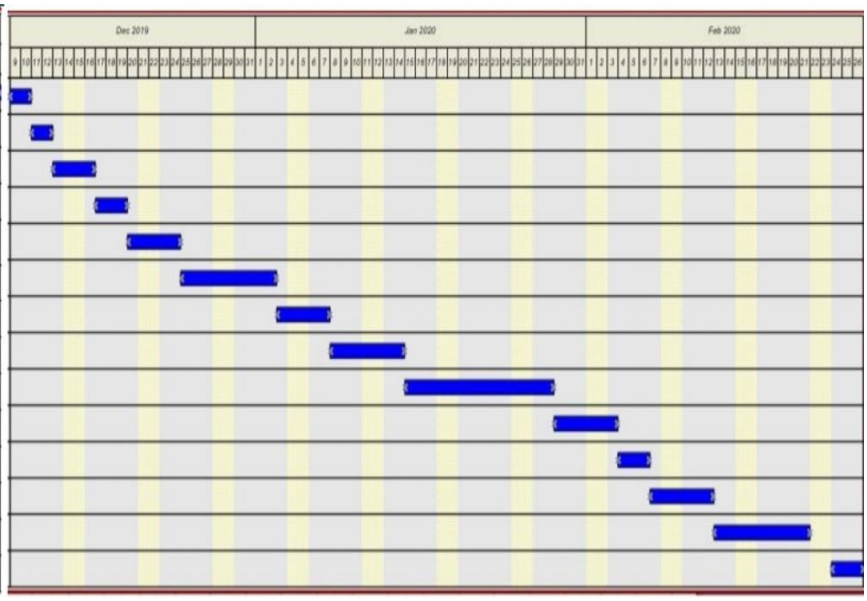
PROCESS MODEL

We will be using evolutionary prototyping for this project in which the software development method is used where the developer or development team first constructs a prototype. After receiving initial feedback from the customer, subsequent prototypes are produced, each with additional functionality or improvements, until the final product emerges. Any analytical or artificial intelligence based software's need constant changes to support the size and the type of data we use. So, evolutionary prototyping will be most suitable for creating the sentimental analysis tool.

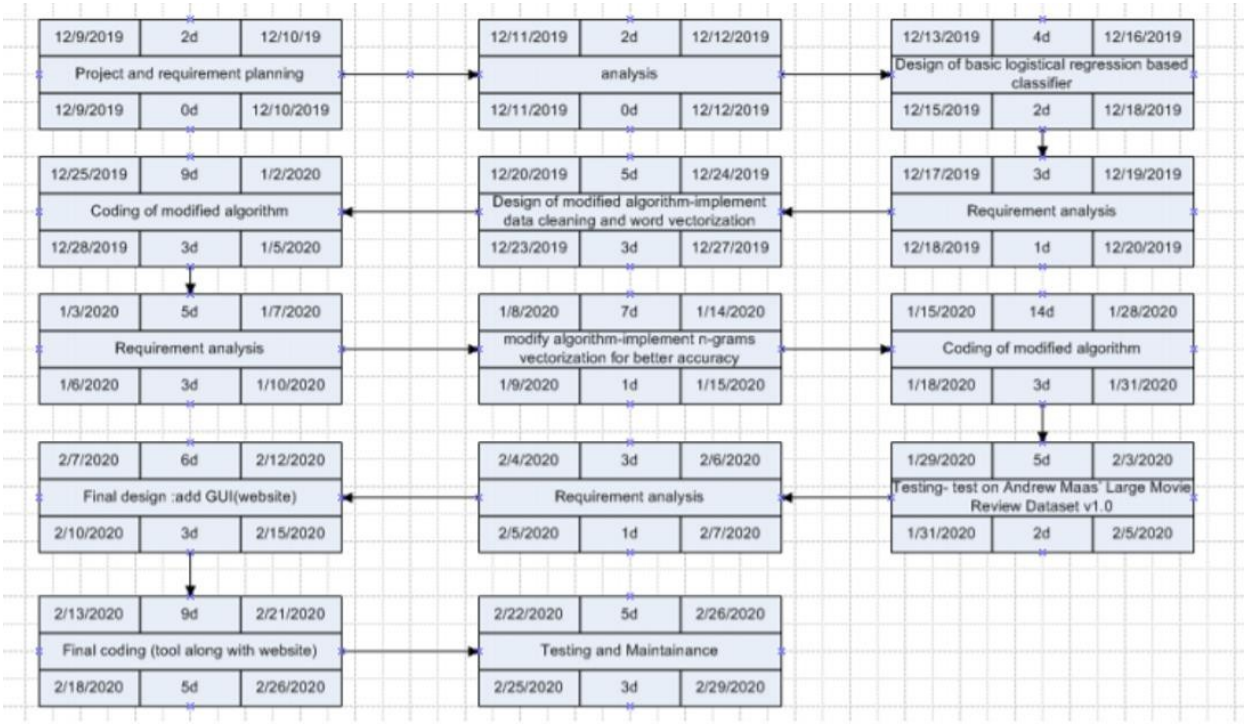


SCHEDULING (GANTT CHART)

