

Detection of Emotion of Students using Text Mining and Machine Learning

Final project report submitted for B.Tech Project

B.Tech

by

Nishant Kumar (2016IPG-063)
Prakash Gatiyala (2016IPG-069)
Tulasi Kumar Gupta (2016IPG-112)



ABV-INDIAN INSTITUTE OF
INFORMATION TECHNOLOGY AND
MANAGEMENT
GWALIOR-474010

2019

CANDIDATES DECLARATION

We can say that the work is done in this report is by us. The title of the report is **Detection of Emotions of Students using Text Mining and Machine Learning**. This is report is submitted as the Final project report submitted for B.Tech. This project is submitted for B.tech degree **Bachelor of Technology** and submitted to institute for a proof that We work during summer 2019 between *May 2019* to *September 2019* under the guidance of **Dr. Ajay Kumar** and **Dr. Vinay Singh**.

In this report, The text and figure that we have taken from the internet are cited with reference.

Date:

Student Signatures

The work done by students is correct and it is checked by me with my best knowledge.

Date:

Signatures of the Supervisors

ABSTRACT

In today's world technologies are advancing at a fast rate. Interaction between human and computer are growing rapidly. Therefore human and computer interaction becomes a very popular field of research. There are many aspects of human behavior in which emotions can play an important role in human-computer interaction. So the computer interface should be able to recognize the emotion of the users to exhibit truly intelligent behavior. People express their emotions in the form of speech, facial expressions, and text. Every day we get a massive amount of data from different social sites, blogs, etc. So performing any operation on these data is very complex because they contain both plain-texts as well as short messaging language. In this paper, we are focusing on emotion detection from text and describing different methods to do this job.

ACKNOWLEDGEMENTS

We are worked under **Dr. Ajay Kumar** and **Dr. Vinay Singh**. They give us full autonomy to work on this project. Because of that we can easily implement our idea in this project and obtain the desired result. We experimented with multiple ideas in this project with the proper guidance of our supervisor. Our Supervisor support us in this project throughout the entire summer of 2019. We are thanking them for their this great help in this project. Without proper guidance from our supervisor, this will not be possible. Our supervisor is very motivating and helpful. They find our mistake in this project and correct us whenever there is a mistake and guide us that how to avoid mistakes.

In the end, We want to thank our supervisor for their help.

(Nishant Kumar)

(Prakash Gatiyala)

(Tulasi Kumar Gupta)

Contents

ABSTRACT	2
1 INTRODUCTION AND LITERATURE SURVEY:	7
1.1 INTRODUCTION	7
1.2 BACKGROUND	8
1.3 LITERATURE REVIEW	8
1.4 OBJECTIVE	10
2 DESIGN DETAILS AND IMPLEMENTATION	11
2.1 Overview of Methodology	11
2.1.1 Chats dataset	11
2.1.2 Used Python Library	12
2.1.3 Preprocessing of the dataset	16
2.1.4 Feature extraction of the datasets	17
2.1.5 Feature selection of the datasets	17
2.1.6 Training	17
2.1.7 Classification	18
2.1.8 Flowchart of methodology	19
2.1.9 chat application(frontend)	20
3 RESULTS AND DISCUSSION	22
3.1 RESULTS	22
3.2 DISCUSSION	22
4 CONCLUSION	24

List of Figures

2.1	dataset.	12
2.2	SVM figure 1	18
2.3	SVM figure 2	19
2.4	Graphical representation of used methodology.	19
2.5	working of chat module	20
2.6	Working of Complete Application	21
3.1	Text Input.	23
3.2	Output.	23

ABBREVIATIONS

SVM	support-vector machine
NLTK	Natural Language Toolkit
NLP	Natural language processing
ML	machine learning
TF	term frequency
TF-IDF	term frequency–inverse document frequency
IDF	Inverse document frequency
UI	user interface
RIA	rich internet application
HTTP	HyperText Transfer Protocol
POS	parts of speech
TV	television

Chapter 1

INTRODUCTION AND LITERATURE SURVEY:

This chapter includes the details of Sentiment Analysis, Machine Learning techniques and literature review related to work done in this field.

