TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A Minor Project Proposal

On

MACHINE LEARNING IMPLIED SENTIMENT ANALYSIS AND PERSONALITY PREDICTION

SUBMITTED BY

Gokarna Adhikari

075BEI 014

Submitted to

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

(December 12, 2021)

Abstract

The minor project we completed is a fully software based web application that uses machine learning techniques that can analyze the sentiment of the textual data and predicts the personality based on the questionnaire with satisfactory accuracy.

The project contains two parts: Sentiment analysis part and Personality prediction part.

AI, Artificial Intelligence is a rapidly evolving process in context of today's world. The fact that the machine can approach to the behavior of the human is the arguably the most interesting topic in present context.

The project also revolves around the idea of "machine approaching behavior of a human". We plan to create a *fully web based application*, which uses *a suitable machine learning* technique to master textual sentimental related task. In this project, primarily we aim to master our application *sentimental analysis*, besides we would like to incorporate the *personality prediction* features in our web application as well.

The reasons for selecting this topic for our project is its versatility and the wide learning opportunity it will provide that includes multiple yet correlated machine-learning topics.

Acknowledgement

We would like to express our humble and sincere gratitude to the Department of Electronic and Computer Engineering, Central Campus, Pulchowk for providing us the opportunity to choose the project work which is enlisted in the syllabus of third year as per the Minor project IOE, Tribhuvan University.

We would like to extend our heartfelt thanks and gratitude to the college for the arrangement and support for working in the academic project. We are particularly indebted to our revered supervisors **DR .Nanda Bikram Adhikari, DR. Dibakar Raj Pantha**, **Dr.Surendra Shrestha**, **Prof. DR. Ram Krishna Maharjan, Mr. Lok Nath Regmi and all the other respected faculty members** for their constructive and encouraging suggestions regarding this project, explaining the concepts of sentimental analysis and personality prediction, providing us the guidelines for using software and analysis and motivating us for progressing the project work.

We place on record, our sincere gratitude to one and all who, directly and indirectly have lent their helping hand and suggestions in this venture.

Table of Contents

List of figures

Introduction

The project we completed is a web application that uses a machine-learning model to analyze the sentiment of the textual data and predict the personality based on questionnaire. We have given our project the name "THREADER".

Background

Sentimental analysis analyses the textual data and processes the emotions hidden into them. It is one of the rising and popular topics in the field of machine learning. For a machine to approach the behavior of a human, it is mandatory that it can interpret the emotions of the statements expressed by a human. The topic is popular because it has a versatile application in various fields of industry.
Personality prediction is the other part of our project that predicts the personality of a person based on psychological questionnaire.

Motivation

In present context, communication through texting via internet is becoming more and more popular. Vast majority of organizations use online feedback system to enhance the quality of service they provide. For this, the sentimental analysis is one of the vital aspect for knowing how they are performing from the perspective of their customers. A human could do the feedback analysis but the task seems very tedious and time consuming.
The personality prediction test assist to clarify a clinical diagnosis, guide therapeutic interventions, and help predict how people may respond in different situations.
Using a machine to analyze the sentiment of large number of feedbacks received by an organization, we thought, is a very good idea. It can provide future insight for the organization and aid the public to take decision on their purchase. Similarly, multinational big tech giant companies like Meta (Facebook), Alphabet (Google), Twitter, Amazon etc. are also investing huge resources on the research and development of a good sentimental analysis model as well.

Objectives

We plan to create a web application, which uses a suitable machine-learning algorithm to analyze the sentiments of a text and perform personality prediction test with reasonable amount of error. We primarily aim to master our application to perform sentimental analysis with 0.9+ accuracy and perform personality prediction that could be accurate for at least quarter of people.

Scope of project

Threader has versatile scope and serve numerous applications some of which are listed below:

i. Support and feedback
   - It helps for the analysis of the product based on the sentimental analysis of the feedbacks done by the customers. It can assure a good and respectable place for the voice of both customers and employees.

ii. Personality detector
   - It can help an organization review the employees, recognize a good candidate on an interview in case of new hiring etc.

iii. Recommender system
iv. Social media monitoring
v. Chat bot

System Features

The main feature of our web application is that it helps to determine the opinion of the people on products, government work, movie review, politics or any other by analyzing the texts. Our system is capable of training the new text taking reference to previously trained texts. We have feedback-receiving interface to receive the feedback of the result. If the model prefects the wrong result, the model finds that it predicts the wrong result through the feedback system and finds the correct result and it uses the feedback data to improve the accuracy of the model by itself.

The computed or analyzed data will be represented in various diagrams such as Bar Graph.

Literature Review

Sentimental analysis analyses the textual data and processes the emotions hidden into them. For a machine to effectively analyze the emotions of a text written or stated by a human is onerous task. Hence, the accuracy parameters comes into play while developing the application.

Different papers had been published about the sentimental analysis in the past; three most relevant of them are discussed here.

i. **Topic**: A Sentimental Education: Sentimental Analysis using subjectivity summarization n based on Minimum Cut.
Author: BoPang and Lillian Lie [Cornell University]
Description: This paper proposed that SVM and NB are better technique for improving the performance of a model up to 86.4 %.

ii. **Topic**: Automatic Sentiment Analysis in Online Text
Author: Erik.Boiy, Pieter Hens, Koen Dschacht and Marie Francine Moens
Description: This paper shows the varying level of accuracy when symbolic and machine-learning methods were applied to different social network dataset.

iii. **Topic**: Sentiment Analysis in data of Twitter using Machine learning algorithm
Author: Dr. Sefer Kurnaz, and Mustafa Ahmed Mahmood.
Description: This paper proposes a new technique that offers an accuracy of 98 % when compared with Deep learning method, SVM and Maximum Entropy method.

After analyzing the works done on these fields, we found out that following algorithms are used in existing research works:

i. Rule bases
ii. Naïve Bayes
iii. Support Vector Machine
iv. Multilayer perception
v. Maximum entropy
vi. Decision Tree

vii. Convolution neural network
viii. Bayesian Network

Theoretical background

The project will incorporate a suitable machine learning algorithm for achieving the entire desired task.

Machine learning

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning.

Approaches

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system**.**

Supervised learning

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Types of supervised learning algorithms include active learning, classification and regression.

Unsupervised learning

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. However, unsupervised learning encompasses other domains involving summarizing and explaining data features.

Reinforcement learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov decision process (MDP). Many reinforcement-learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

Algorithms

Any machine learning algorithms can be analyzed based on their three major components.

Representation

Different methods can be incorporated to represent machine learning algorithms into various types, but we have only included those algorithms that were implemented on our project.

i. Support vector machines

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

ii. Decision Trees

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

iii. Random Forest Classifier

The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

Evaluation

A machine learning algorithm can be evaluated on basis of different parameters. The project was completed with the classification algorithms only. Hence, following parameters are analyzed closely in context of our project.

i. Accuracy

Accuracy is how close or far off a given set of measurements (observations or readings) are to their true value. Precision is how close or dispersed the measurements are to each other.

ii. Balanced Accuracy

Balanced accuracy is calculated as the average of the proportion corrects of each class individually.

Balanced accuracy = (Sensitivity + Specificity) / 2

- Sensitivity: The "true positive rate" – the percentage of positive cases the model is able to detect.
- Specificity: The "true negative rate" – the percentage of negative cases the model is able to detect.

iii. F1-score

The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision.

The **F1-score** of a classification model is calculated as follows:

$$\frac{2(P * R)}{P + R}$$

P = the precision
R = the recall of the classification model

iv. Precision

Precision is how close or dispersed the measurements are to each other.

Optimization

There are mainly three optimization technique for machine learning algorithm:

i. Combinational Optimization
ii. Convex Optimization
iii. Constrained optimization

Based on these three components we would select the best algorithm fit for our project.

REQUIREMENT ANALYSIS

Feasibility Study

A feasibility study takes into account various constraints within which the system should be implemented and operated. In this stage the resources needed for implementation such as computing equipment, manpower and costs are estimated. The estimates are compared with available resources and a cost benefit analysis of the system is made. The main objectives of the feasibility study are to determine whether the project would be feasible in terms of the following categories:

- Technical feasibility
- Economic feasibility
- Operational feasibility
- Schedule feasibility

Technical Feasibility

Firstly, we proposed to use java to develop the desktop application. After some feasibility study, we find that the platforms and the process are both outdated. We proposed to use NodeJS to develop the web application and python to train and test the machine learning model.

Since the web application using software technologies and tools are freely available and technical skills required can be easily manageable and the system servers are adequate and manageable in future.

It is found that the hardware and software meet the needs of the system. It is clear that the proposed project is technically feasible.

Economic Feasibility

Economic feasibility attempts to weigh the costs of developing and implementing a new system, against the benefits that would gather from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

Since the tools and technology that are going to be used are freely available and open source, there will not be an economic problem going through the project.

So the project is economically feasible.

SCHEDULE FEASIBILITY

We have a time span of three month to complete the project. Since the project was economically and technically feasible, based on the initial feasibility study we find that the project can be completed within the given span of time.

Hence, we came to the conclusion that the project is schedule feasible.

Functional Requirements

A functional requirement is a description of the service that the software must offer. It describes a software system or its components. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification.

The following are the functional requirements of our project

- Text entry for sentimental analysis.
- User interface.
- Opinion entry for personality prediction.
- Feedback providing interface.

- Result visualization.

Non-Functional Requirements

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The plan for implementing functional requirements is detailed in system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually architecturally significant requirements.

Based on these, the non-functional requirements of the project are as follows:

- The system should be easy to use, user friendly in operation.
- The system should have a very good response time.
- Secure and reliable

Methodology

The tools and technology used, software development approach, the system block diagram and overall system design process is discussed on this section.

Tools and technology

Python

Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. Rather than focusing on loading, manipulating and summarizing data, Scikit-learn library is focused on modeling the data.

NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

Pandas

Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python

ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.

JavaScript

NodeJS

Nodejs is a free open-source server environment which runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.) . It uses JavaScript on the server. Nodejs can generate dynamic page content and can create, open, read, write, delete, and close files on the server. It can also collect form data and add, delete , modify data in your database.

ReactJS

React is an open source, JavaScript library for developing user interface (UI) in web application. React is developed and released by **Facebook**. Facebook is continuously working on the *React* library and enhancing it by fixing bugs and introducing new features. ReactJS is a simple, feature rich, component based JavaScript UI library. It can be used to develop small applications as well as big, complex applications. ReactJS provides minimal and solid feature set to kick-start a web application. React community compliments React library by providing large set of ready-made components to develop web application in a record time.

MySQL

MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).

MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it's clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL

Software development approach

We developed the project as a web application *ReactJS* and *NodeJS* that incorporates different machine learning techniques, written in *Python* to achieve entire desired tasks.

We use the prototype approach for software development.