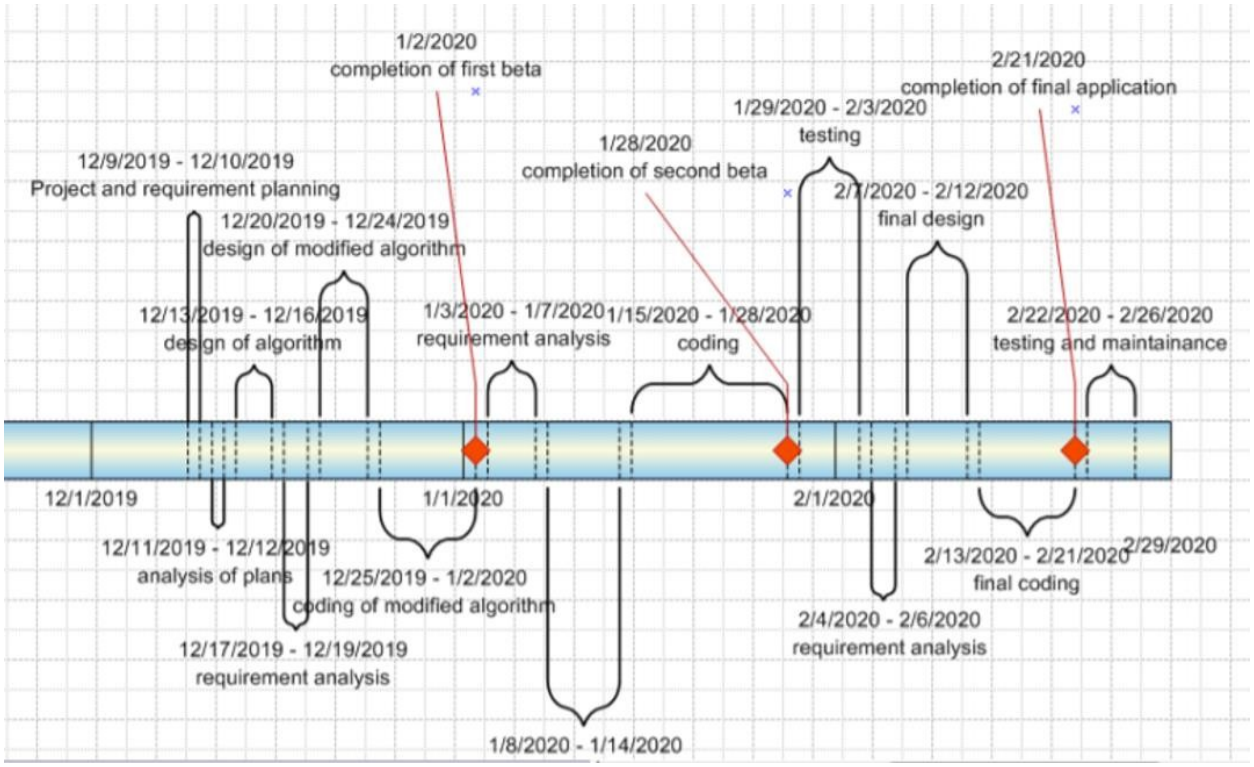
ID	Task Name	Start	Finish	Duration
1	Project and requirement planning	12/9/2019	12/10/2019	2d
2	Analysis	12/11/2019	12/12/2019	2d
3	Design of basic logistical regression based classifier	12/13/2019	12/16/2019	2d
4	Requirement analysis	12/17/2019	12/19/2019	3d
5	Design of modified algorithm-implement data cleaning and word vectorization	12/20/2019	12/24/2019	3d
6	Coding of modified algorithm	12/25/2019	1/2/2020	7d
7	Requirement analysis	1/3/2020	1/7/2020	3d
8	modify algorithm-implement n-grams vectorization for better accuracy	1/8/2020	1/14/2020	5d
9	Coding of modified algorithm	1/15/2020	1/28/2020	10d
10	Testing- test on Andrew Maas' Large Movie Review Dataset v1.0	1/29/2020	2/3/2020	4d
11	Requirement analysis	2/4/2020	2/6/2020	3d
12	Final design :add GUI(website)	2/7/2020	2/12/2020	4d
13	Final coding (tool along with website)	2/13/2020	2/21/2020	7d
14	Testing and maintainance	2/24/2020	2/26/2020	3d



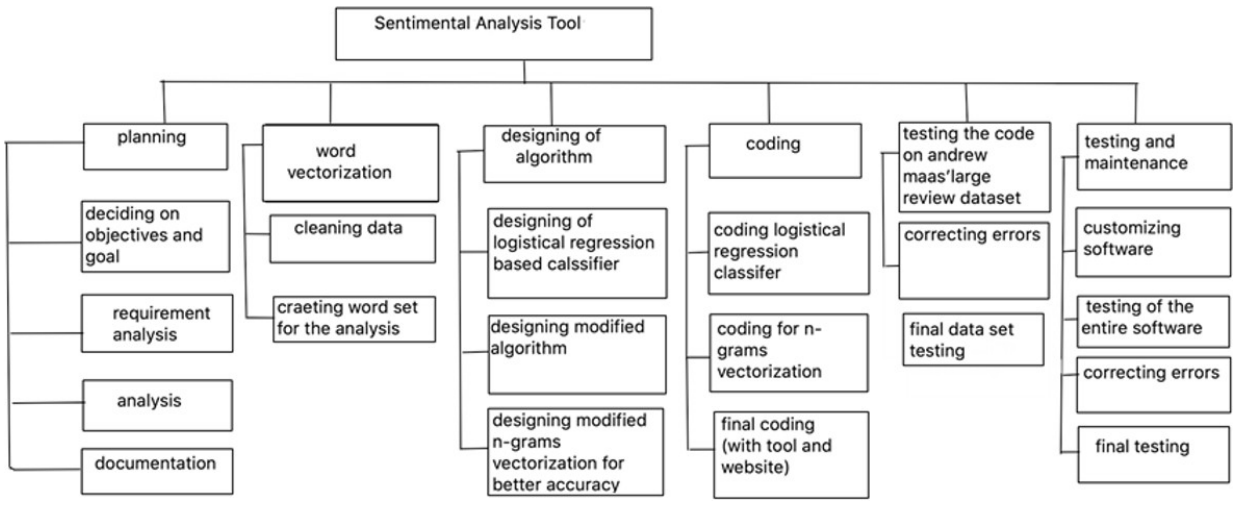
SCHEDULING (PERT CHART)



SCHEDULING (TIMELINE CHART)



WORK BREAKDOWN STRUCTURE



SOFTWARE REQUIREMENT SPECIFICATION

FUNCTIONAL REQUIREMENTS

- Dataset upload
- This functionality helps the user provide the review set to be analyzed
- Any size of dataset should be supported
- Sentiment Analysis
- Calculate sentiment for each review
- Based on review set give a overall sentiment (positive, negative)
- Login and authentication

NON-FUNCTIONAL REQUIREMENTS

- Performance
  - Tool should be able to analyze 10,000 reviews in under 3 min.
  - Memory used should be below 512 MB.
- Privacy
  - Secured user data
- Consistency
  - Tool should provide consistent results
- Usability
  - The a tool should be easy to use
  - Responsive UI

USER INTERFACE REQUIREMENTS

The user interface is going to have the following elements:

• A selection menu to select review dataset to upload.

• A space to display the reviews.

• A button to reload and recalculate the recommendations.

