# Identifying Critical Quality Attributes Of Personality Trait Of An Individual From His/Her Multi-Modal Social Media Activities

*Submitted in partial fulfillment of the requirements for the degree*

*of*

**Bachelor of Technology**

*by*

**Sanjeev Pratap (13000120063)**
**Sudakshina Majumdar (13000120078)**
**Suman Raj (13000120080)**
**Sumana Sen (13000120081)**


DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TECHNO MAIN SALT LAKE
EM 4/1, SALT LAKE, SECTOR - V, KOLKATA - 700091

# CERTIFICATE

This is to certify that the project entitled "Identifying Critical Quality Attributes Of Personality Trait Of An Individual From His/Her Multi-Modal Social Media Activities" prepared by Sanjeev Pratap (13000120063), Sudakshina Majumdar (13000120078), Suman Raj(13000120080) and Sumana Sen(13000120081) of B.Tech (Computer Science & Engineering), Final Year, has been done according to the regulations of the Degree of Bachelor of Technology in Computer Science & Engineering. The candidates have fulfilled the requirements for the submission of the project report.

It is to be understood that, the undersigned does not necessarily endorse any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

| | |
|---|---|
| (Signature of the Internal Guide) | (Signature of the HOD) |
| (Signature of External Guide, if applicable) | (Signature of the External Examiner) |

.....

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

.....

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project guide in Computer Science and Engineering department. We are extremely thankful for the keen interest he / she took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staffs for the gracious hospitality they offered us.

.....
**Sanjeev Pratap (13000120063)**

.....
**Sudakshina Majumdar (13000120078)**

.....
**Suman Raj (13000120080)**

.....
**Sumana Sen (13000120081)**

Techno Main Salt Lake
Date: 28-05-2024

Identifying Critical Quality Attributes Of Personality Trait Of An Individual From His/Her Multi-Modal Social Media Activities

# Contents

Identifying Critical Quality Attributes Of Personality Trait Of An Individual From His/Her Multi-Modal Social Media Activities

# Abstract

Social networking sites play an important role in our everyday lives and users are now free to choose from various social networking platforms. In 2017, more than half of the globe uses the world wide web network with 2.7 billion live social media accounts world widely Praphulla et al. (2023) and every user of those social platforms leaves a mark as digital footprint and tend to present themselves in a type of behavior usually determined by their personality traits. Personality traits analysis involves assessing a person's individual characteristics and patterns of thought, behavior, and emotion. There is need to design a system capable of analysing users' activities on social media platforms to predict their personality traits accurately. Understanding these traits can offer insights into cognitive abilities, which are critical for tasks ranging from problem-solving to decision-making.

Problem Statement:
Gain a deeper understanding of how individuals showcase their personality traits through social media interactions and understand how these affect cognitive abilities, which are critical for tasks ranging from problem-solving to decision-making.

Solution:
This project incorporates diverse social media data like texts (posts, comments) and images (profile pictures) from the users' social media. An initial personality trait analysis is conducted through a self-reported 50-item IPIP questionnaire for Big Five Model. The personality trait scores obtained for each person and their extracted social media textual data like posts and comments are used to train the Text-based model. Textual data from social media platforms like Facebook and LinkedIn are used for analysis to improve the accuracy of the Text-based model. Then further analysis is done using the image-based model trained through people's profile images extracted from their social media handles and their personality trait scores. An initial cognitive skill assessment is done through a self-reported questionnaire. The obtained cognitive scores and the predicted personality trait scores is used to train a ML model. This model is then used to predict the associated critical qualities (cognitive trait scores) of an individual when provided with his/her social media credentials.

Expected Results:

1. Development of a robust and reliable ML models: The developed models are capable of predicting personality traits scores and cognitive scores with high accuracy.

2. Insights into Personality and Cognitive Scores: Identification of correlations between personality traits scores and cognitive scores and understanding which personality traits have the most significant impact on cognitive scores.