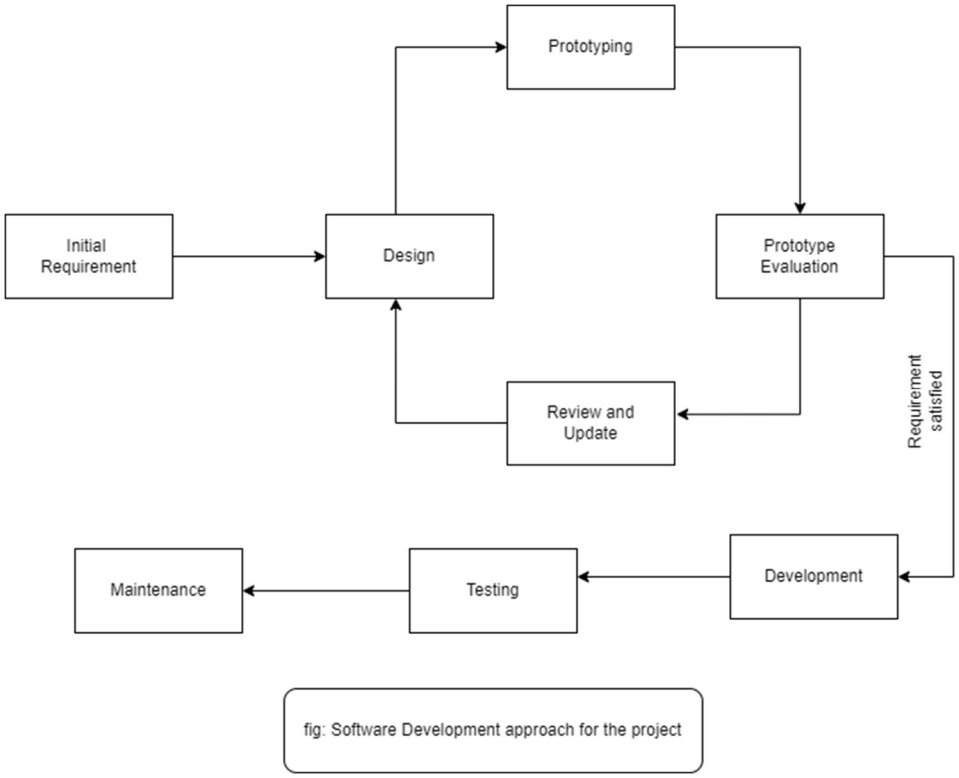


fig: Software Development approach for the project

Firstly, we begin with initial requirements gathering, and the requirements of the system are defined in detail. And the gathered requirements were analyzed. When requirements are known, a preliminary design or a quick design for the system is created. It was not a detailed design; however, it includes the important aspects of the system, which gives an idea of the system. It helps in developing the prototype.

Information gathered from quick design is modified to form a prototype. The first prototype of the required system is developed from quick design. It represents a 'rough' design of the required system. The prototype was thoroughly evaluated by us team members. And strength and weakness were identified. We list the weaknesses and try to cover them in the next prototype. And the next prototype was evaluated in the same way as the previous. This process was continuously processed to minimize the weaknesses and to satisfy all the requirements which were gathered at initial state and those which were listed during prototype evaluation. When all the requirements were addressed, we developed the final project. The final system is thoroughly evaluated and tested.

For each prototype cycle

The overall software development process for each prototype cycle:

i. Data collection
   - All the data used in the project was collected from ***https://www.kaggle.com/*** which is very reliable source on internet

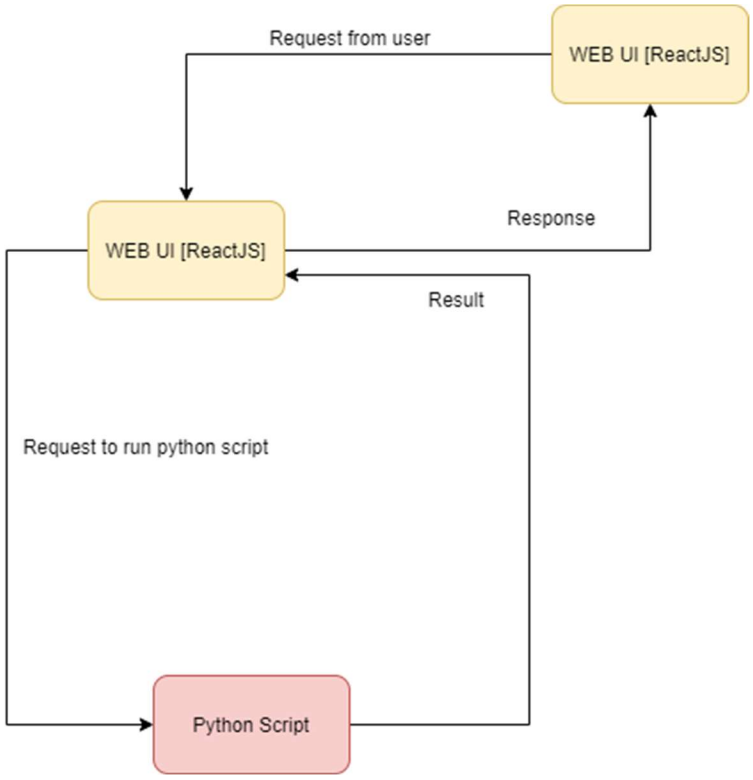
ii. Training /Testing and model development
   - For sentiment analysis, we used Support Vector Machine and Multinomial Naïve Bayes classifier and for the personality prediction, we used Random Forest classifier.

iii. GUI development
   - Graphical user interface development as a web application

iv. Integrating models and the GUI
   - Integrating the developed model with the web application

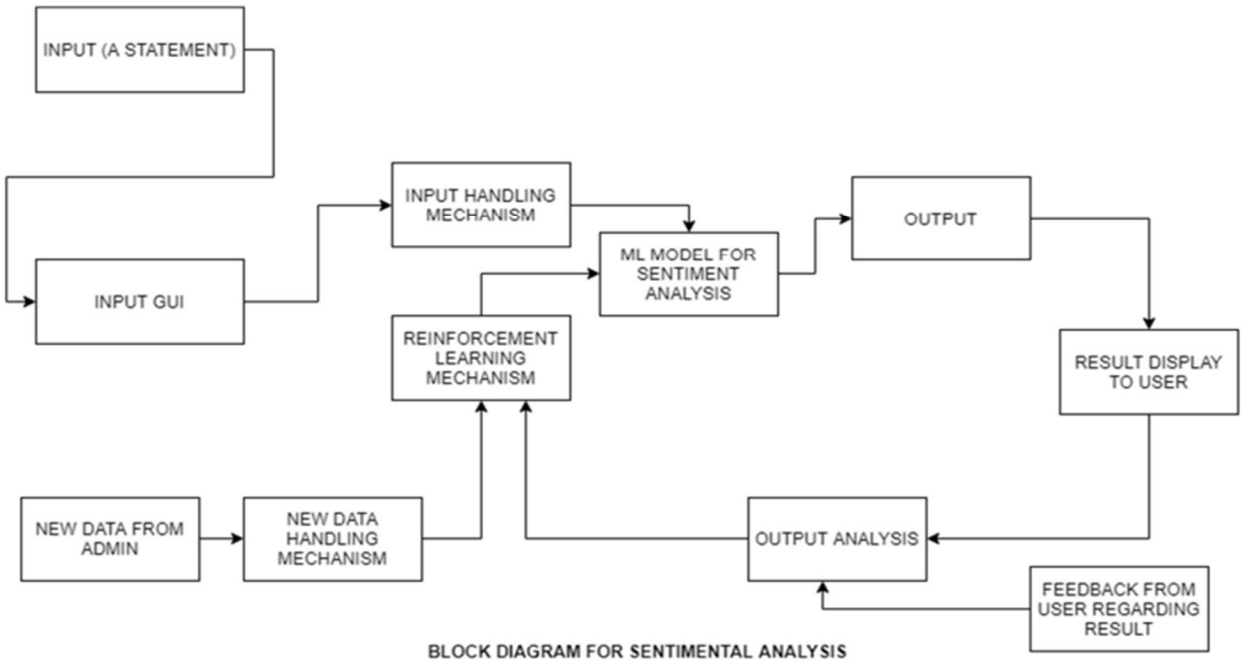
System block diagram

Overall System



The comprehensive system block diagram of all the components along with the explanation is discussed in this section. Firstly, the webserver takes request form user and the request was sent to the backend server. Backend server load model and predict the data and send to frontend server where user can visualize the data.

Sentiment analysis



BLOCK DIAGRAM FOR SENTIMENTAL ANALYSIS

Sentiment Analysis in our project predicts the sentiment of the test written in English language. This section can be divided into following subsystems:

1. Text Validation System
2. Text Processing System
3. Model Training Section
4. Result Combination System

Main system

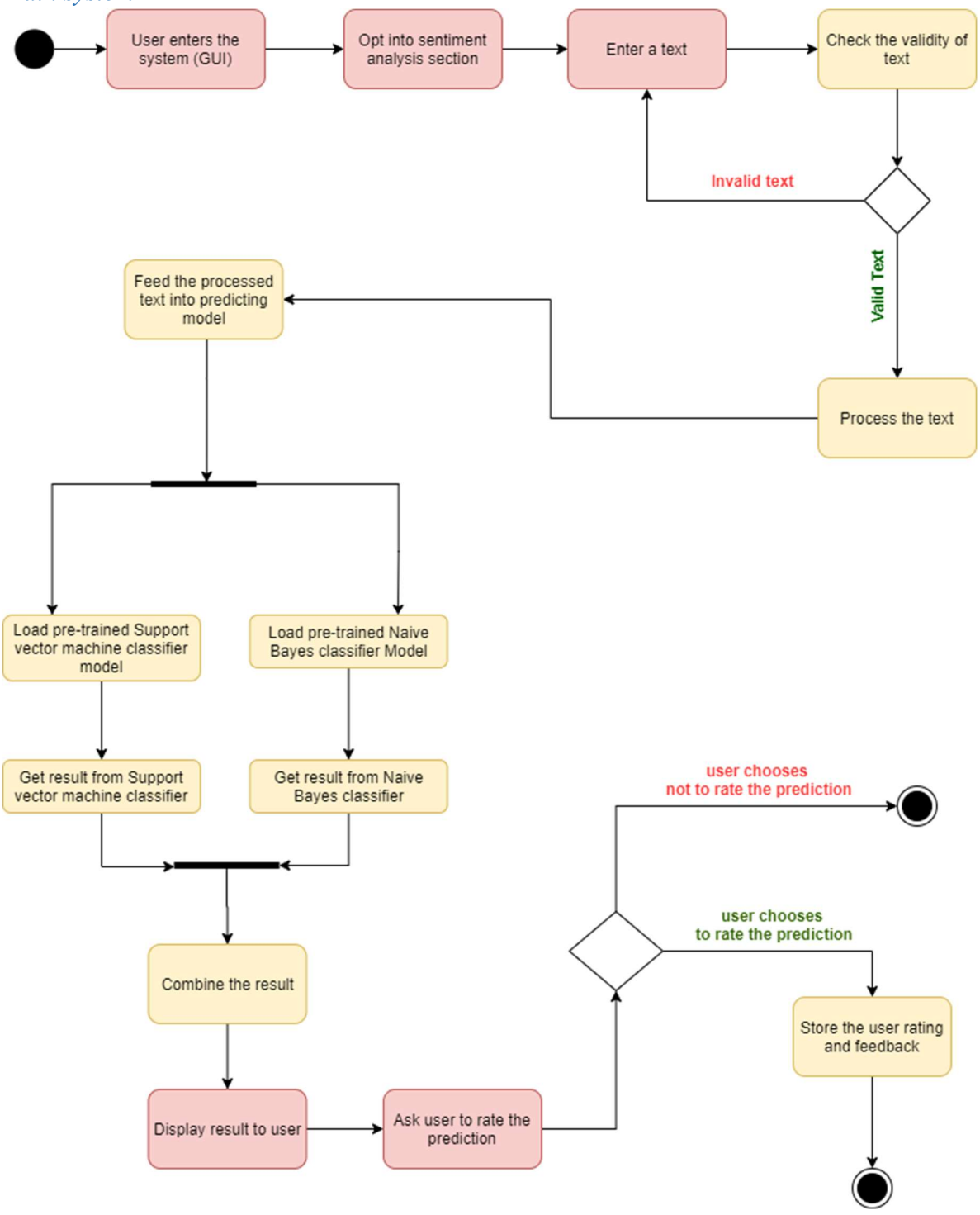


Figure 1: Activity Diagram for Sentiment Analysis

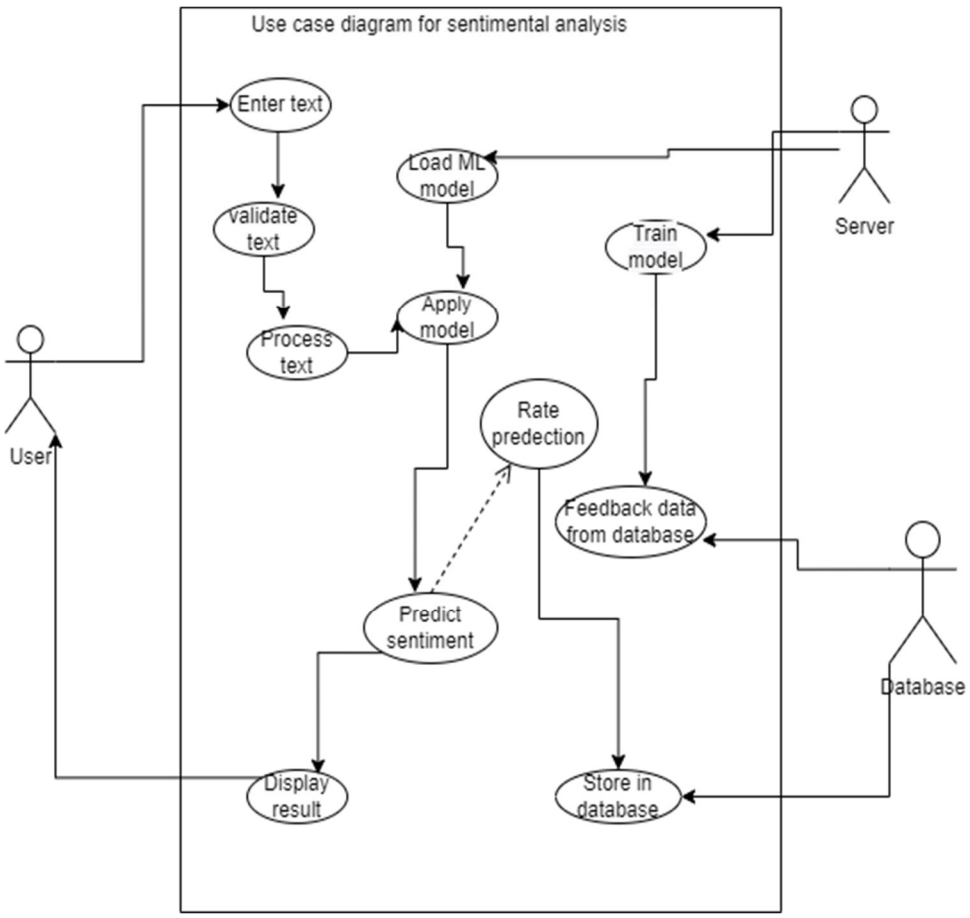


Figure 2: Use case for Sentiment Analysis

The main system:

- User enters the system and opts into sentiment analysis
- He enters the text to be analyzed
- The system checks the validity of text
- System Processes the text
- The processed text is now feed into pre-trained ML model
- Obtain and show the result to user
- Receive the rating from user

Text validation system

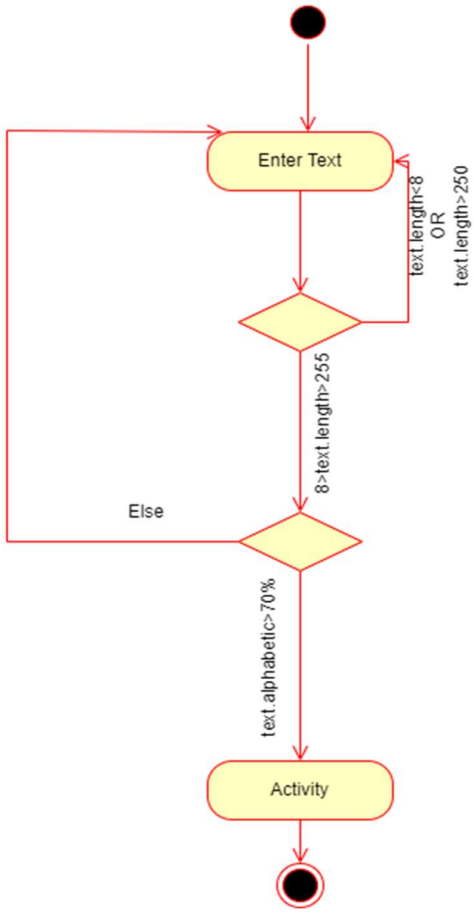


Figure 3: Activity diagram for Text Validation system

Scenario/ Description

1. Start
2. Enter the text.
3. Then direct to signup option.
4. Check if the length of entered text is greater than 8 and smaller than 250 .

    a. If yes proceed the process.

    b. Else enter text again.

5. Validate the text has more than 70%.

a. If yes proceed the process.
b. Else enter text again.

6. Feed the data preprocessors.

7. End

Text Processing System

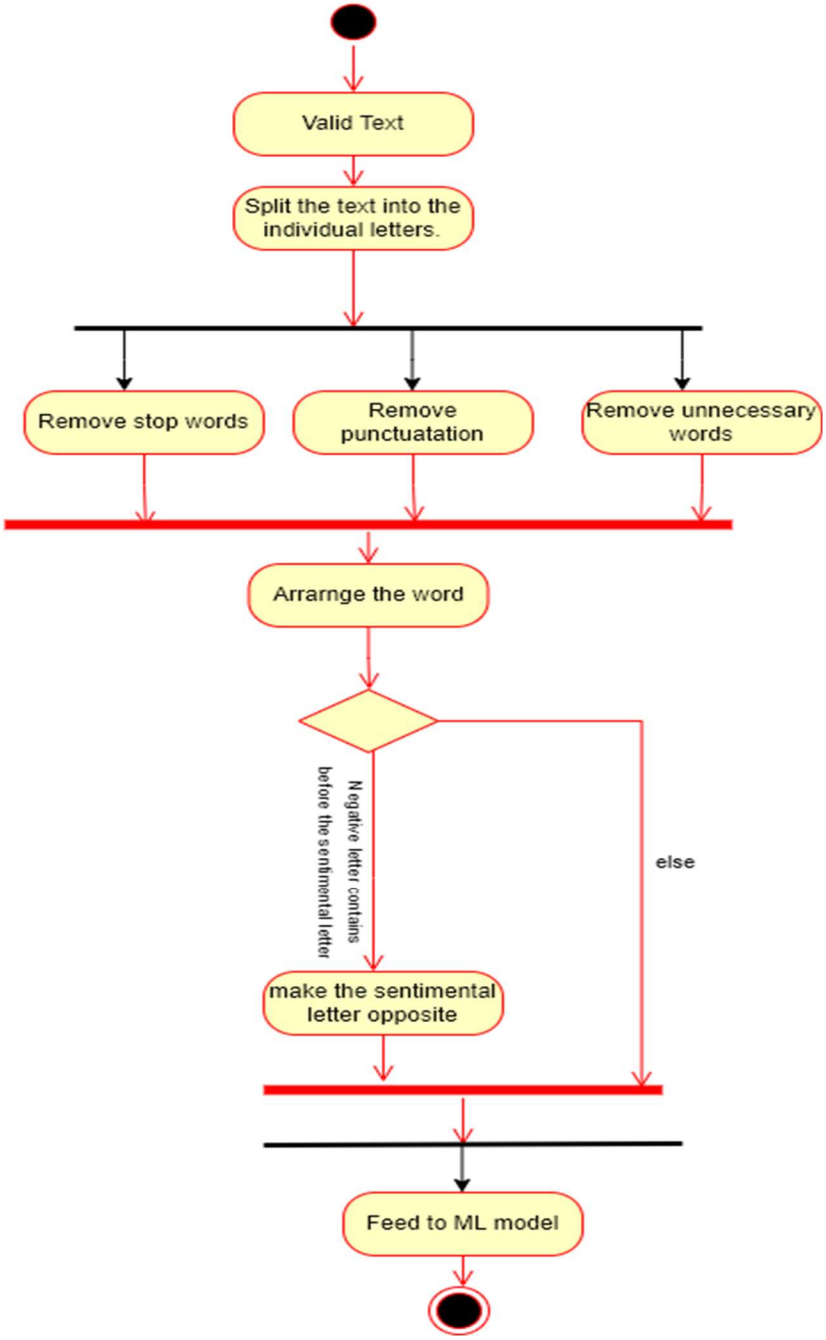


Figure 4: Activity diagram for text processing system

Scenario/ Description

1) Start.
2) Valid text input.
3) Split the text into individual letters called tokens.
4) And remove

a) Stop words.
b) punctuation
c) Unnecessary words

5)Arrange the words.

6) Check if negative letter word is contained before the sentimental words.

a) If yes make the sentimental letter opposite.

b) Else proceed further.

7) Feed to ML model.

8) End.