• A button to clear all the selections

STAKEHOLDERS

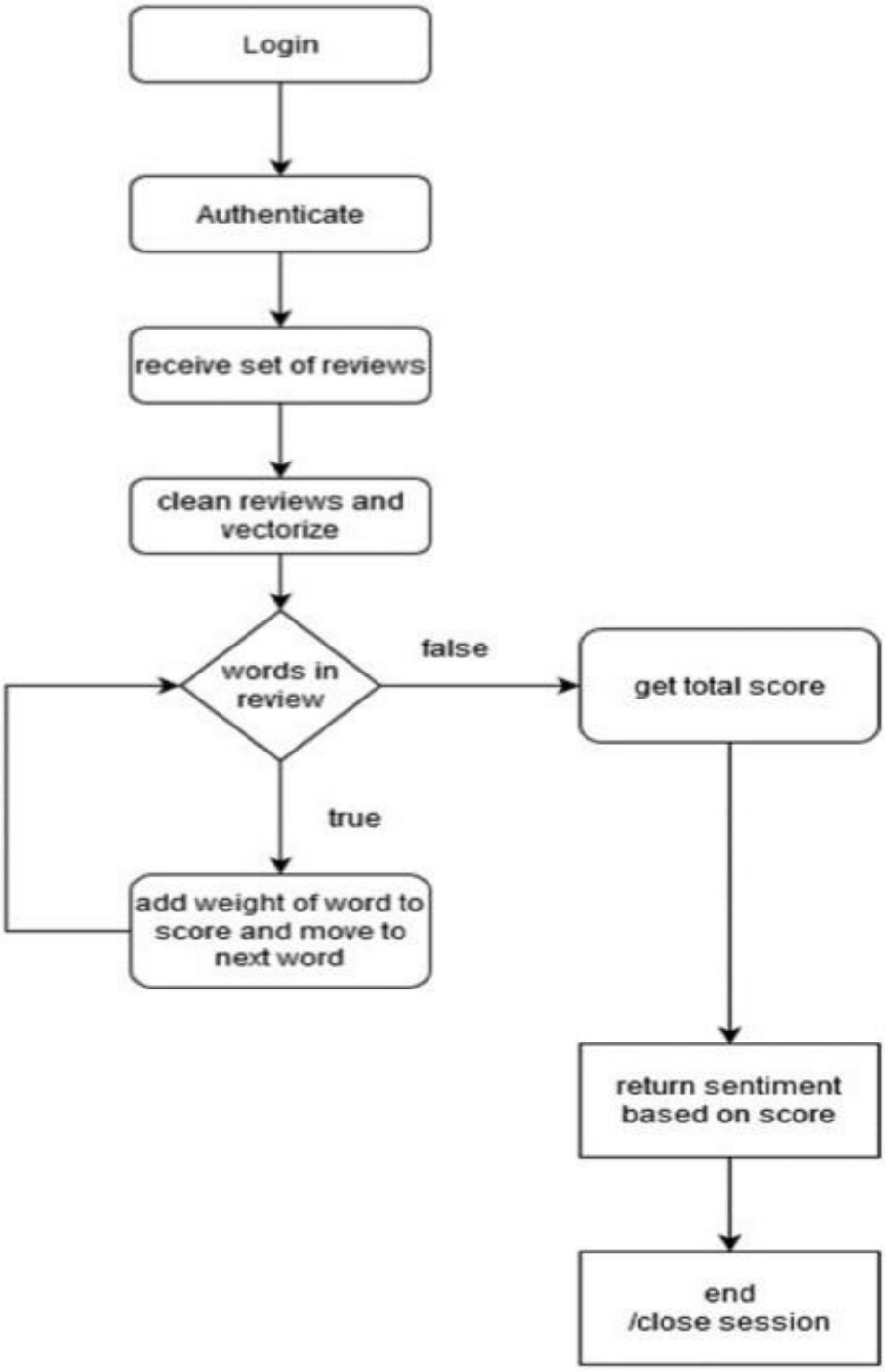
- Companies/brands who want analysis of their reviews
- User of products/services who provide reviews
- Advertisement agencies
- Clients sales and marketing department
- Clients Customer Care department

POSSIBLE RISKS

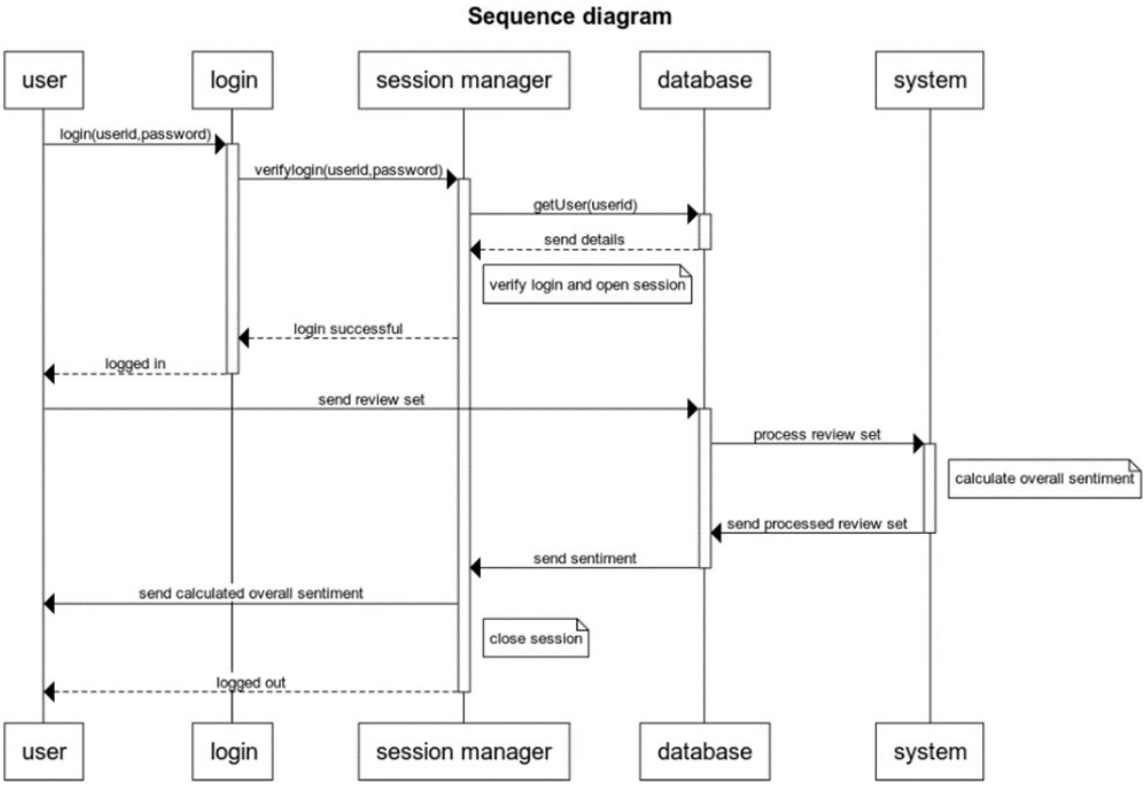
- Sentiment analysis tools can identify and analyze many pieces of text automatically and quickly. But computer programs have problems recognizing things like sarcasm and irony, negations, jokes, and exaggerations - the sorts of things a person would have little trouble identifying. And failing to recognize these can skew the results.(risk type: product)
- Unavailability of team member(risk type: project)
- Reducing time constraints (risk type: project)
- Change in requirement analysis(risk type: project and product)
- With short sentences and pieces of text, for example like those you find on Twitter especially, and sometimes on Facebook, there might not be enough context for a reliable sentiment analysis. However, in general, Twitter has a reputation for being a good source of information for sentiment analysis, and with the new increased word count for tweets it's likely it will become even more useful.(risk type: product)
- Possible attack on system with engineered reviews (adding words which the system may tag as very positive or very negative in order to get a desired sentiment)(risk type: product)
- Tool can be pirated easily due to small size and offline availability(risk type :business)