Project Benefits:
Improved personal growth and development, improved organizational productivity, enhanced mental health assessments, improved recruitment and talent management, customized learning and student support.

# 1 Introduction

The project provides a comprehensive, data-driven assessment tool to understanding personality traits and cognitive abilities of an individual utilizing behavioural observations from social media platforms. Applying machine learning to analyse the extensive curated dataset from traditional psychological assessment methods like Big 5 questionnaire and cognitive assessments methods, this project offers an integrated tool to scrutinize the differences between predicted and self-reported personality traits, thus providing a deeper, more accurate view of individual's personalities. The relationship between the predicted personality trait scores and cognitive scores is also studied. This will contribute to advancements in psychological research and practical applications in various fields such as human resources, mental health, recruitment, education and social sciences.
The purpose of this project is to develop a predictive model that estimates individuals' cognitive scores based on personality traits inferred from their multi-modal social media activities like posts, comments and profile pictures.

## 1.1 Technical Domain

Hardware Requirements

- CPU
- RAM 2 GB
- HDD 200 GB
- Network adapter
- UPS

Operating System Requirements

- Windows OS – latest version
- Any Linux OS – latest version
- MAC OS

Software Requirements

- Web Scraping: BeautifulSoup, requests
- Text Processing: NLTK, deep-translator googletranslator
- Image Processing: OpenCV
- Machine Learning: Keras, Scikit-learn, numpy, pandas
- Deep Learning Framework: TensorFlow, VS Code
- Version Control: Git and GitHub

## 1.2 Glossary

| Sl No. | Term | Description |
|---|---|---|
| 1 | Social Networking Sites | Online platforms that allow users to create profiles, connect with others, and share content and interactions. |
| 2 | Digital Footprint | The trail of data and information left by a person's online activities, including social media interactions. |
| 3 | Personality Traits | Unique psychological characteristics, behaviors, and tendencies that define an individual's personality. |
| 4 | Linguistic Features | Characteristics related to language and text, such as sentiment analysis, word frequency, and topic modeling. |
| 5 | Mean Squared Error (MSE) | A metric used to measure the average squared difference between predicted and actual values in regression tasks. |
| 6 | Real-time Analysis | Continuous monitoring and analysis of data to provide insights and adapt to changing behaviors and circumstances. |

Table 1: Glossary

# 2 Related Studies

The prevalence of social platforms triggered multiple research endeavors in human personality estimation and prediction. Through social media, people can easily express themselves to the whole world, which is impossible physically Liu et al. (2016) . They enjoy the circumstances in which they can interact with other people without meeting the person physically Azucar et al. (2018) , i.e., without seeing them, or specifically completely obscuring their feelings. It is convenient for them to hide their true emotions, and make a statement to the outside worldSorokowska et al. (2016). It is usually observed that the posts shared by users on social media have a direct correlation with their ideologies and personalities Golbeck et al. (2011) . Statistically, it is observed that women are more prone to post selfies, group pictures, and other media than men Guntuku et al. (2017). Among women, depending upon their social circles, the posts may differ quite a bit. For example, if someone has a bigger social circle and is outspoken, they usually post about various events in their lives, most of which are various social interactions with her circle Sorokowska et al.