1.1 INTRODUCTION

Emotions can be seen, where people interact and communicate with each other. The exchange of [Por+16] emotions can be seen through messages, the post of articles and blogs, facial expression and person's speech and it is known as text, facial and speech-based emotion respectively. Recently, the work on facial and speech emotion detection has been done enough but attraction on text-based emotion detection system is needed. Emotions can be represented by a single word or a group of words. Word level emotions can be detected easily but on the sentence and document level emotion detection is not easy to find their emotions. Sentences have the main information of the documents, therefore on the document level emotion detection depends on the individual sentences on the document. Globally, the emotions are expressed [HCR93] in happy, sad, angry, love, fear, surprise, disgust, etc.

The motivation behind this work has come from the recent growing interest in emotion detection from the text because the world wide web has facilitated due to online communication system, written content and blog post, and these open a new avenue to detect the emotions on the text data. By using Text Mining approach, the idea of this project is to determine the user's mood according to incoming messages in chat application from that particular user.

1.2 BACKGROUND

Text mining is a [YP07] technique to extract high-quality information from text. Therefore, making the information accommodate in the text usable to the different algorithms. Information can excerpt to determine the outline contained in the script. Hence, you can inspect words, clusters of words used in the script. In familiar terms, text mining will “turn text into numbers”. Such as guessing data mining projects, the application of unsupervised learning methods. After extracting high-quality information using text mining, now researchers are doing emotion analysis on the information to know people’s behavior, mood, and emotions. Emotion analysis is a process of determining the polarity of the text. Nowadays emotion analysis became a very useful technique to know the behavior of the crowd during elections. Most of the party uses this technique to know what people are thinking about their party, their leaders and their policies. They use tweets, posts from the social media app, and then perform sentiment analysis on the data.[Mat+14] Ecommerce companies like(Zomato, Flipkart, Amazon) uses emotion analysis to know the opinion of products. They get their feedback and then sentiment analysis on their reviews and try to figure out what customers exactly wants. Companies can know the likes and dislikes of the customers.they also develop a recommendation system after analyzing their reviews. It is also socially beneficial because of most of the students living away from their parents. They may be in a good situation, but there may be a possibility that he is facing some issues, but not having much courage to tell their parents. In this situation, text mining and emotion analysis are very useful. Parents can understand according to their messages.

1.3 LITERATURE REVIEW

In Natural Language [LBH15] Processing and Sentiment Analysis field, plenty of research has been done with different methods. We are going to look into the past research in this field for this project. In [YD17] this project, We come up with a little bit different approach from the typical techniques. Examining [Kum+08] a big dataset can give us a large number of convenient details about sentiment analysis. But in our project, we use a small dataset and try to find the exact details about it. In our project, We are executing real-time analysis of text data of chat. We are given input from user chat to our machine learning model. That model will analyze it and produce the result with emoji. Moreover, our dataset is not huge for processing it with the machine learning model. We will easily analyze it with real-time and produce a result to the user during the chatting. It will assist them to understand each other responses quickly, and by this relationship between them can get better.

In this project [YD17] we examine each user emotion analysis and find the different categories about people’s mindset and we can classify them in different sections according to their characteristics. We use data mining and NLP techniques to analyze the sentiment analysis of the user chat in real-time.

In previous research, We have been done in the small dataset of Twitter users. They use the [KS12] Lexicon method for examining the sentiment analysis of user tweets on twitter. The dataset they picked for analysis is from a different category of people and is more popular tweets, i.e. the dataset is very dynamic. After that, they find the different outcomes from the large dataset of the twitter dataset.

For analyzing the large dataset of user tweets on twitter, We need human assistance that can help to find the details of the text very quickly and give results on the spot. The result of small data of tweets can be differentiated from predefined [sen] sentiment 140 datasets. And the research of the above method that we discussed gives the consistent result and are helpful for small dataset. The process in this [YD17] method is quite similar to our project.

In the past research, the [YC17] researchers have examined emotion detection with different machine learning models or we can say that they used the variety of machine learning models the get the result for emotion detection. The most common methods of machine learning used are Naive Bayes and the [KP07] support vector machine and another two methods are artificial neural networks and maximum entropy. There are three [Med14] different methods for analyzing the emotion detection. The name of these machine learning methods are dictionary-based, The dictionary-based method can describe that in this method we examine each word by word of our dataset to word available in the dictionary. After that, we can get the result of the text dataset i.e. we can say we can find the different emotions out of the text. Another method for emotion detection is Hybrid and machine Learning.