UML DIAGRAMS

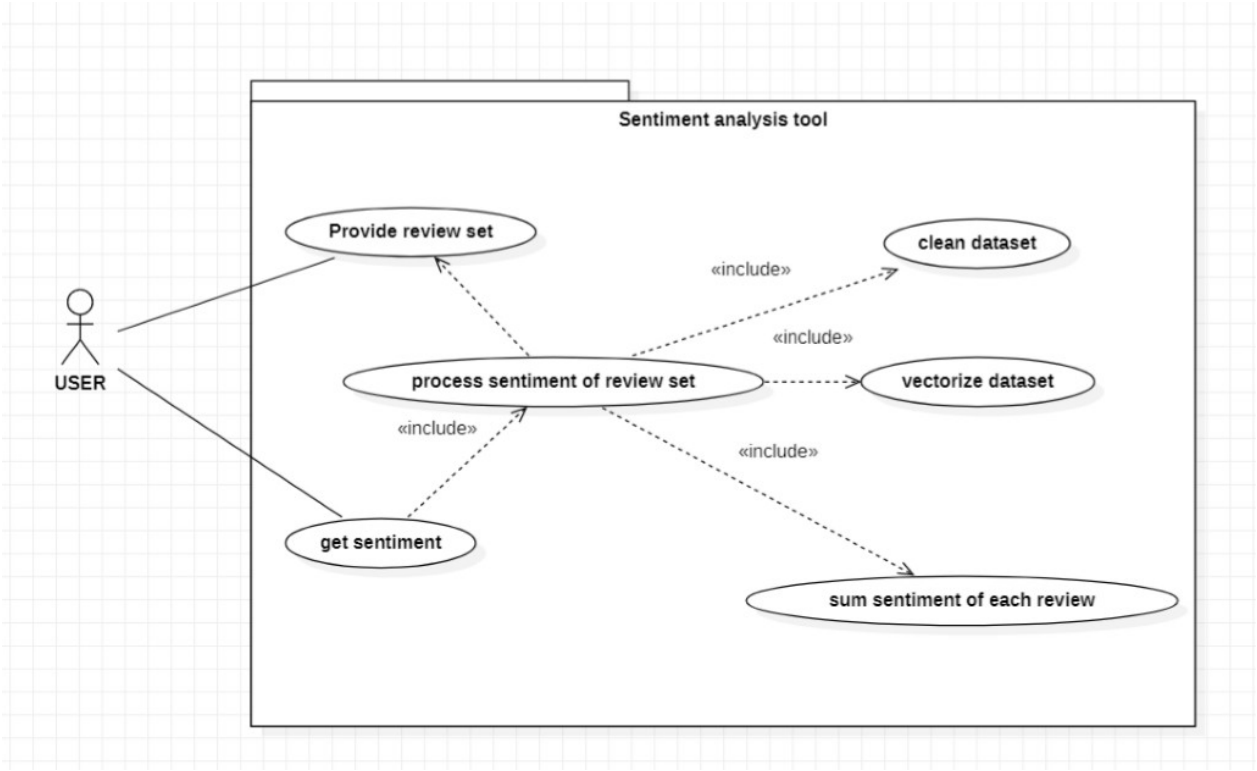
Activity diagram



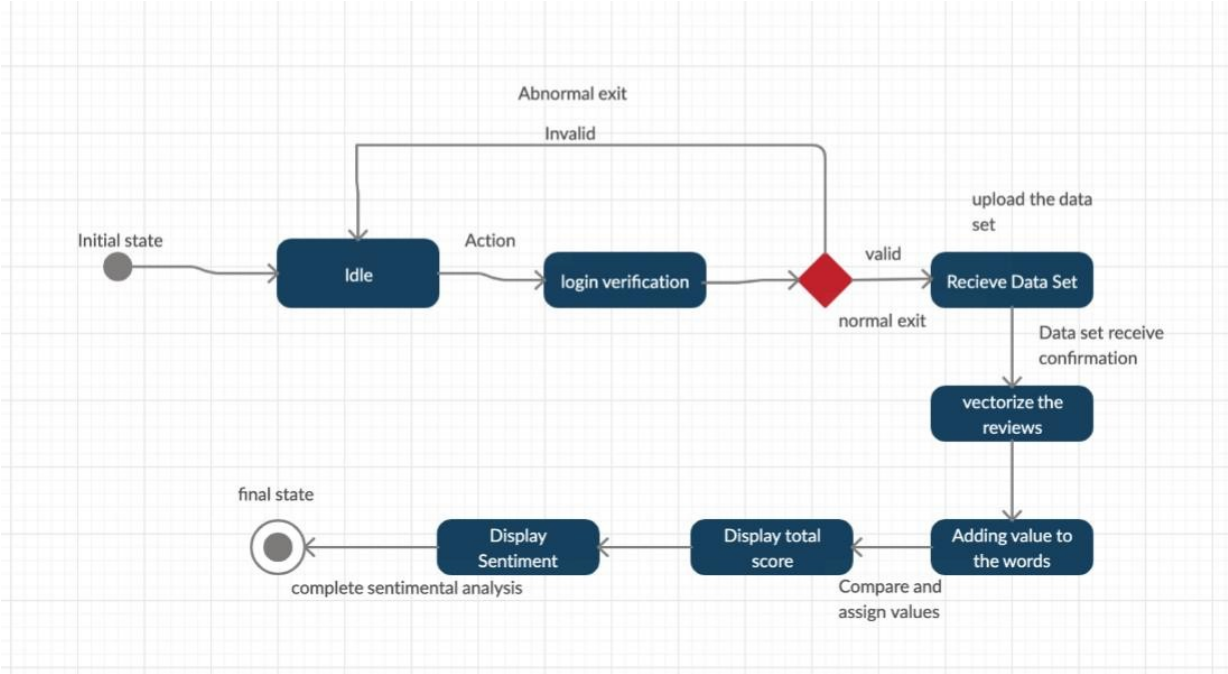
Sequence Diagram



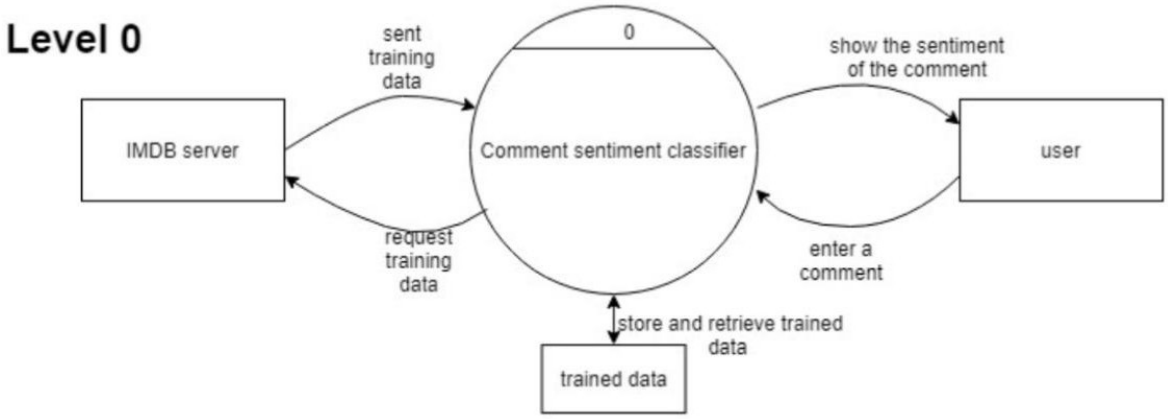
Use case diagram

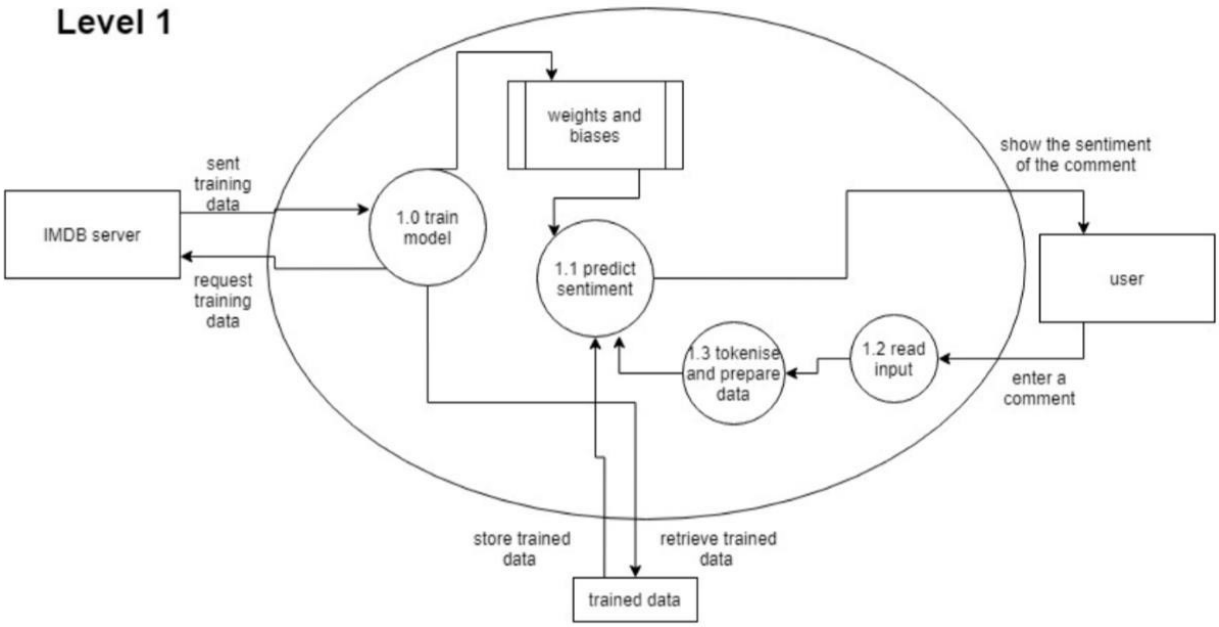


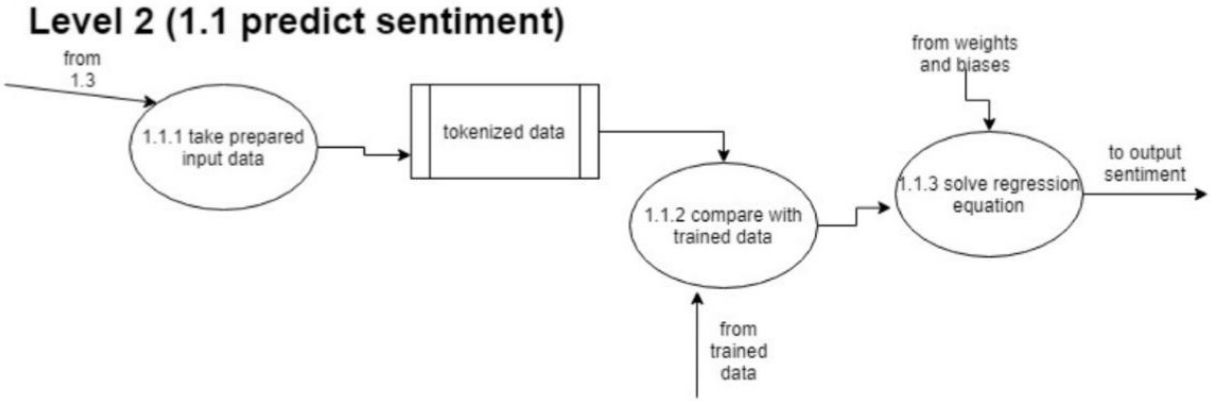
STATE CHART DIAGRAM



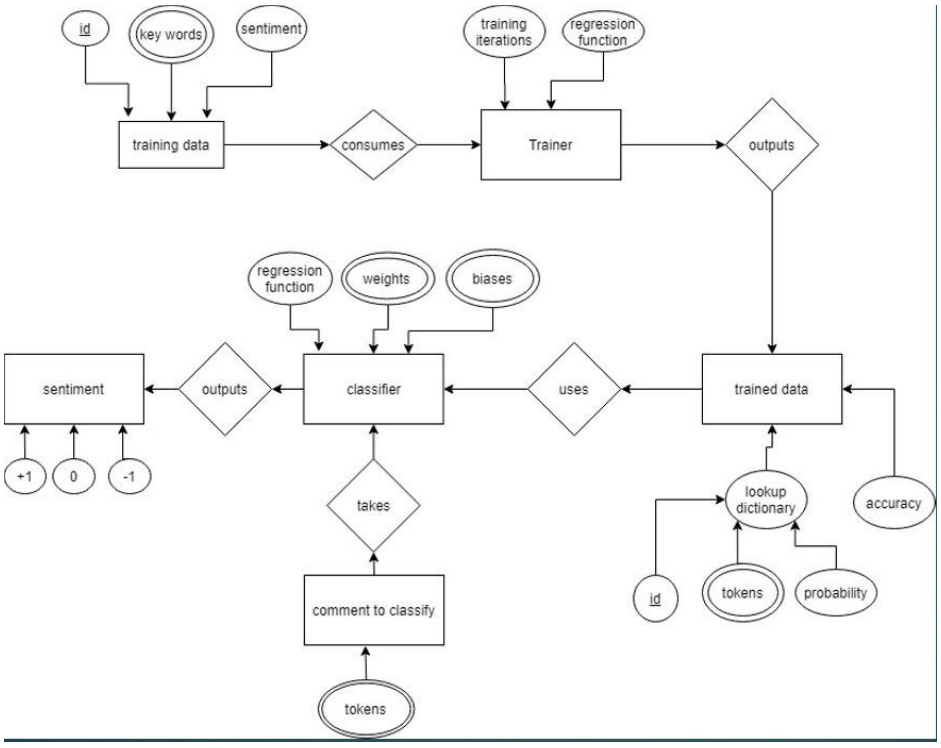
DATA FLOW DIAGRAM



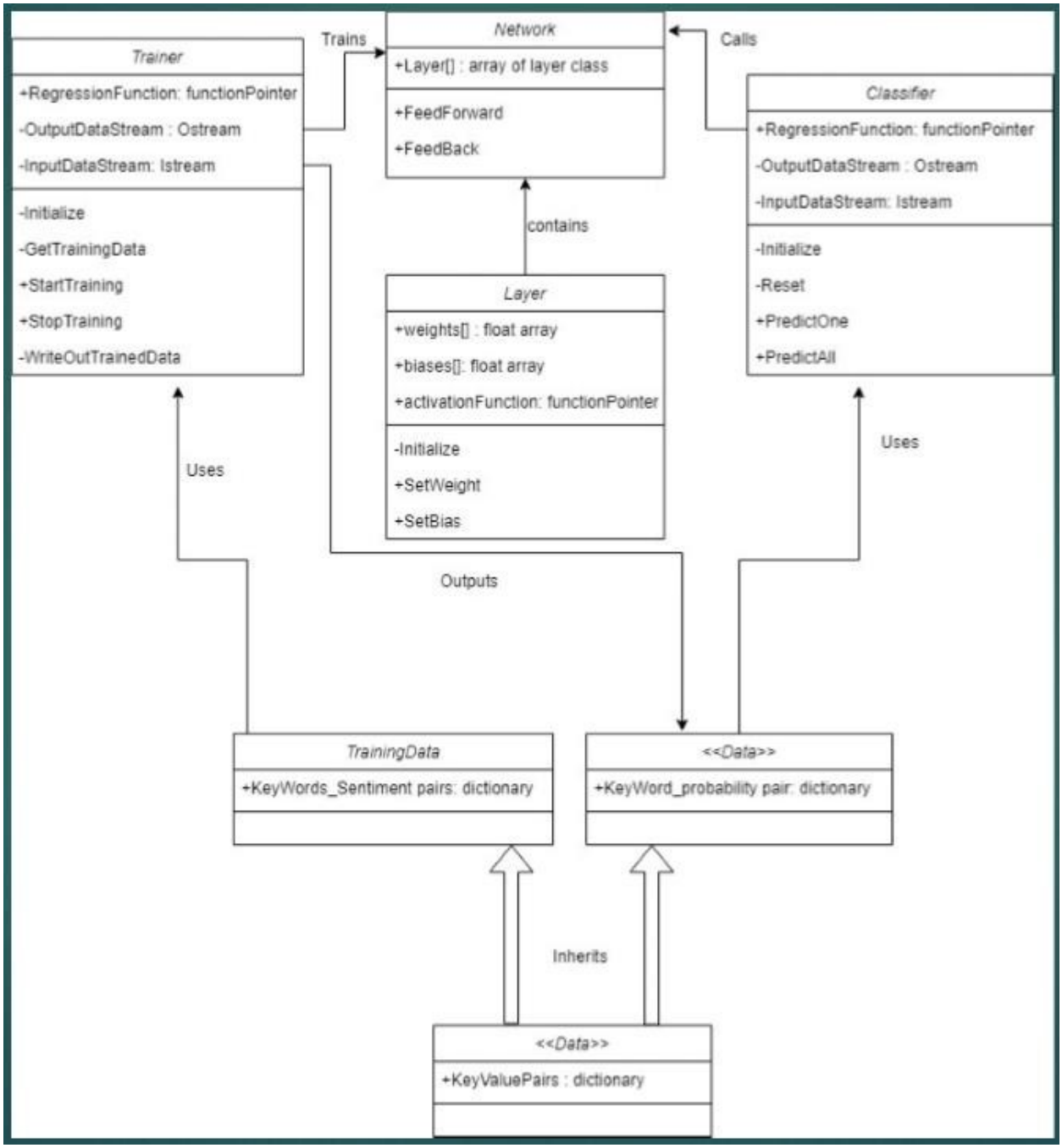




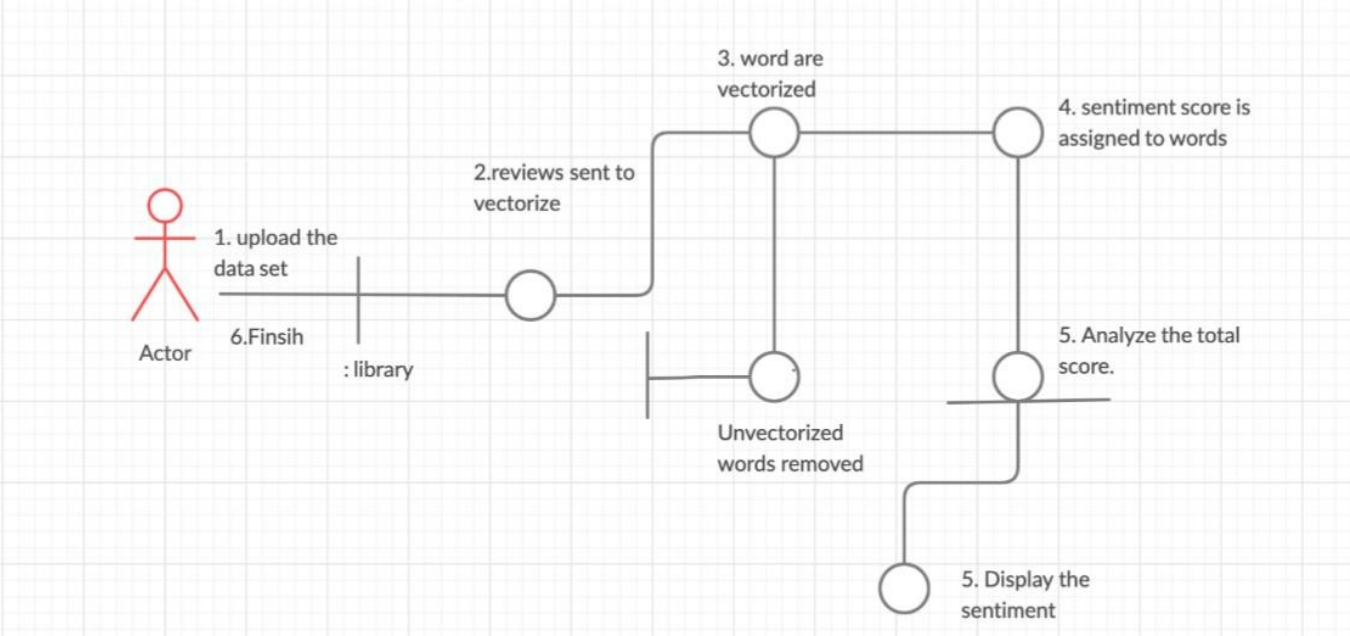
ENTITY REALTIONSHIP DIAGRAM



CLASS DIAGRAM



COLLABORATION DIAGRAM