Model Training System

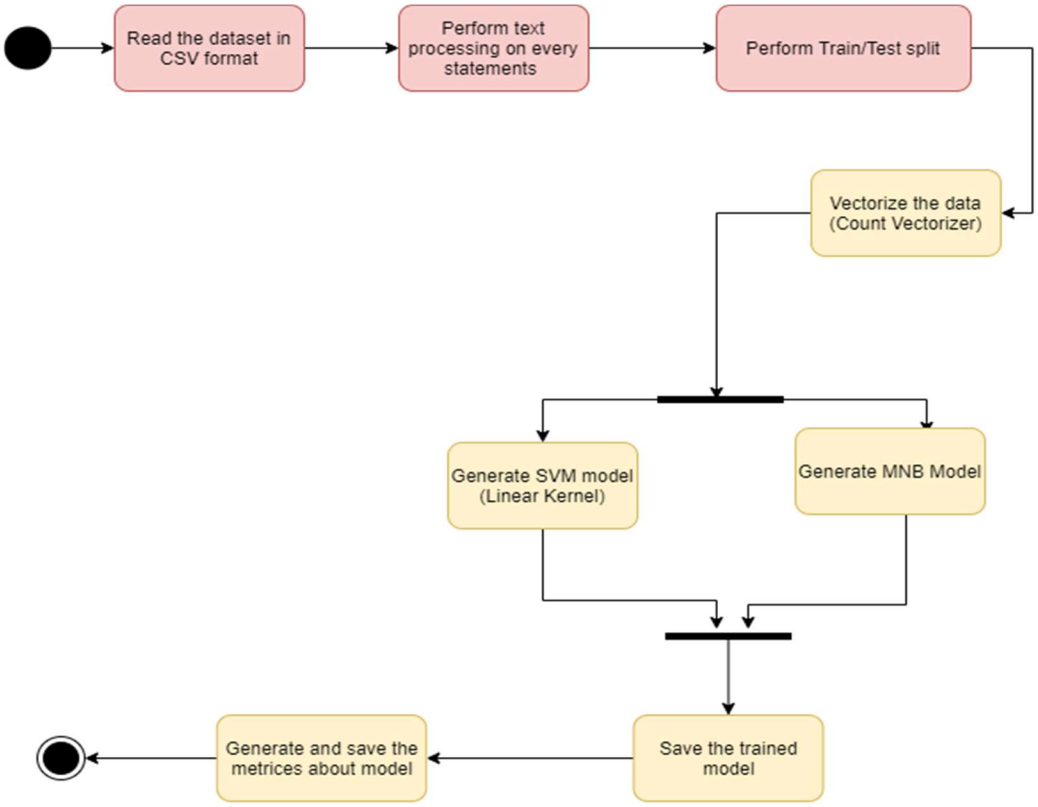


Figure 5: Model training system, activity diagram

Result Combination system

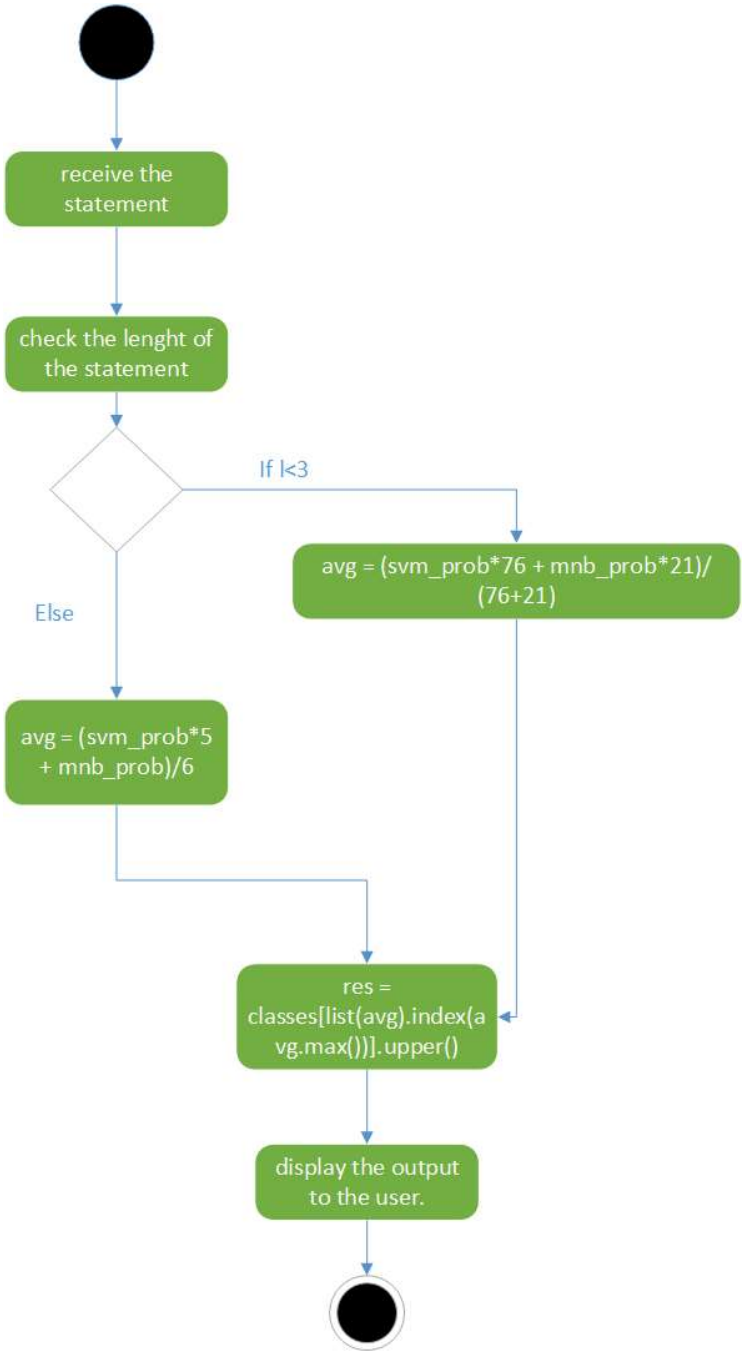


Figure 6: Result combination system, Activity diagram

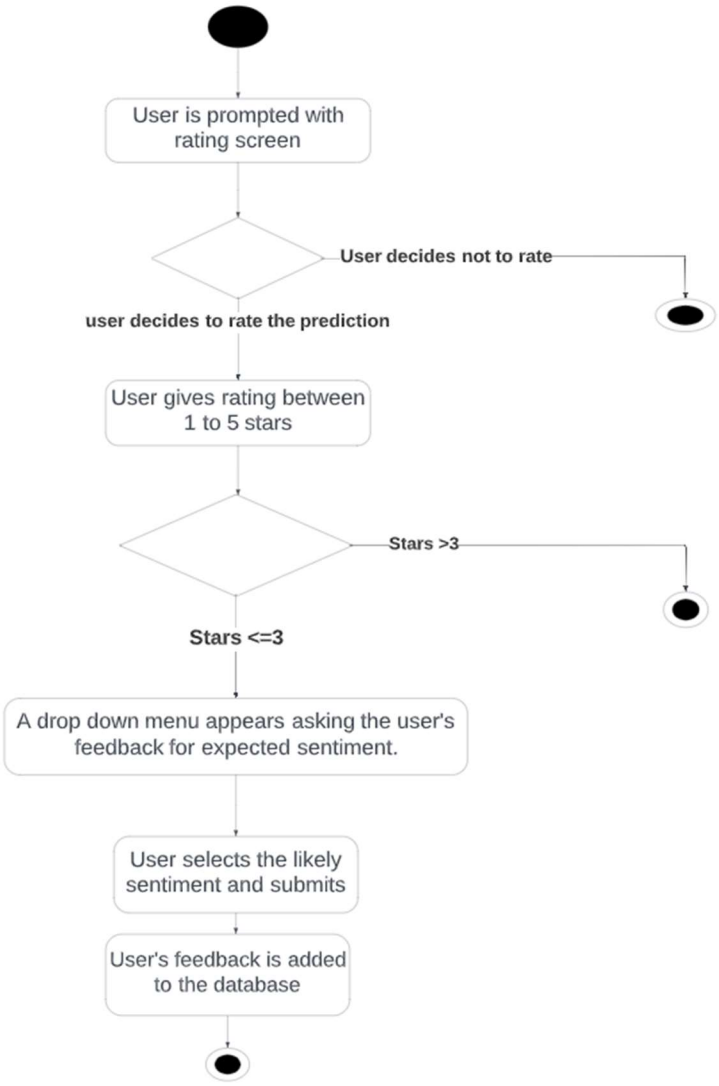
Scenario/ Description

1) Start
2) Server receive the preprocessed statement.
3) It checks the length of statement.

a) If length.text is less than 3 the combine result is calculated as avg = (svm_prob*76+mnb_prob*21)/(76+21).
b) Else avg = (svm_prob*5+mnb_prob)/6.

4) Display the output the the user.
5) End.

User Feedback Storing System



Scenario/Description

The activity diagram shows the working structure for the user feedback storing system of our very application. The feedbacks given by the users for their input statement are captured in

the database and utilized later while training the model to make our prediction even more accurate and practical.

The user feedback storing system is only available when the user is not fully satisfied with the prediction given by our app and decides to rate our prediction with stars less than or equal to 3. So this considering this very disappointment of user, we allow user to input the most practical and feasible sentiment for their input statement, which will be later stored in our database and utilized to make our model stronger.

In this way this portion of our system has been designed to assure more and more quality production as per the passage of time and usage. This enables our app to be more reliable and accurate as the users keep on using the app.

The feedback is stored in database whose ER-diagram is shown below.

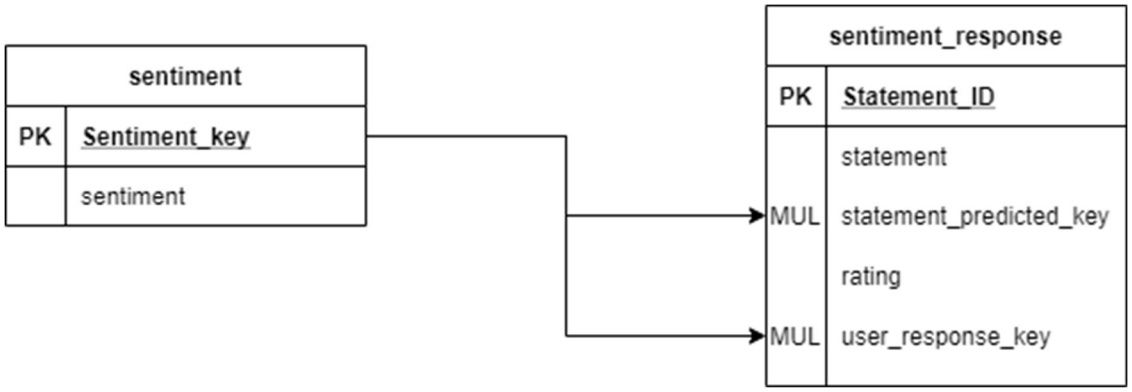


Figure 7: Feedback Storing Database, ERD

Similarly, there is another table in the database for storing admin information.

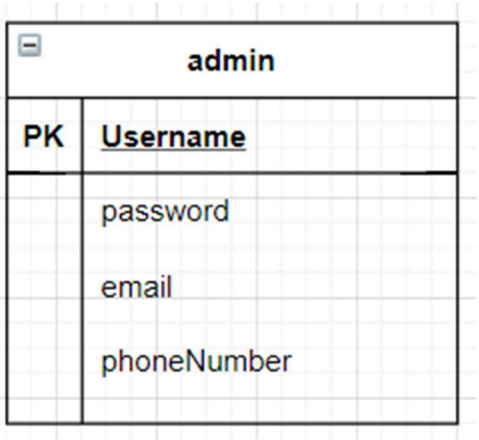


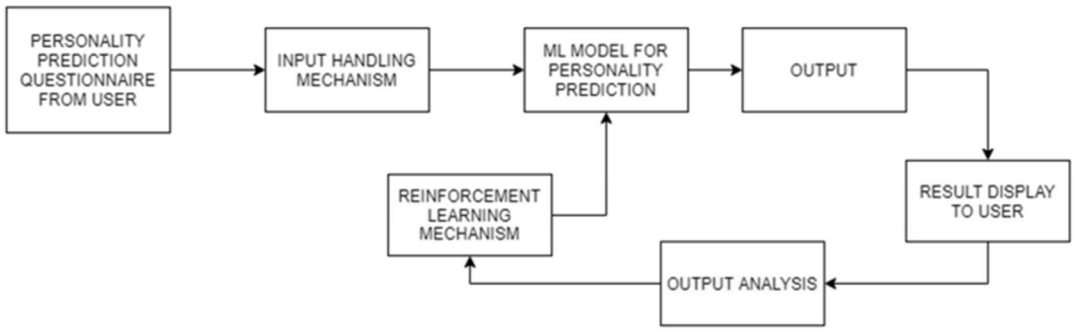
Figure 8: Admin DB table structure

Admin Privileges

There are four admins for the project, each group members who can log into the system and modify the sentiment analysis model and download the database content from the front end of the application.