However, the [She12] machine learning methods for emotion detection is most accurate than dictionary-based methods. The accuracy of dictionary-based methods is not up to the mark that can produce the correct emotions out of the text dataset. The machine learning model outperforms in emotion detection is and can provide good accuracy of text for emotions. We can accurately predict the right emotions that can help to analyze the dataset and take decisions according to the emotions and provide the user the most efficient approach to make better decisions among them about their various works. The accuracy of machine learning methods is very huge because of this researcher has produced the research in this field in the [YC17] past for analyzing the sentiment. In the [MM11] past research on sentiment analysis on text data has been done on only the English language. So there is a good opportunity to analyze the sentiment of the dataset in different languages of the text dataset. So we can wind-up that emotion detection is a very challenging field and there are more issues in this field for analyzing the sentiment out of text dataset. So we can say there is a huge opportunity for anyone to research in this field and get a significant result by research and that will be helpful to people and it is a hot area to work on.

1.4 OBJECTIVE

The major objective of the study is to determine the emotion (temporary mental state) of human. Machine learning model will be developed to achieve the desired results.

The following are the objectives of the study:

- To study the different emotions/temporary mental state of human and to represent them in a graphical format.
- To develop machine learning model while considering emotion in different duration of the day (morning, afternoon and evening and night as well).
- To depict the emotions using emoji/ emotional depiction chart.
- To calculate the results (in percentage) of different emotions of human using text mining techniques.
- To measure the correctness of the model by using appropriate statistical past data of emotions of a human.
- To verify and validate the output with the help of currently available data.
- Training more data about emotions/temporary mental state to machine learning model to improve the performance of the system.

Chapter 2

DESIGN DETAILS AND IMPLEMENTATION

2.1 Overview of Methodology

The methodology is divided into different steps:

- 2.1 - Chats dataset
- 2.2 - Preprocessing of the datasets
- 2.3 - Python Library for Sentiment Analysis
- 2.4 - Feature extraction of the datasets
- 2.5 - Feature selection of the datasets
- 2.6 - Training
- 2.7 - Classification
- 2.8 - Flowchart of methodology
- 2.9 -Building chat application
- 2.10 -Integrating sentiment analysis module and chat application

2.1.1 Chats dataset

This dataset is provided by kaggle. This chat dataset will consist of 95,000 sentences (over 6,30,000 words) and it is the conversation between boy to close friends, girlfriend, their family members, their relatives and company supervisors, and we are going to define four categories like happy, stressed, angry and others on this dataset. The purpose of the dataset is to find the mental status of the students on the basis of their conversations.

This dataset is unbalanced in which labels are happy(4243), stress(5463), angry(5506) and others(14948).

```
In [6]: df.groupby("label")["id"].count().plot.bar()
```

```
Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x1b5155c80b8>
```

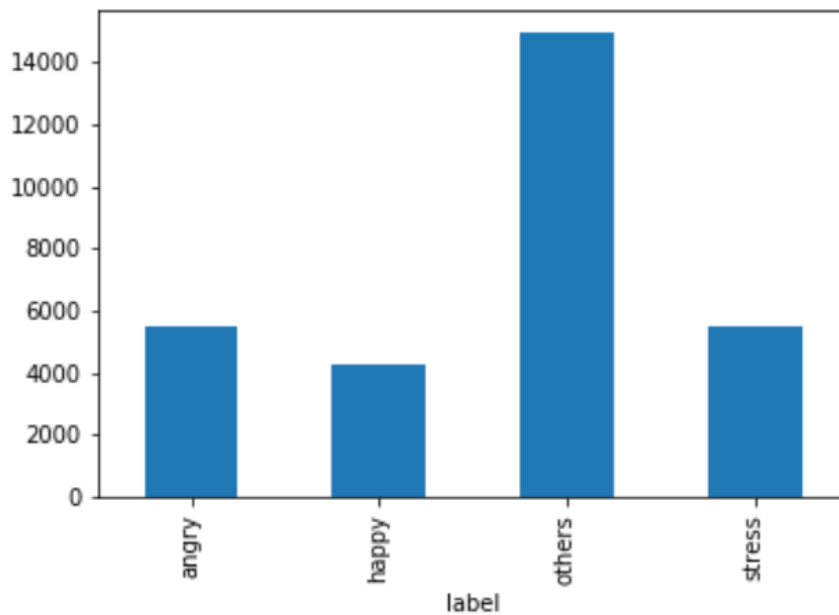


Figure 2.1: dataset.