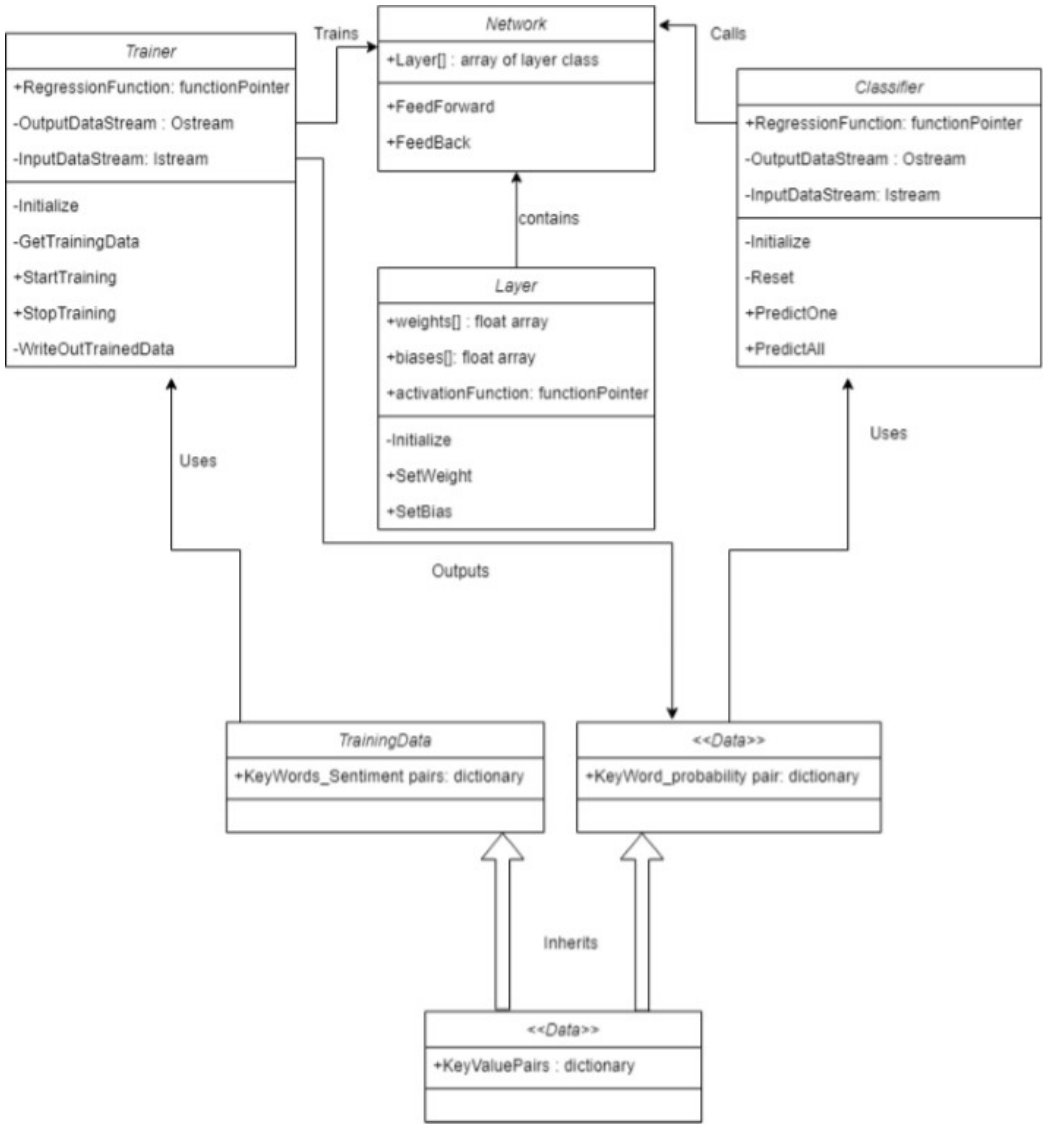


System architecture description

Overview of modules / components

The structure of our project can be summarized by the following class diagram. It should be noted that the network class, the trainer class and the classifier classes are implemented by the scikit-learn module in the python library and the training data and trained data structs are internally calculated and stored inside the said module.