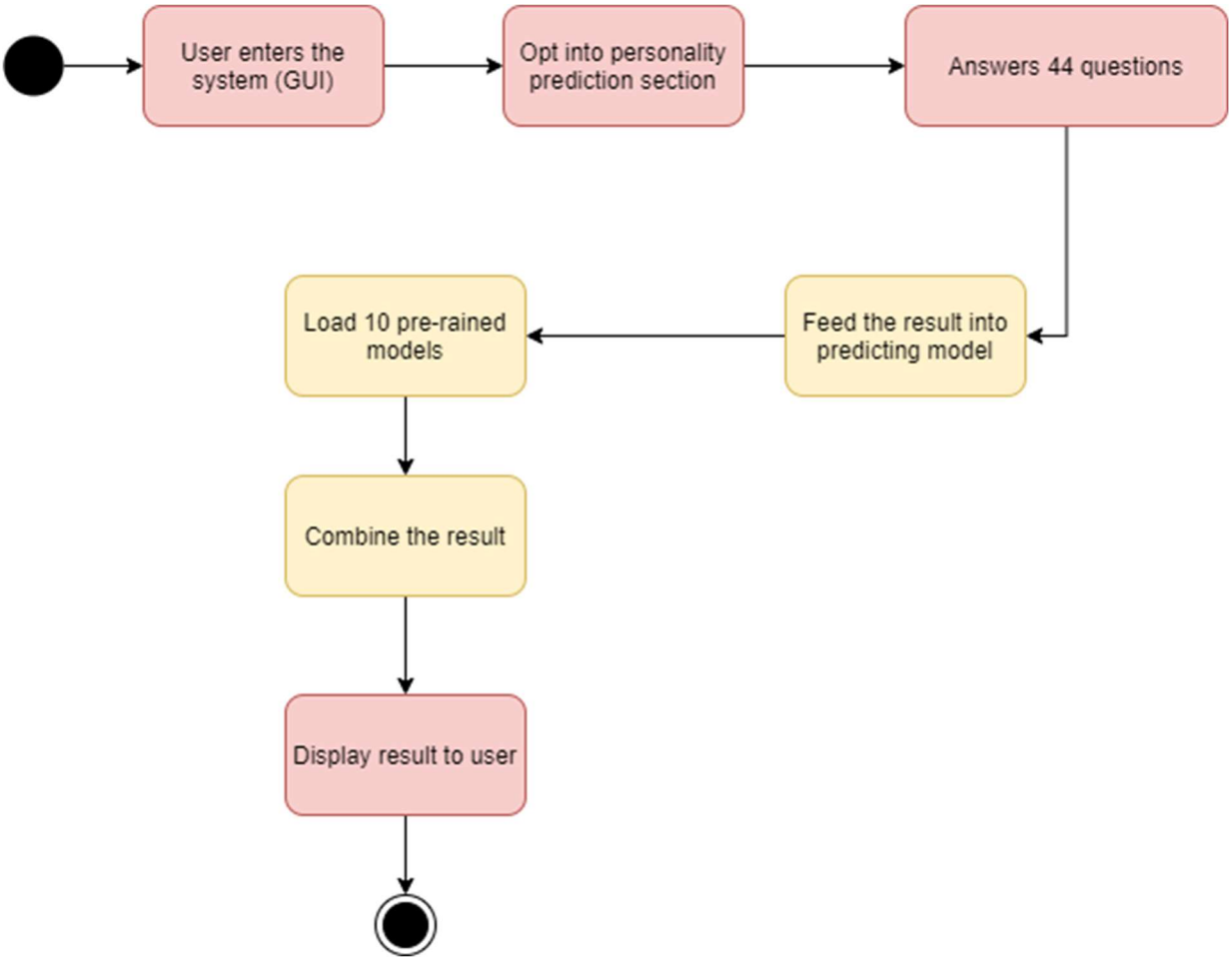
Personality Prediction

Our model predicts the personality of the user based on how he answers the 44 psychological questions.



BLOCK DIAGRAM FOR PERSONALITY PREDICTION

Main system



Overall Process:

- User gets into the system
- He opts into the personality prediction section
- He answers the 44 questions related to psychology
- The answers are then fed into the system
- The result is finally displayed to the user

Result and Analysis

Sentiment Analysis

1. Multinomial Naïve Bayes (NMB)
   - The results obtained from using NMB only are:

Scores	MNB
Accuracy	0.838666667
Balanced Accuracy	0.657740829
F1- Score	0.823646399
Precision	0.845989391

   - The obtained results were lower than that of SVM in every points.

2. Support Vector Machine
   - Kernel Used
     - We tested the accuracy of the model trained with linear, polynomial, RBF and sigmoid kernel and the results obtained is shown in graph.

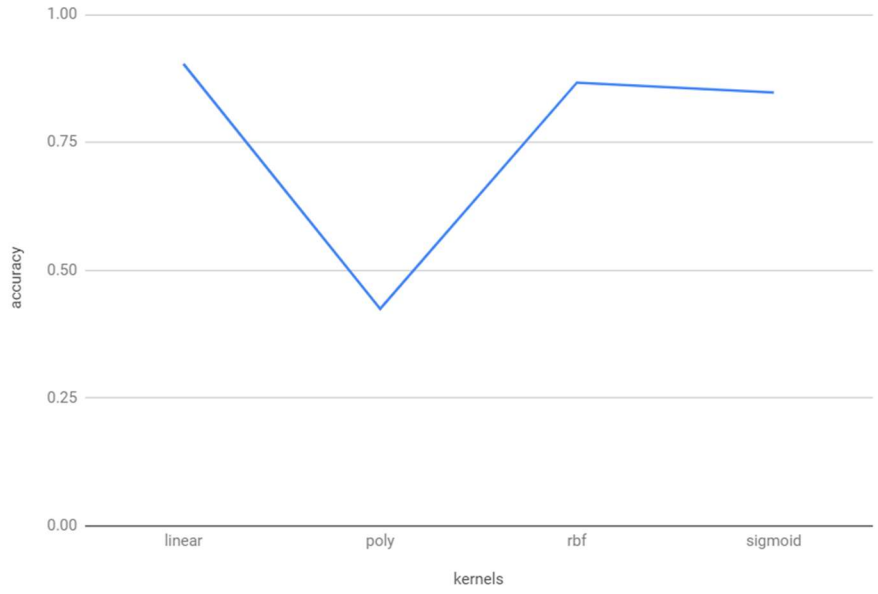


Figure 9: Accuracy vs Kernel used

   - We can see from the result that linear kernel is best for our model and hence linear kernel was used for Support Vector Machine classifier.

- The results obtained from using NMB only are:

Scores	SVM
Accuracy	0.901666667
Balanced Accuracy	0.85867667
F1- Score	0.902196636
Precision	0.902997113

3. Combination of result
   - We found that while combing the model in certain combination rather than taking average of them, the results were better.
   - We first, classified the statements into two groups with less than 3 words and other with more than 3 words.
   - The combination of result was done by testing out all the possible combinations from ratio 1 to 100.
   - The results obtained from the test are demonstrated with the graph shown below.

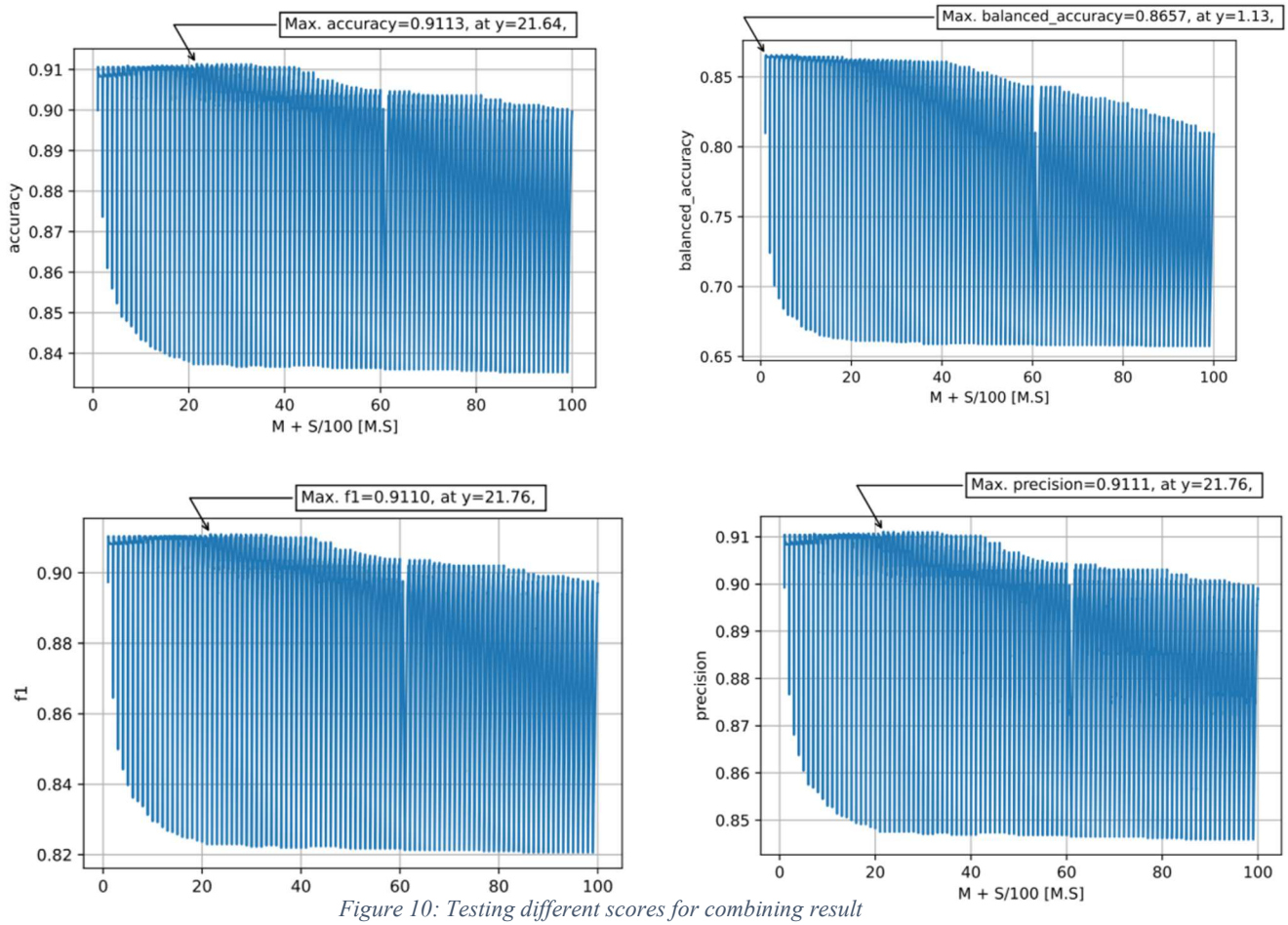


Figure 10: Testing different scores for combining result

- Hence, analyzing the result, MNB and SVM probabilities were combined with 21:76 ratio for short inputs and for longer input, the ratio was kept 1:6.
- Final result obtained are as follows:

Scores	MNB	SVM	Combined
Accuracy	0.838666667	0.901666667	0.905333333
Balanced Accuracy	0.657740829	0.85867667	0.859047472
F1- Score	0.823646399	0.902196636	0.903384676
Precision	0.845989391	0.902997113	0.903475342

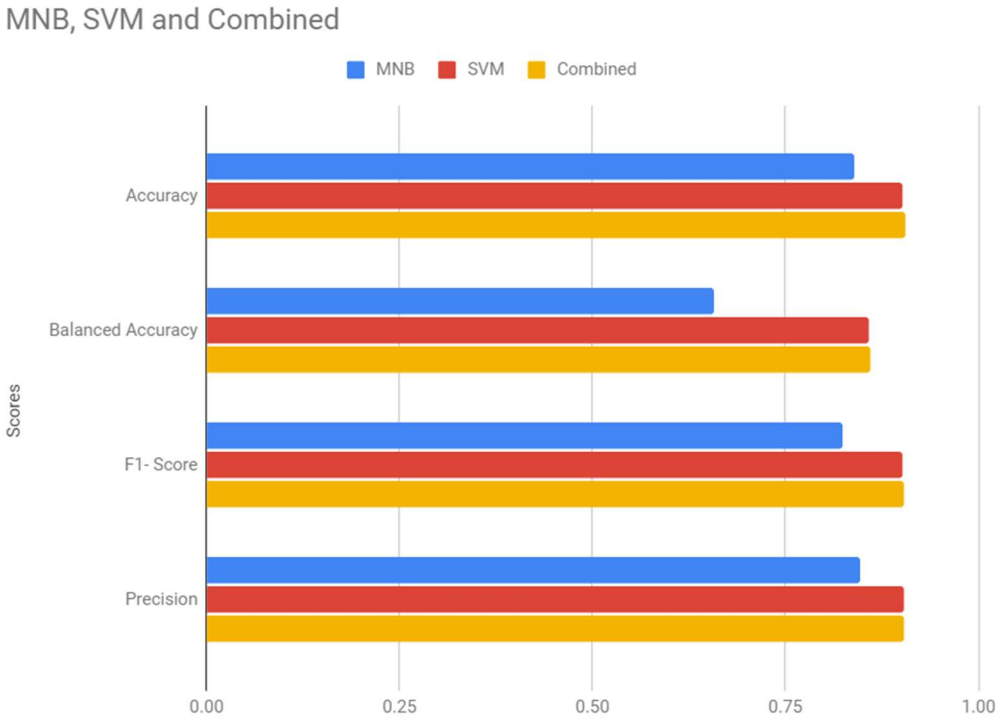


Figure 11: Overall Result for sentiment analysis

Personality Prediction.