2.1.2 Used Python Library

- scikit-learn
- nltk
- numpy
- pandas
- pickle
- joblib
- matplotlib and seaborn

Scikit-learn

[SN] Scikit-learn is the most popular python library. It is used for sentiment analysis and it is free ML software. Scikit-learn widely used for data mining and data analysis. There are other python libraries like scipy, numpy, and [Lib] matplotlib are related to it by some functionality. It contains different ML libraries for example svm, random forest, knn, k-means, etc. For complex data analysis, We use this library because it finest library. It has a cross-validation property that can help in using more than one metric. It contributed to training methods for their development, for example, nearest neighbors and logistics regression.

Features of Scikit-learn

- 1.[SN] Cross-validation: For finding the accuracy for supervised models of unknown data, We have multiple methods for it.
2. Unsupervised learning algorithms: We have many kinds of algorithms that work for unsupervised learning. For example factor analysis, clustering and principal component analysis to unsupervised neural networks
3. Feature extraction: Feature extraction helps extract the features from text and images.

Usefulness of Scikit-Learn

It is useful for implementing the data mining and machine learning task. Various tasks for example model selection, clustering, regression, classification and reducing dimensionality.

NLTK(Natural Language Toolkit)

The [NLT] NLTK platform gives approachable interfaces to more than fifty lexical and corpora sources mapped to machine learning algorithms, as well as a strong option of utilities and parsers. Apart from its facilities for sentiment analysis, the NLTK algorithms include named topic segmentation, part-of-speech (POS), tokenizing, and entity recognition. NLTK also possess a fine selection of third-party add-on, as well as the most broad extending language support of any of the libraries explained here.

On the contrary, this versatility can also be immense. The utter diversity of some of its tool classification (it has nine stemming libraries as against to SpaCy's single stemmer, for example) can construct the framework look like an unfocused grab-bag of [Too] NLP archive material from the last one and half decade. This could attach a layer of complication to our logistical planning and project ideation.

The constructive side of this is that no contestant to NLTK can brag such a useful and comprehensive base of documentation, as well as subordinate online

resources and literature. At liberty in progress prop up is supplied by a lively Google Group.

Numpy

It is considered as most popular python library in machine learning. [SN] Numpy used internally by TensorFlow for executing multiple operations on Tensors. Numpy most important features is Array interface.

Features of Numpy

- 1.Mathematics: [SN] It convert complex mathematical implementations to simple.
2. Interactive: It is easy to use and very interactive.
- 3.Lot of Interaction: It have a lot of open source contributions due to widely used.
- 4.Intuitive: It makes the concepts easy and makes coding real easy.

Where is Numpy Used?

Numpy interface can be deployed for sound waves, expressing images and other binary raw streams as an array of real numbers in N-dimensional. For full stack developers having knowledge of this library is important.

Pandas

Pandas is a machine learning library in [SN] python that gives various tools for analysis and data structure of high-level. One of the best features of this library is the ability to interpret complex operations with data using one or two commands.

Pandas have various inbuilt techniques for filtering, combining data, grouping and time-series functionality.

Features Of Pandas

Pandas has the ability to make the whole process of manipulating data will be easier. Support for operations such as Visualizations, Concatenations, Aggregations, Sorting, Iteration and Re-indexing.

Where is Pandas Used?

At present,[SN] Pandas library have few releases which includes changes in API, enhancements, bug fixes and hundreds of new features. Despite being under development it has ability to sort data and group, gives support for performing custom types operations and select best suited output for the apply method. Data Analyst often uses this library. When We use Pandas library with other libraries than that will produce a good amount of flexibility and high functionality.

pickle

For de-serializing and serializing python object structures, [Pyt] pickle module can be used. It is also called flattening and marshalling. Serialization is the process of transforming an object in memory to a byte stream and that byte stream can be kept on disk or forward over a network. After that, this character byte stream can then be get back and de-serialized back to a python object. Compression is other than Pickling. This is basically used for transforming of an object from one depiction(data in Random Access Memory(RAM)) to another (text on disk), than afterwards the process of encoding data with fewer bits, by which we can save the disk space.

Where is pickle Used?

Pickling is important for [Pyt] execution where you requirement some degree of perseverance in your data.State data of program can be stored to disk, so you can continue working afterwards. Pickle can be used to send data over a socket connection or Transmission Control Protocol(TCP), or to save python object in a database. Pickle is very important library of python for machine learning algorithms. Which you can use for making new prediction after some times, without writing again the same content or there is also no need of training of model again.