This whole structure is packaged up into its separate module and used inside a callback function to facilitate the process of making a GUI for the end user. The said GUI is implemented using the tkinter module of the python library. The .NET (WPF) solution, although it plays well with the windows environment, was scratched in favor of tkinter due to interoperability reasons as it provides a streamlined workflow between the backend and the GUI.



Structure and relationships

As we can see in the diagram, the abstract struct- data, is inherited to form the training and trained data which are then used by the trainer and classifier respectively. This struct is internal to the scikit-learn module of the python library.

Apart from this, we have the trainer and classifier classes, which use the LinearSVM regression model and are also implemented and nicely packaged inside the scikit-learn module.

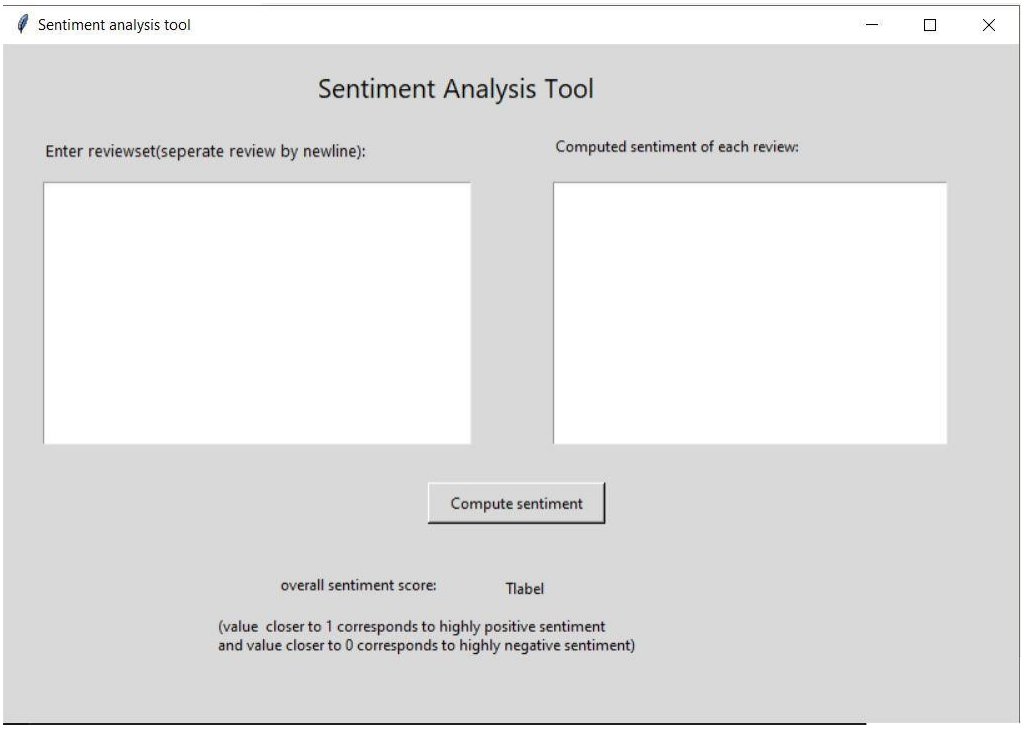
One module not part of the class diagram but still required for the operation is a class that handles Regular Expressions for the preprocessing of the input data.