The model was trained with *Random Forest Classifier (RFC).* The n-estimators was determined by analyzing the result with the N estimators ranging from 0 to 700, and the result obtained is illustrated by the graph below.

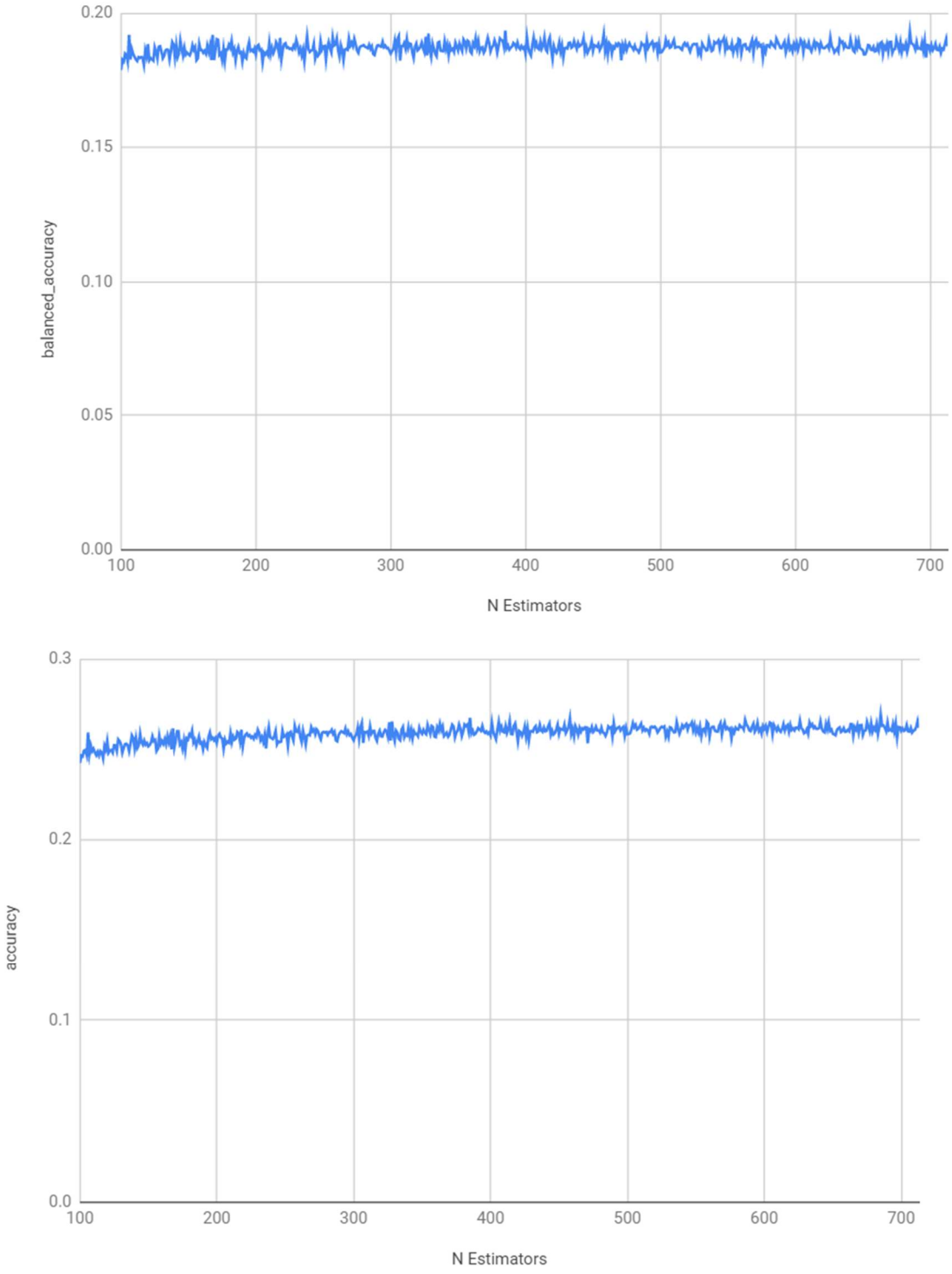


Figure 12: Graphs for optimizing the N estimators

- It was seen that for N-estimator = 385, overall score of the model was best, hence the RFC was trained for N-estimators = 385.

- The result obtained are:

Metrics	N Estimators 385
accuracy	0.2673974256
balanced_accuracy	0.1934210141
f1	0.2184797752
precision	0.2377734224

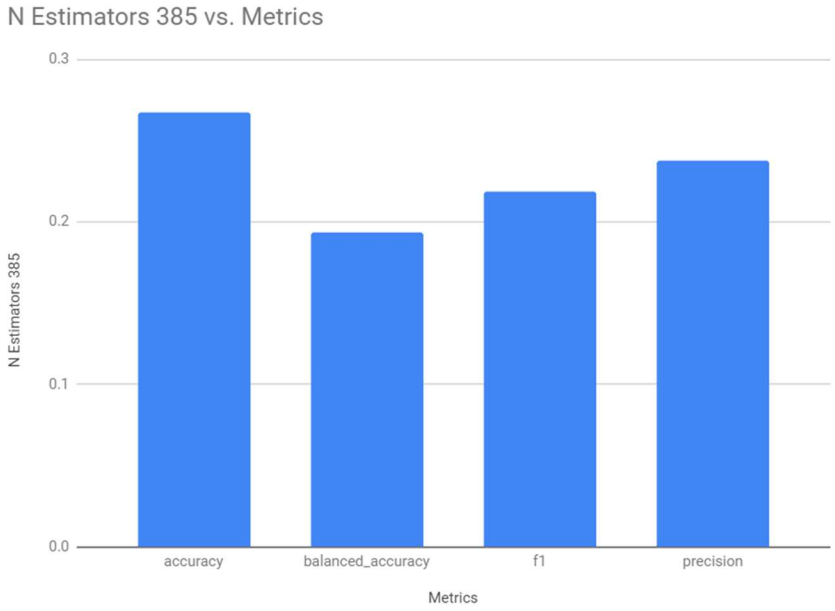
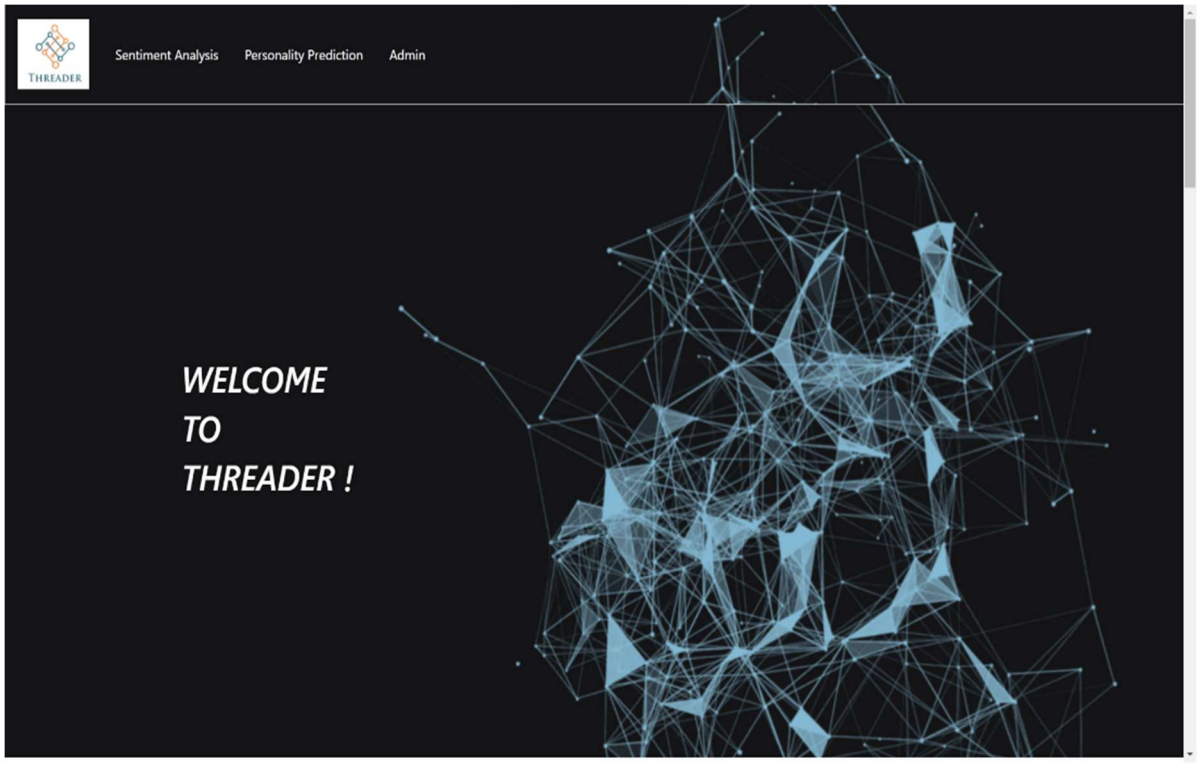


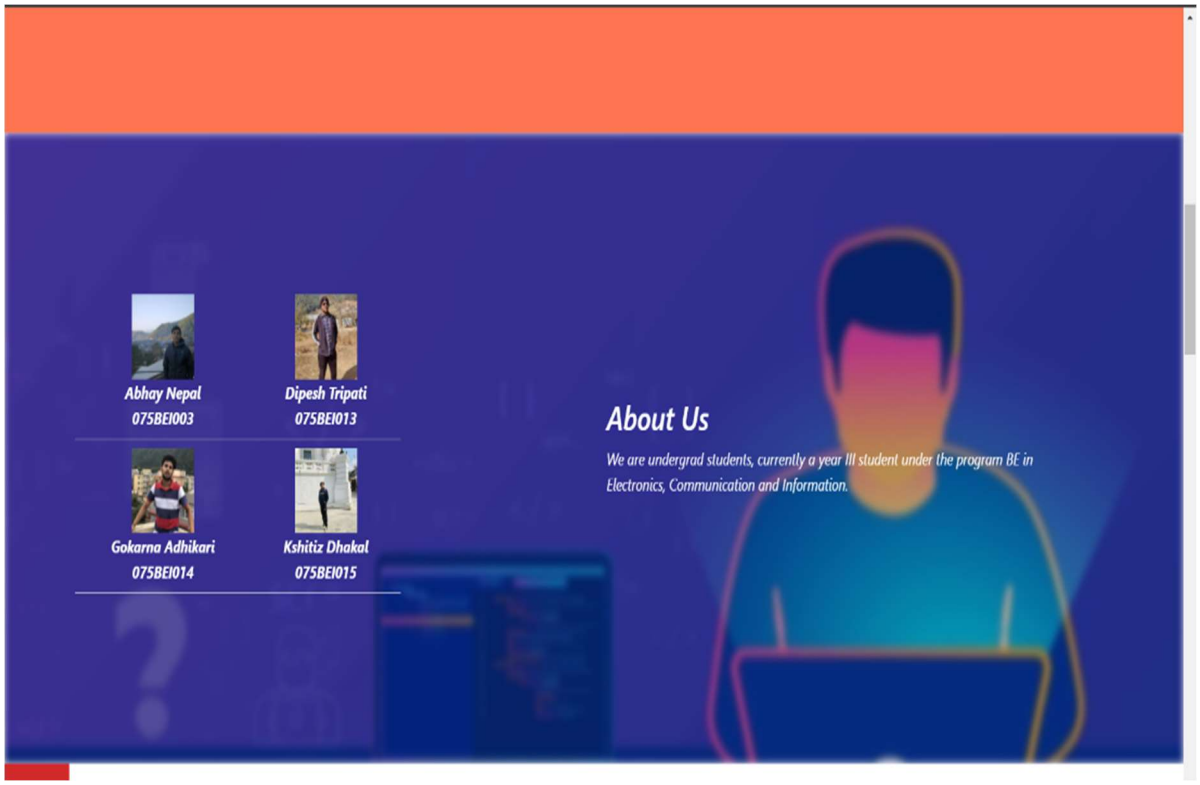
Figure 13: Result for personality prediction