Matplotlib

For plotting 2 Dimensional graphics, We required [Lib] matplotlib.pyplot library in python language. Matplotlib can be utilized in web application servers, python scripts, shell and user interface toolkits. It has a variety of toolkits for python matplotlib functionality. In which some of them are unconnected downloads and some of them have external dependencies.

- Basemap: Basemap is a map plotting toolkit and it has different map projections, political boundaries and coastlines.

- Cartopy: Cartopy is a mapping library featuring image transformation capabilities, polygon, line, arbitrary point and object map projection definitions.
- Excel tools: Matplotlib gives functionality for interchanging data with Microsoft Excel.
- Mplot3d: It can be used for 3-D plots.
- Natgrid: Natgrid is an interface to the natgrid library for irregular gridding of spaced data.

Joblib

It is a group of tools to give [Doc] lightweight pipelining in Python.

In specific:

1. Joblib is easy simple parallel computing and it is also robust and fast on huge data in specific and has particular optimizations for numpy arrays.
2. It has function like lazy re-evaluation (memoize pattern) and transparent disk-caching

Main features:

1. Embarrassingly parallel helper: It help to write readable parallel code and debug it quickly.
2. Fast compressed Persistence: It is substitution for pickle to save efficiently on Python objects holding large data (`joblib.dump` `joblib.load`).
3. Transparent and fast disk-caching of output value: It is functionality for Python functions that service well for arbitrary Python objects, including very large numpy arrays. It has different persistence and flow-execution logic from domain logic or algorithmic code by putting down the operations as a group of steps with well-defined inputs and outputs: Python functions. It can store their computation to disk and return it when it is needed.

2.1.3 Preprocessing of the dataset

Once the dataset is collected from the chats after that the next is pre-processing of dataset before implementation. During the pre-processing various steps are involved, they are,

1. Convert all uppercase letters to lowercase
2. Remove all stopwords
3. Spelling correction
4. Remove Non-English word

5. Tokenization
6. Stemming and Lemmatization
7. join the splited text

2.1.4 Feature extraction of the datasets

Selection of the useful words from the chat datasets is known feature extraction. In this feature extraction step we extract the aspects of the pre-processed chat dataset.

1. Features like unigram, bigram, ngram
2. POS (parts of speech tagging like nouns, verbs, adjectives, adverbs and organisations etc.)
3. Position of terms
4. Negation

2.1.5 Feature selection of the datasets

Selection of correct feature techniques take an important role in identifying attributes and improving classification accuracy. They are categorised as -

1. NLP (Natural Language Processing)
2. Statistical
3. Hybrid
4. Clustering

2.1.6 Training

Before training the datasets, we convert the raw text to feature vectors that can be used as input estimators using TfidfVectorizer. After that, we trained the datasets(vectors) using different machine learning techniques and then we can predict the unknown data.

TF-IDF: TF-IDF is a combination of two terms tf and idf and can be calculated as a by-product of tf and idf. TF-IDF is a weight that is used for information retrieval and data mining.

TF(Term frequency): It measures the frequency of words in the documents.

$tf = (\text{number of time word } W \text{ in document} / \text{total number documents with word } W \text{ in it})$

IDF(Inverse document frequency): It measures the importance of word within the document.

$idf = \log(\text{total number of documents} / \text{number of documents with word } W \text{ in it})$

$tf-idf = tf * idf$

2.1.7 Classification

After pre-processing, feature engineering and training the data, classification is done by using SVM.

SVM : Support vector machine is the machine learning model used for classification, regression and outlier detection. SVM creates a hyperplane or a group of hyperplanes in a high or infinite-dimensional space.

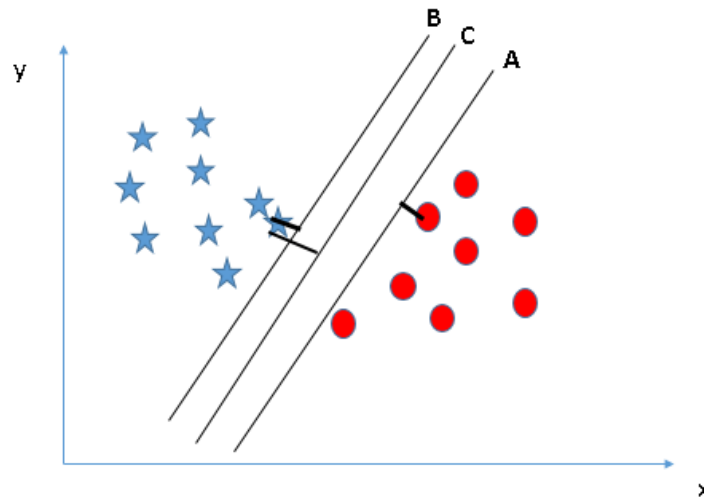


Figure 2.2: SVM figure 1

Margin: The distance between the nearest data point is called the margin. In the above figure, we can see that the three hyper-plane A, B, and C in which