This entire functionality is packaged up into one function called calculate_sentiment and is used as a callback function for a button present on the GUI.

User interface issues

The main issue faced during making of the user interface was the lack of time and interoperability between python and .net code. The following image shows the user interface designed. It consists of the input data on the left and the output sentiment on the right list box. In the far bottom, we have an overall sentiment score of the reviews combined. In the bottom, we have one button for calculating the sentiment.

Due to time constraints and issues with interoperability, we used tkinter module from the python library itself to draw the GUI using python. This makes our application kind of cross platform and also plays nice with the backend system that uses scikit-learn. The original design was preserved and the new design looks exactly like the one originally intended.



Detailed description of components

Component template description

In the following sections, we are going to go over each component in order. This includes:

- Implementation of the training
- Implementation of the actual prediction
- Preprocessing of the inputs
- Postprocessing of the outputs to be displayed
- The GUI component

Implementation of training:

To train the system, we got the data set from the IMDB api which consists of various datasets of user reviews matched with comments that make it ideal dataset to use for sentiment analysis training.

We use the CountVectorizer class from the scikit-learn module to train the system. We preprocess the data and call the corresponding functions including fit to train the model.

After the model is trained, the result is stored inside the module to be used during prediction stage.

Implementation of prediction:

Since the model is already trained and all the weights and biases are determined, we can send it the preprocessed data to be predicted and obtain the results. For this, we use the LinearSVC class from the scikit- learn module to call the function predict on it to get the results of the predictions.

We then use the result to draw elements on the GUI and display the overall sentiment score.

Preprocessing and Postprocessing of input/output texts:

For the transformations and cleaning of the text that either has to go as an input or gathered as an output, we have used **Regular expressions** to intelligently weed out the stop words, punctuations, symbols and any unwanted characters that may either provide diminishing returns or render our system useless in terms of calculating and displaying the data.

GUI component:

As mentioned a few times before, we are using the tkiner module to draw buttons and textboxes in the window. We have assigned a callback function for when the calculate button is clicked which calls the appropriate function to take the text from the input box and use it to calculate the sentiment score and display on the bottom label placed on the window.

Design decisions and tradeoffs

Due to time constraints and issues with interoperability, we had to ditch the idea of making a GUI using the.net framework and connecting the two components using a REST API. This would have supported the possibility of having many different clients for our backend and also would have a solid desktop application custom built for windows with performance in mind. However, since face-to-face communication was not possible and shifting of the deadlines, the initial plan was to be abandoned for a cross-platform GUI using python itself.

Pseudocode for components

The following section shows the extracts of the code for the crucial parts of our application.

Training:

```
ngram_vectorizer=CountVectorizer(binary=True,ngram_range=(1,3),stop_words=stop_words)

ngram_vectorizer.fit(reviews_train_clean)

X =

ngram_vectorizer.transform(reviews_train_

clean) X_test =

ngram_vectorizer.transform(reviews_test_c

lean)
```

Predicting:

```
#initialization

classifier =

LinearSVC(C=0.01)

classifier.fit(X, target)


#inside predict function:

def compute_sentiment(inputText):

    review_set_clean = preprocess_reviews(inputText)

    reviewset_vectorized =

    ngram_vectorizer.transform(review_set_clean)

    result=classifier.predict(reviewset_vectorized)
```

Pre/Post Processing:

```
reviews = [REPLACE_NO_SPACE.sub("", line.lower())

for line in inputText] reviews =

[REPLACE_WITH_SPACE.sub(" ", line) for line in
```

inputText]

GUI:

```
def vp_start_gui:
    global val, w, root
    root = tk.Tk()
    top = Toplevel1 (root)
        sentiment_analysis_tool_support.init(root, top)
    root.mainloop()
```

Callback for predict function:

```
self.Button1 = tk.Button(top)
self.Button1.place(relx=0.418, rely=0.646, height=33,width=142)
#some styling options, omitted for the sake of brevity
self.Button1.configure(text="'Compute sentiment"')
self.Button1["command"]=self.predictSentiment
```

CONCLUSION

Along with the rise of social media platforms, there has been a rise in the amount of information, both true and false, spread.. However, there hasn't been much transparency in terms of information analysis till no. so we created a sentimental analysis tool to help us make decisions with the help of information we have.Sentiment analysis gives you a clear overview of customer satisfaction, agent by agent. This means you can keep an eye on the quality of service each team member is offering customers, as well as their more subtle ability to create happy customers. Emotional triggers drive our decisions. This tool can identify the emotive triggers that change customer mood and Understanding what messages trigger certain emotions in your customers can help you give better service, and is also useful for creating effective marketing materials and reduce annoyance.

LINK FOR THE DEMO AND CODE FILES:

https://drive.google.com/drive/folders/1ryrYqGR2O9zQ0L6xcz60HyZM0TsZUZCv?usp=sharing

References

- Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, *2*(1-2), 1-135.
- Tian, X., Saito, H., Preis, S. V., Garcia, E. N., Kozhukhov, S. S., Masten, M., ... & Panchenko, N. (2013, May). Practical simd vectorization techniques for intel® xeon phi coprocessors. In *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*(pp. 1149-1158). IEEE.
- Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, *9*(3), 293-300.
- Drucker, H., Burges, C. J., Kaufman, L., Smola, A. J., & Vapnik, V. (1997). Support vector regression machines. In *Advances in neural information processing systems* (pp. 155-161).
- Lundh, F. (1999). An introduction to tkinter. *URL: www. pythonware. com/library/tkinter/introduction/index. htm*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.