Results

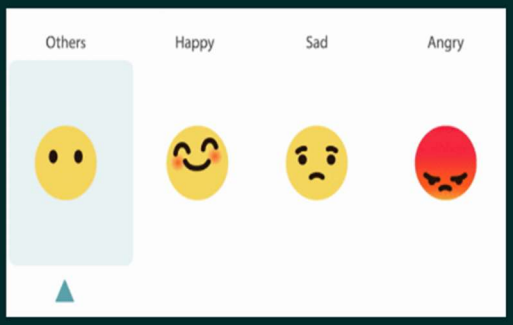
Graphical Interface

Home Screen





The Project

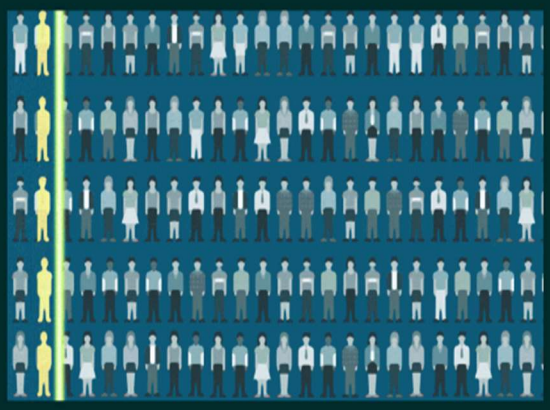


Sentiment Analysis

We can analyze the sentiment of a english text with more than 90% accuracy !

Personality Prediction

We can predict human personality that is true for Quarter of the world population based on more than 70 thousands data !

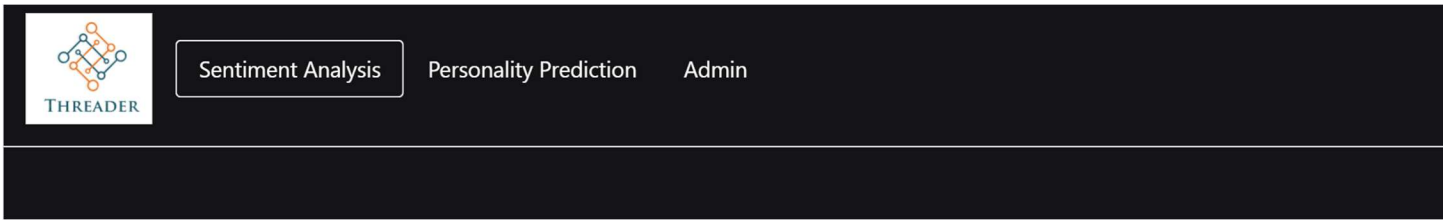


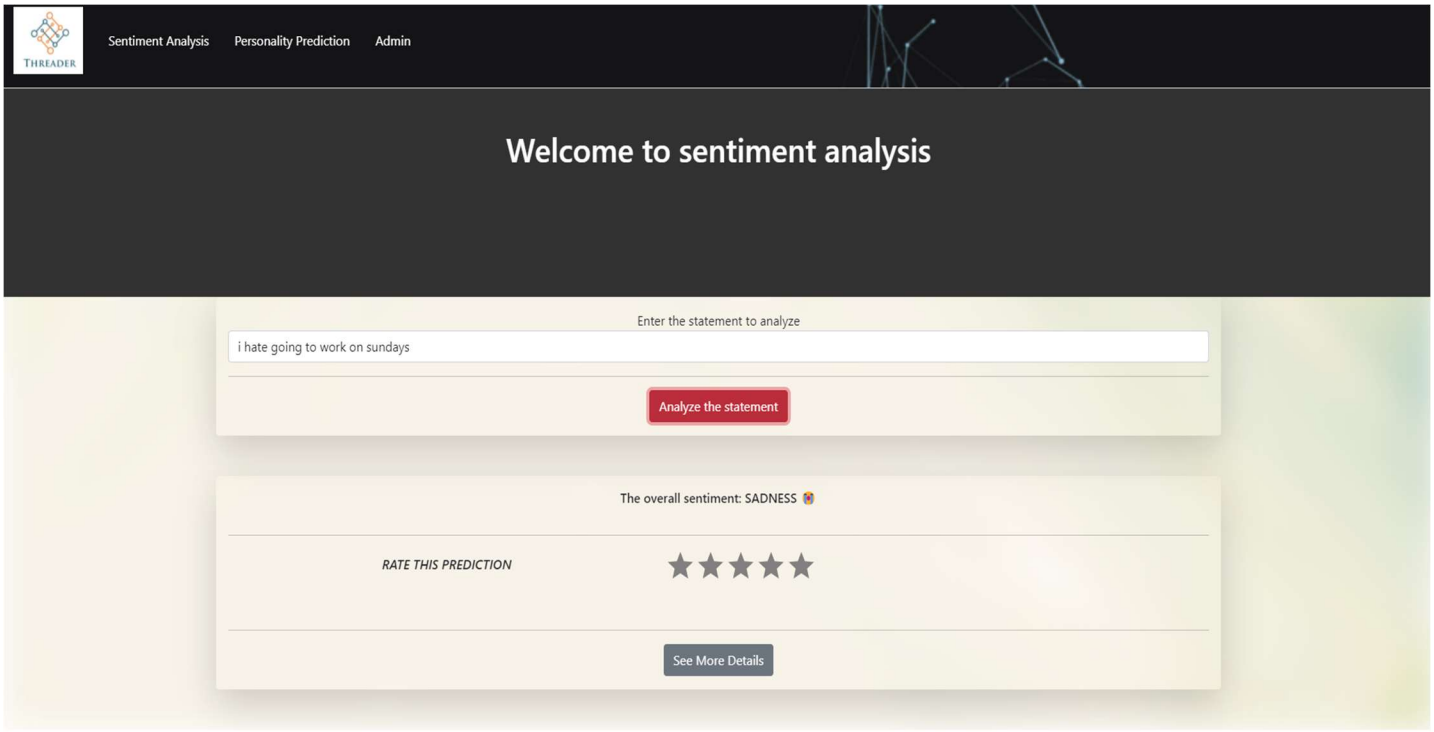
THREADER - A ML implied perceptive emotion analyzer!

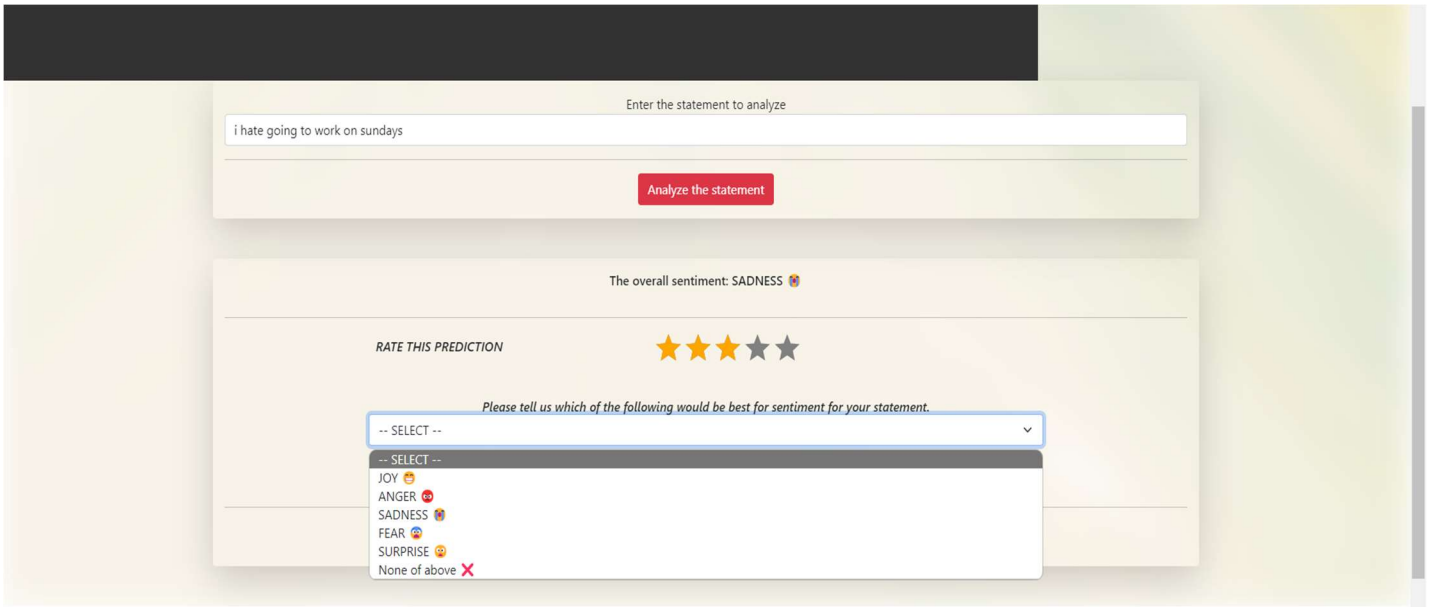
THREADER is a project completed by Abhay, Dipesh, Gokarna and Kshitiz as the minor project in the undergrad year iii part ii. This app can perform sentimental analysis of a text, predict your personality and generate your handwriting font !

THREADER

Sentiment Analysis







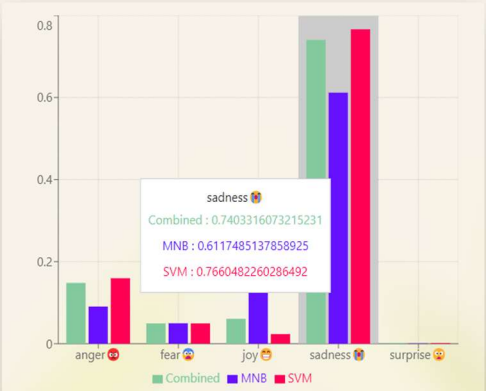
Hide More Details

Original Statement: *i hate going to work on sundays*
Statement fed into ML Model: *hate going work sundays*
Confidence: *0.7403316073215231*

Model/Sentiment	anger	fear	joy	sadness	surprise
SVM	0.159599	0.049446	0.023369	0.766048	0.001537
MNB	0.090593	0.049760	0.247050	0.611749	0.000849
AVG	0.148098	0.049498	0.060649	0.740332	0.001422