(2016). Whereas someone with limited social interaction does not have a lot to post about his/her daily life Walsh et al. (2013) Seidman (2013). In a similar context, someone who is shy is not usually observed to be very active in social media platforms, Kavakci & Kraeplin (2017)Park et al. (2015) whereas someone who is bold is usually much more active. Based on this trend, it can be predicted that the person is introverted or extroverted depending on their posts and their activity on social media platforms Jin et al. (2019) which include how often they are online, or, frequency at which they post, etc. Pratama & Sarno (2015). In this part, we summarize recent research efforts in predicting individuals' personality from appearance. Liu et al. presented a large-scale analysis of profile images and personality at Twitter microblogging platform. They used a wider range of interpretable aesthetic and facial features to capture correlations with the personality that is in line and complement psychological research. The superior outcomes for every personality trait are achieved by blending engineered features derived from jointly social networking sites [Li et al. (2017)]. Cristani et al. investigated the rising size of multimedia information users generate and engage online and consider it as a probable contributing factor to our what so-called online appearance. The paper also confers that visual patterns correlates with personality score and can be used to predict personality where also they found that the favorite images users assign in his/her profile can be used eventually to build prediction models to estimate their preserved online personality. In 2013, Wan et al. Wan et al. (2014) tried to predict the Big Five personality types of users from the texts they collected from a Chinese social networking site, Weibo, using machine learning techniques. They were successful in their work. In another study, Li, Wan, and Wang Valanarasu (2021)used three different models namely the multi-tasking model, the multi-regression model, and the grey prediction model to predict the personality of individuals based on the Big Five personality model. They concluded that the grey prediction model is more efficient than the other two models. The degree to which personality and cognitive ability are related is a question that has generated intensive research and intense debate. Some authors have concluded that intelligence test performance may be influenced by some non-ability traits but that intelligence and personality are two independent constructs (Zeidner  Matthews, 2000). Others (e.g., Ackerman, 1996) have argued that personality traits play a significant role in the development of intellectual skills.

# 3 Problem Definition and Preliminaries

## 3.1 Scope

- **Human Resources and Recruitment:**
  Enhance talent acquisition, employee development, and succession planning by predicting cognitive and personality traits. Identify candidates with the best fit for specific roles, tailor training programs to optimize employee growth and performance, pinpoint high-potential employees for leadership roles, all based on individual cognitive strengths and personality profiles.

- **Marketing and Customer Insights:**
  Enhance targeted advertising, customer segmentation, and product development by understanding customers' cognitive traits and personality profiles. Develop personalized marketing strategies, create precise customer segments, and design products and services that resonate with customers' cognitive preferences and personality characteristics, improving engagement and conversion rates.

- **Education and E-Learning:**
  Enhance personalized learning and student counselling by customizing educational content, teaching methods, and guidance based on students' cognitive traits and personality profiles, improving learning outcomes and career guidance.

- **Healthcare and Well-being:**
  Enhance mental health care and patient engagement by predicting cognitive and personality traits to offer personalized care plans, interventions, and communication strategies tailored to individual profiles.

- **Financial Services:**
  Enhance risk assessment and customer service by using cognitive and personality traits to assess risk tolerance, predict financial behavior, and understand customer needs and preferences, thereby providing more personalized financial advice and support.

- **Retail and E-commerce:**
  Enhance the personalized shopping experience and customer retention by offering product recommendations and shopping experiences tailored to individual

cognitive and personality profiles, and developing loyalty programs that resonate with customers' cognitive traits and personality characteristics.

- **Gaming and Entertainment:**
  Enhance game design and content recommendation by creating engaging gaming experiences and suggesting personalized streaming content based on players' and viewers' cognitive traits and personality profiles.

- **Professional Development and Coaching:**
  Enhance executive coaching and career planning by providing personalized guidance and support based on insights into individuals' cognitive and personality traits, assisting executives and professionals in making informed decisions and achieving their career goals.

## 3.2 Exclusions

- Private Data: The project will not access or analyze any private or restricted data from social media platforms. Only publicly available data will be considered.

- Real-time Analysis: The project will not focus on real-time personality analysis or adaptability to changing behaviors on social media. It will primarily analyze historical data.

- In-depth Sentiment Analysis: While linguistic features will be considered, in-depth sentiment analysis, including sentiment towards specific topics or entities, is outside the scope of this project.

- Ethical and Privacy Considerations: The project will not address the ethical and privacy implications of personality analysis, such as the potential for bias or invasion of privacy. These considerations should be addressed separately.