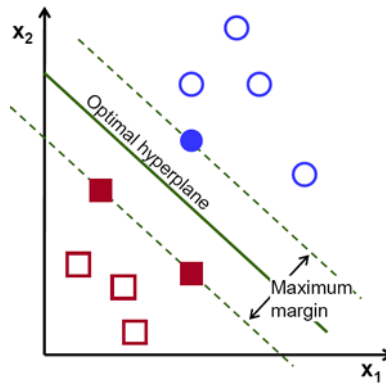


Figure 2.3: SVM figure 2

hyper-plane C has the highest margin from support vectors because it has the highest distance from support vectors. Therefore, the hyper-plane C is the resultant hyper-plane which is used for classification.

2.1.8 Flowchart of methodology

Graphical representation of used methodology

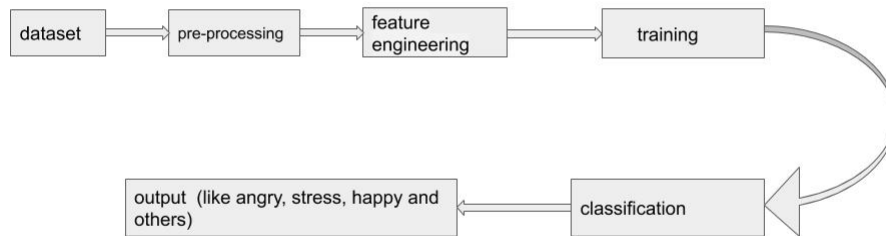


Figure 2.4: Graphical representation of used methodology.

2.1.9 chat application(frontend)

The chat application works in a way that each user connects via a single link. After accessing the link they registered their names and start sending messages to each other. whenever a user sends the message it is visible to all users who are connected through this link. Suppose there are two users and they are chatting with each other. When user A sends a message to user B, user B gets the message with the polarity of the message. We have defined the polarity of messages in happy, sad, angry, neutral and others. After seeing the message the user B is able to predict the mood of the user A. Then the user B starts responding according to user A mood. We express the mood by using emojis in our chat application using emotion analysis module, which is running in the backend. Server sends the messages to the machine learning module to check the polarity of the message and sends an emoji according to the message.