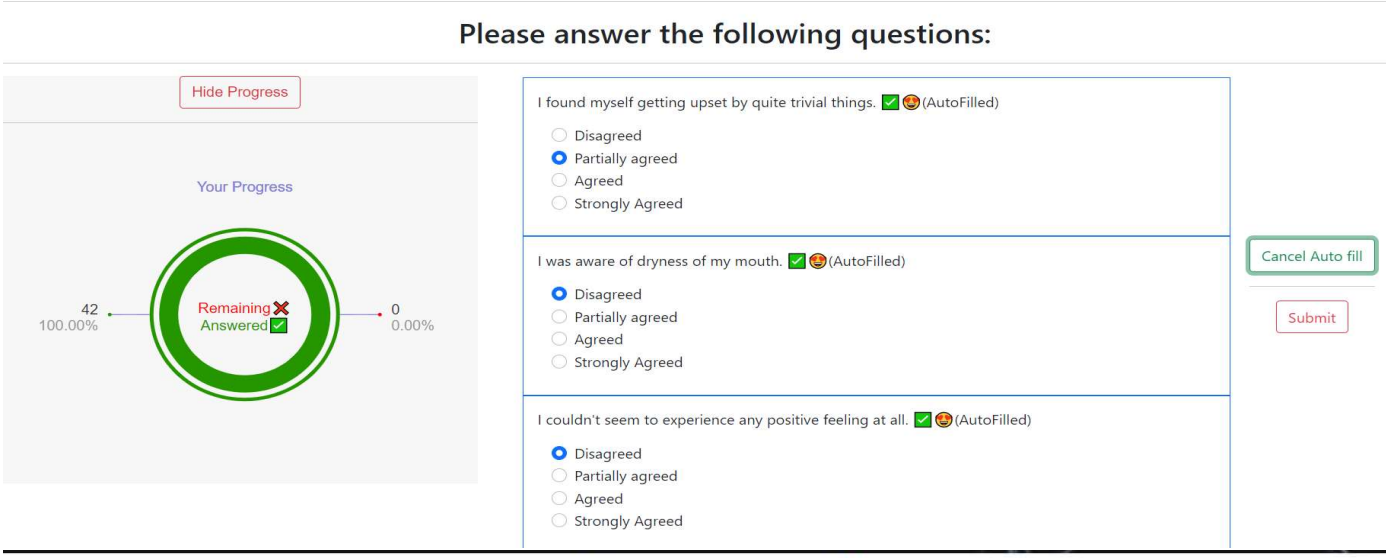
Visualize this data

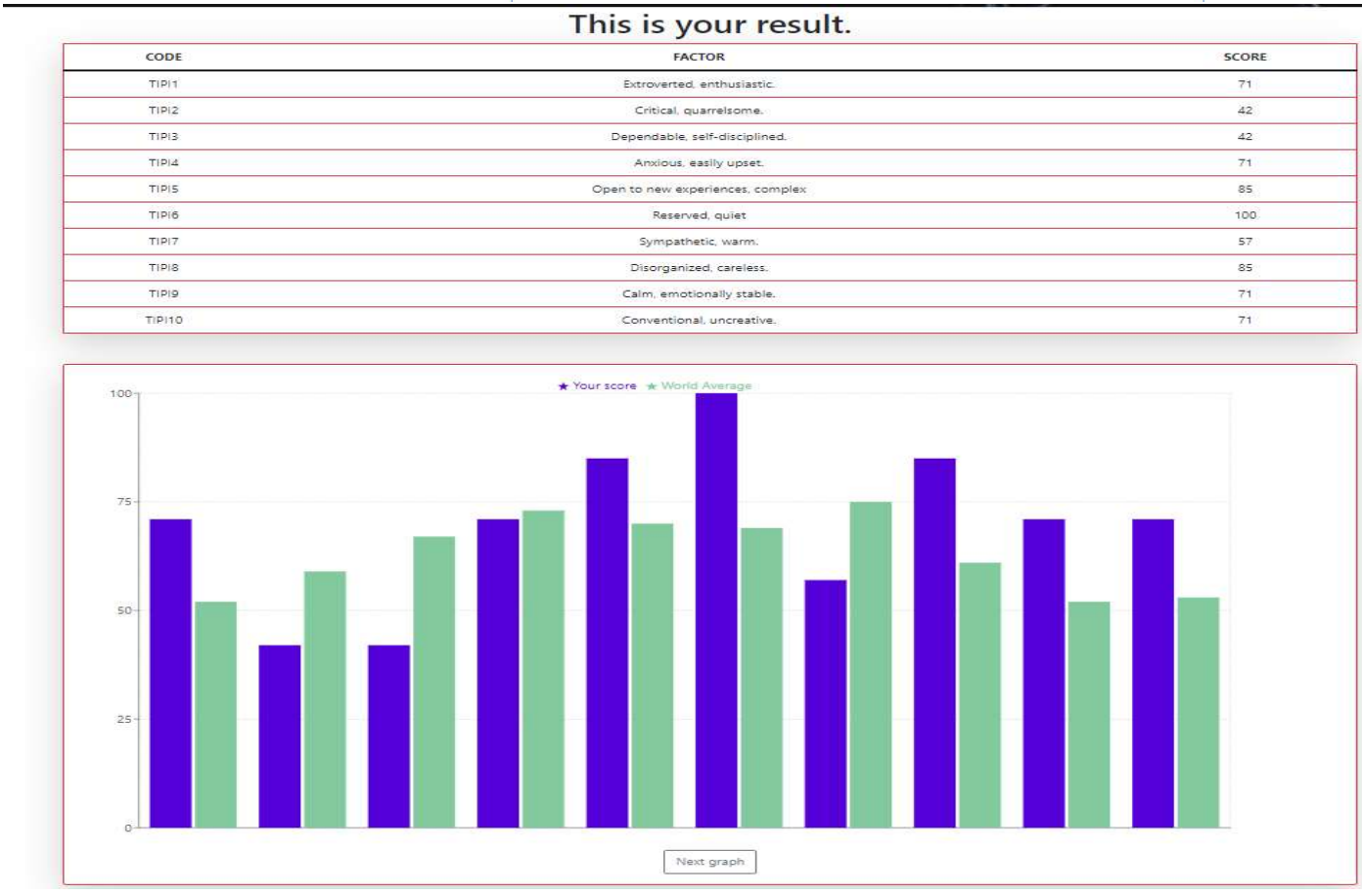
Visualize this data



Hide individual Model result

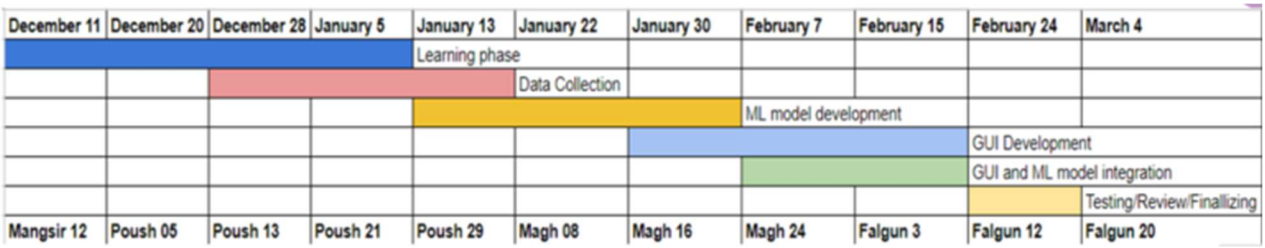
Personality Prediction





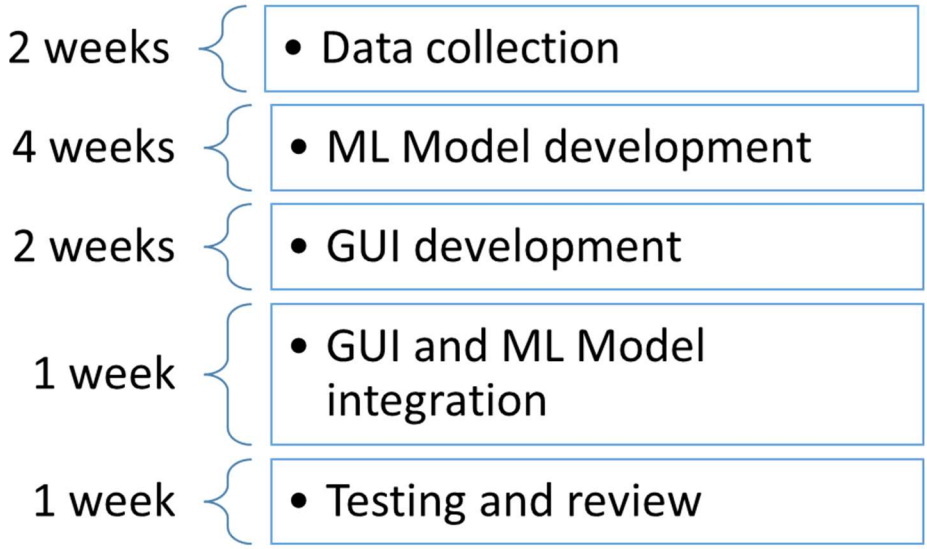
CODE	FACTOR	SCORE
TIPI1	Extroverted, enthusiastic.	71
TIPI2	Critical, quarrelsome.	42
TIPI3	Dependable, self-disciplined.	42
TIPI4	Anxious, easily upset.	71
TIPI5	Open to new experiences, complex	85
TIPI6	Reserved, quiet	100
TIPI7	Sympathetic, warm.	57
TIPI8	Disorganized, careless.	85
TIPI9	Calm, emotionally stable.	71
TIPI10	Conventional, uncreative.	71

Project Schedule



Total: 80 Days

Project Timeline



Project Budget

Since we will be using all open-source software and libraries, the budget of our project will not be much; it is expected to be less than Rs. 10,000 excluding the computers/laptop we are going to use unless we plan to incorporate it in cloud-based web application in future.

Particulars	Description	Cost
Computer/Laptop	Hardware setup for programming	Rs. 150,000
Visualization tools, APIs		Rs. 5,000
Reports and documentation		Rs. 3,000
	Grand Total	Rs. 158,000

Conclusion:

The task of sentiment analysis, in the generic domain , is still in the developing stage and far from completion of the final expected . We propose a couple of ideas which we feel are worth exploring in the future and may result in further improved performance.

For the scope of this minor project, 'Threader' has been completed to a stage where it can predict 'sentiments of statements', 'Personality based on questionnaire' in English language with the desired output and satisfactory level of accuracy. We were able to obtain about 91% accurate output by combining the Support Vector Machine and *Multinomial Naive Bayes* algorithm in our project. Our Personality Prediction part was done by using the concept of Random Forest Classifier of Sklearn library. Our project is implemented as an Web application using NodeJS.

Finally, we would like to thank our department and our respected teachers to give us this opportunity to work on the minor project which helps us to understand the experience of working together and to gather the knowledge of our scope which can be applied through our studies.

Future enhancement

The possible future enhancements of our project are listed below:

1. We can add Nepali Language to our Sentimental Analysis.
2. We can expand the datasets for a greater number of sentiments.
3. The UI of our web application can be made more user friendly and interactive.
4. The feedback obtained can be used for new training datasets.
5. The proposed approach is currently incapable of interpreting sarcasm. The main goal of this approach is to empirically identify lexical and pragmatic factors that distinguish sarcastic, positive and negative usage of words.
6. Analyzing sentiments on emojis.
7. Potential improvement can be made to our data collection and analysis for personality prediction model.
8. To recognize the Neutral statements.

References

1. https://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-python
2. scikit-learn Tutorials — scikit-learn 1.0.2 documentation
3. Tutorial: Intro to React – React (reactjs.org)
4. *https://monkeylearn.com/blog/sentiment-analysis-applications/*
5. *Research papers:*
    a. *https://www.cs.cornell.edu/home/llee/papers/cutsent.pdf*
    b. *http://www.ijstr.org/final-print/apr2020/Literature-Review-On-Sentiment-Analysis.pdf*
    c. *https://ieeexplore.ieee.org/document/6897213*
    d. *E. Kouloumpis, T. Wilson, J. Moore, Twitter sentiment analysis: The good the bad and the omg! , Proc. 5th Int. AAAI Conf. Weblogs Social Media, pp. 538-541, 2011.*
    e. *H. Saif, Y. He, H. Alani, Alleviating data sparsity for Twitter sentiment analysis, Proc. CEUR Workshop, pp. 2-9, Sep. 2012*
    f. *Nupur Kalra, Deepak Yadav, Gourav Bathla. (2019), SynRec: A prediction technique using collaborative filtering and Synergy Score*