## 3.3 Assumptions

Many factors are taken into account in the project assumption. The list below indicates the major milestones that have been scheduled:

- The project deadline of 4th June, 2024.
- The final presentation is on 3rd June, 2024.
- The peer evaluation deadline is on 1st June, 2024.
- The project assumes that the results of personality trait predictions can be interpreted and translated into actionable insights, even though the complexity of human personality may limit the precision of such predictions.
- The project assumes that machine learning models can achieve a reasonable level of accuracy in predicting personality traits based on social media activities.

NOTE: Since the deadline of project is 4th June, 2024, running out of time will have its reflection on the product, and submission schedule can't be revised. All possible measures will be taken to finish the project as per schedule.

# 4 Proposed Solution

Explain your proposed work in details. Clearly state your specific contributions and reusable components deployed in the project.

1. Comprehensive Integration of Social Media Data:
    - Aim to surpass traditional text-based methods.
    - Utilize diverse social media data including texts (posts, comments) and images (profile pictures) for personality insights and cognitive score assessment.
2. Initial Personality Trait Analysis and Text-based Model Training:
    - Conduct a self-reported 50-item IPIP questionnaire based on the Big Five Factor Markers.
    - Use personality trait scores and social media textual data for training the Text-based model.
    - Employ data from multiple platforms like Facebook and LinkedIn to enhance model accuracy.

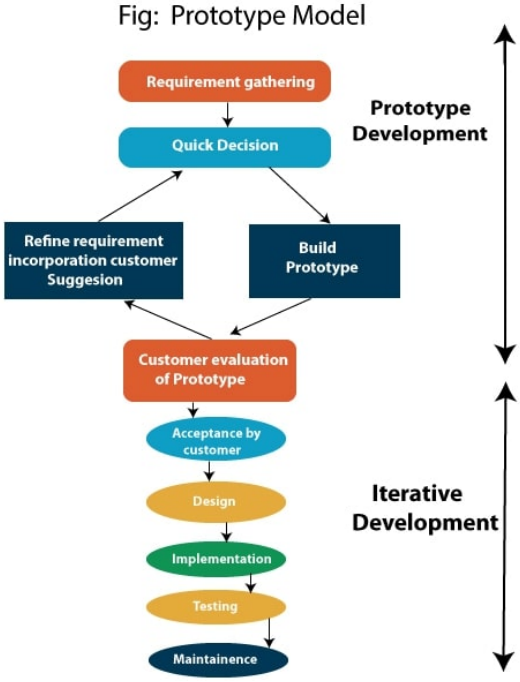
3. Image-based Model Analysis:
    - Utilize profile and post images extracted from social media handles.
    - Perform further analysis using the image-based model alongside personality trait scores.
4. Cognitive Skill Assessment and ML Model Training:
    - Conduct initial cognitive skill assessment through a self-reported questionnaire.
    - Utilize obtained cognitive scores and predicted personality trait scores to train a machine learning model.
5. Prediction of Critical Qualities:
    - Utilize the trained model to predict associated critical qualities (cognitive scores) when provided with an individual's social media credentials.

# 5 Project Planning

## 5.1 Software Life Cycle Model

The prototype model will be used for this project. This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested, and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

Fig: Prototype Model




Steps of Prototyping Model:

- Step 1: Requirement Gathering and Analysis: This is the initial step in designing a prototype model. In this phase, users are asked about what they expect or what they want from the system.
- Step 2: Quick Design: This is the second step in the Prototyping Model. This model covers the basic design of the requirement through which a quick overview can be easily described.
- Step 3: Build a Prototype: This step helps in building an actual prototype from the knowledge gained from prototype design.
- Step 4: Initial User Evaluation: This step describes the preliminary testing where the investigation of the performance model occurs, as the customer will tell the strengths and weaknesses of the design, which was sent to the developer.
- Step 5: Refining Prototype: If any feedback is given by the user, then improving the client's response to feedback and suggestions, the final system is approved.