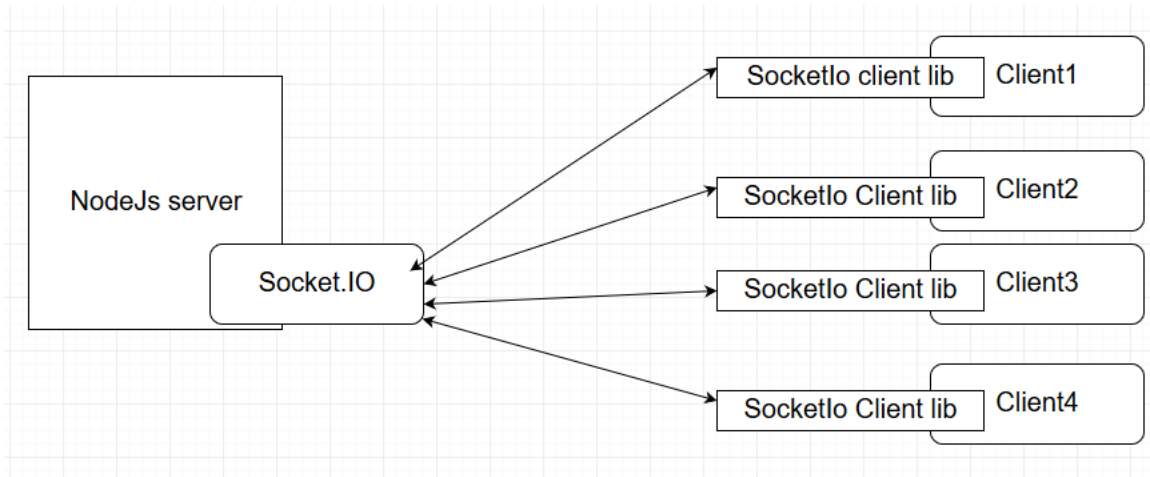


Figure 2.5: working of chat module

Chat Application Module: The first step approaching the development of the entire project was to develop the chat application. To develop the chat application we are going to use node.js, react.js, socket.io(socket programming) and style sheet to style the user interface. Socket.IO is a library that facilitates the bidirectional, real-time and event-based connection between the browser and the server. The advantage of using socket.IO is that it can establish a connection in the presence of proxies, load balancers, personal firewall, and antivirus software. Node.js will handle the HTTP request made by the client to the server. To build the user interface we are going to use react.js.react.js is a front end (UI) library that promotes the production of rich internet applications (RIA). React.js gives way to create stateful, reusable interactive UI components to make our user interface more attractive and easy to use we are going to use a cascading style sheet to style the components. The working of user interface

is first of all user has to submit his/her name then he will be able to send the message. In the message field we will show the message with the name of user and the polarity of the messages. We can add as many users as we want.'

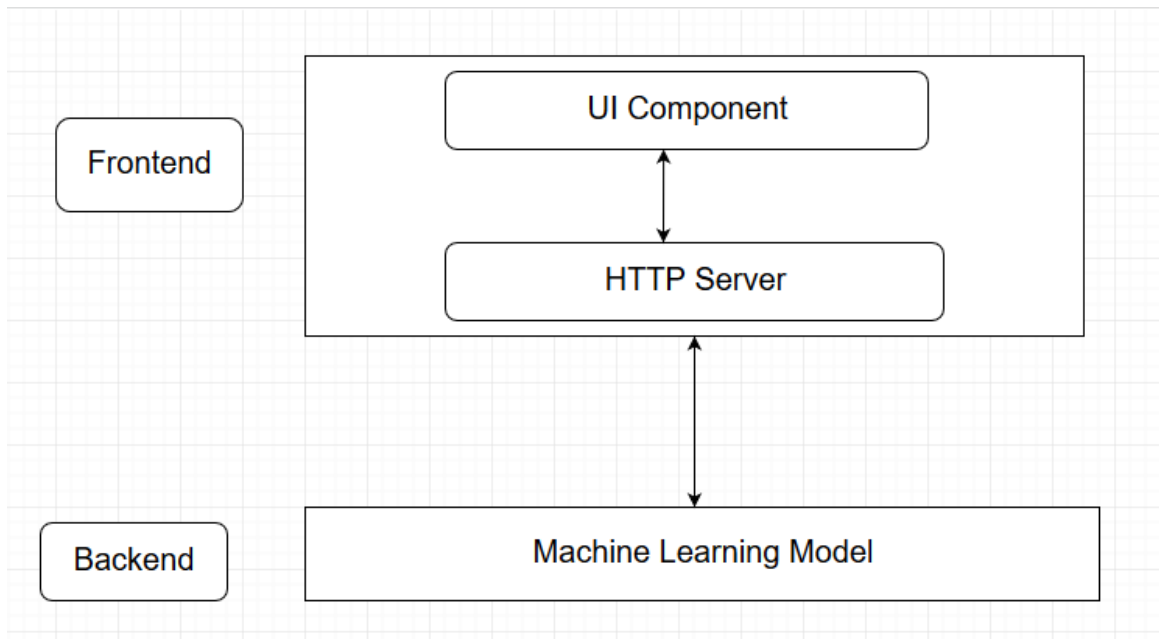


Figure 2.6: Working of Complete Application

Chapter 3

RESULTS AND DISCUSSION

3.1 RESULTS

Here the project methodology is completed into different steps such as collecting dataset, preprocessing on the dataset, feature engineering, training, and classification. For performing tasks on the dataset, firstly it has been splitted into a training set(80 percentage) and test set(20 percentage). After that, we trained the training set using the SVM model and Naive Bayes and then We predict the unknown data or test set and get different results, such as SVM with 85.6 percentage accuracy and naive Bayes with 77.8 percentage.

3.2 DISCUSSION

Nowadays, It has been seen that the text-based emotion detection plays an important role for human being to interact with computers on their chats but the emotion-based search engine does not give the very accurate result due to many reasons such as complexity of English language, unbalancing of labels, model and feature selection, word sense disambiguation and noises in the data, and the result can be improved by using deep learning models like fastai, neural networks, and word embedding.