- Step 6: Implement Product and Maintain: This is the final step in the phase of the Prototyping Model where the final system is tested and distributed to production, here program is run regularly to prevent failures.

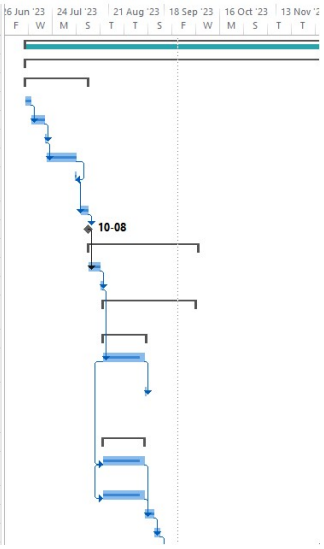
This model is being used as:

- Requirements of the client are not clear and can be subject to changes.
- Modifications can be done with ease.
- Improvement of quality does not increase the cost of the product.
- Errors can be detected much earlier with the help of a prototype.
- Continuous user feedback ensures the proper and intended functioning of the prototype.

## 5.2 Scheduling

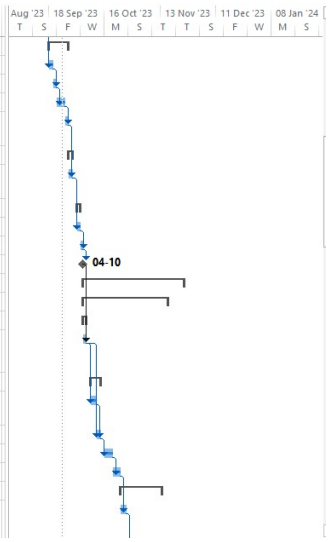
GANTT CHART

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | % Complete |
|---|---|---|---|---|---|---|---|
| 1 | | **CSE Academic Project** | **239 days** | **Mon 10-07-23** | **Thu 06-06-24** | | **84%** |
| 2 | | **Phase 1: 7th Semester Activities** | **108 days** | **Mon 10-07-23** | **Wed 06-12-23** | | **95%** |
| 3 | ✓ | **Project Startup** | **24 days** | **Mon 10-07-23** | **Thu 10-08-23** | | **100%** |
| 4 | ✓ | Team Building | 3 days | Mon 10-07-23 | Wed 12-07-23 | | 100% |
| 5 | ✓ | Brainstorm on Project Topic | 5 days | Thu 13-07-23 | Wed 19-07-23 | 4 | 100% |
| 6 | ✓ | Project agreed with Guide | 1 day | Thu 20-07-23 | Thu 20-07-23 | 5 | 100% |
| 7 | ✓ | Related Study & Documentation | 11 days | Fri 21-07-23 | Fri 04-08-23 | 6 | 100% |
| 8 | ✓ | Deliver Project Synopsis for Guide's review | 1 day | Fri 04-08-23 | Fri 04-08-23 | 7FF | 100% |
| 9 | ✓ | Close review feedbacks | 4 days | Mon 07-08-23 | Thu 10-08-23 | 8 | 100% |
| 10 | ✓ | **Project Synopsis Finalized** | 0 days | Thu 10-08-23 | Thu 10-08-23 | 9 | 100% |
| 11 | | **Requirement Analysis** | **39 days** | **Fri 11-08-23** | **Wed 04-10-23** | | **90%** |
| 12 | ✓ | Gather Requirements | 4 days | Fri 11-08-23 | Wed 16-08-23 | 10 | 100% |
| 13 | ✓ | Prepare Draft Requirement Matrix | 1 day | Thu 17-08-23 | Thu 17-08-23 | 12 | 100% |
| 14 | | **Elaborate Requirement & Documentation** | **33 days** | **Fri 18-08-23** | **Tue 03-10-23** | | **89%** |
| 15 | | **Big 5 Questionnaire** | **16 days** | **Fri 18-08-23** | **Fri 08-09-23** | | **85%** |
| 16 | | Collecting responses from Big 5 Questionnaire | 15 days | Fri 18-08-23 | Thu 07-09-23 | 13 | 85% |
| 17 | | Calculating personality trait scores from Big 5 Questionnaire results | 1 day | Fri 08-09-23 | Fri 08-09-23 | 16 | 85% |
| 18 | | **Data Collection** | **15 days** | **Fri 18-08-23** | **Thu 07-09-23** | | **85%** |
| 19 | | Collect credentials required for social media | 15 days | Fri 18-08-23 | Thu 07-09-23 | 16SS | 85% |
| 20 | | Scrap textual data and images | 15 days | Fri 18-08-23 | Thu 07-09-23 | 19SS | 85% |
| 21 | ✓ | **Data Preprocessing** | **3 days** | Fri 08-09-23 | Tue 12-09-23 | 20,19 | 100% |
| 22 | ✓ | **Feature Extraction** | **3 days** | Wed 13-09-23 | Fri 15-09-23 | 21 | 100% |