ML chat web app

Emotion detection of chats

Enter Your message Here

Do you dance?,Yes I love to dance 🥳,😄😄😄 so you have legs too,

predict

Figure 3.1: Text Input.

ML chat web app

Emotion detection of chats

Results of messages

Happy 😊

Figure 3.2: Output.

Chapter 4

CONCLUSION

Nowadays, the field of emotion detection is an exciting experiment of real-world applications where we discover the people's opinion which is important or not in better decision making. Here, We are still trying to detect the emotion of corpus texts accurately due to complexity in the English language. Currently, many people have started showing their emotions or opinions on the Web that enlarge the need for examining the opinionated online content for various real-world applications.

In this project, we are trying to show the basic way of classifying chats into happy, angry, stress and others category using support vector machine and how language models are related to the SVM and can produce the better results. We can further improve our classifier by using more features from the chats, tuning the parameters of SVM classifier or trying another classifier.

Bibliography

- [HCR93] Elaine Hatfield, John T Cacioppo, and Richard L Rapson. “Emotional contagion”. In: *Current directions in psychological science* 2.3 (1993), pp. 96–100.
- [KP07] Kotsiantis and P. Pintelas. “Supervised machine learning: A review of classification techniques”. In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.
- [YP07] Byungun Yoon and Yongtae Park. “Development of new technology forecasting algorithm: Hybrid approach for morphology analysis and conjoint analysis of patent information”. In: *IEEE Transactions on Engineering Management* 54.3 (2007), pp. 588–599.
- [Kum+08] Sudhir Kumar et al. “MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences”. In: *Briefings in bioinformatics* 9.4 (2008), pp. 299–306.
- [MM11] Louis-Philippe Morency and Rada Mihalcea. “Towards multimodal sentiment analysis: Harvesting opinions from the web”. In: *Proceedings of the 13th international conference on multimodal interfaces. ACM* (2011).
- [KS12] Akshi Kumar and Teeja Mary Sebastian. “Sentiment analysis on twitter”. In: *International Journal of Computer Science Issues (IJCSI)* 9.4 (2012), p. 372.
- [She12] Nilesh M.and Vilas Thakre Shelke. “Survey of techniques for opinion mining”. In: *International Journal of Computer Applications* 57.13 (2012), pp. 0975–8887.
- [Mat+14] Nidhi Mathen et al. “Online Promotions: Exploring the emerging opportunity in Indian market”. In: (2014).
- [Med14] and Ahmed Hassan Medhat Walaa. “Sentiment analysis algorithms and applications: A survey”. In: *Ain Shams engineering journal* 5.4 (2014), pp. 1093–1113.
- [LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), p. 436.

- [Por+16] Soujanya Poria et al. “Fusing audio, visual and textual clues for sentiment analysis from multimodal content”. In: *Neurocomputing* 174 (2016), pp. 50–59.
- [YC17] P. Yang and Y. Chen. “survey on sentiment analysis by using machine learning methods”. In: *2017 IEEE 2nd Information Technology, Network-ing, Electronic and Automation Control Conference (ITNEC)* (2017), pp. 117–121.
- [YD17] B.Djumabaev Y.Arslan A.Birturk and D.Kucuk. “Real-time Lexicon-based sentiment analysis experiments on Twitter with a mild (more information, less data) approach”. In: *2017 IEEE International Conference on Big Data (Big Data)* (2017), pp. 1892–1897.
- [Doc] joblib Documentation. In: (). URL: <https://buildmedia.readthedocs.org/media/pdf/joblib/latest/joblib.pdf>.
- [Lib] Matplotlib Library. In: (). URL: <https://www.edureka.co/blog/python-matplotlib-tutorial/>.
- [NLT] NLTK. In: (). URL: <http://www.nltk.org/>.
- [Pyt] Pickle in Python. In: (). URL: <https://www.datacamp.com/community/tutorials/pickle-python-tutorial>.
- [SN] pandas Scikit-Learn and Numpy. In: (). URL: <https://www.edureka.co/blog/python-libraries/>.
- [sen] sentiment140. In: (). URL: <http://www.sentiment140.com/>.
- [Too] Natural Language Toolkit. In: (). URL: <https://www.iflexion.com/blog/sentiment-analysis-python>.