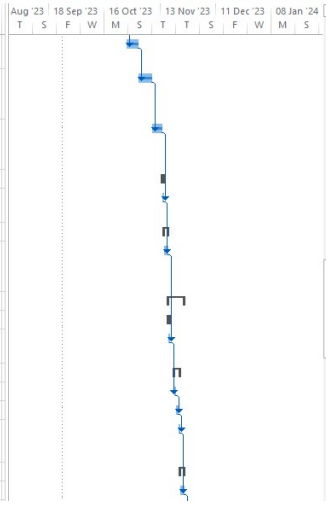
GANTT CHART

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | % Complete |
|---|---|---|---|---|---|---|---|
| 23 | ✓ | **Model Development** | **8 days** | **Mon 18-09-23** | **Wed 27-09-23** | | **100%** |
| 24 | ✓ | Algorithm Selection | 2 days | Mon 18-09-23 | Tue 19-09-23 | 22 | 100% |
| 25 | ✓ | Model Training | 2 days | Wed 20-09-23 | Thu 21-09-23 | 24 | 100% |
| 26 | ✓ | Model Validation and Testing | 2 days | Fri 22-09-23 | Mon 25-09-23 | 25 | 100% |
| 27 | ✓ | Model based personality trait prediction | 2 days | Tue 26-09-23 | Wed 27-09-23 | 26 | 100% |
| 28 | ✓ | **Model Evaluation** | **2 days** | **Thu 28-09-23** | **Fri 29-09-23** | | **100%** |
| 29 | ✓ | Calculate model evaluation metrics | 2 days | Thu 28-09-23 | Fri 29-09-23 | 27 | 100% |
| 30 | ✓ | **Multimodal Fusion** | **2 days** | **Mon 02-10-23** | **Tue 03-10-23** | | **100%** |
| 31 | ✓ | Model Averaging | 2 days | Mon 02-10-23 | Tue 03-10-23 | 29 | 100% |
| 32 | ✓ | Update Requirement Matrix | 1 day | Wed 04-10-23 | Wed 04-10-23 | 31 | 100% |
| 33 | ✓ | **Requirement Matrix Finalized** | 0 days | Wed 04-10-23 | Wed 04-10-23 | 32 | 100% |
| 34 | | **Design** | **37 days** | **Thu 05-10-23** | **Fri 24-11-23** | | **99%** |
| 35 | | **Detailed Design** | **31 days** | **Thu 05-10-23** | **Thu 16-11-23** | | **99%** |
| 36 | ✓ | **Big 5 Questionnaire** | **2 days** | **Thu 05-10-23** | **Fri 06-10-23** | | **100%** |
| 37 | ✓ | Designed google form to collect user responses to the Big Five Personality Traits questionnaire. | 2 days | Thu 05-10-23 | Fri 06-10-23 | 33 | 100% |
| 38 | | **Data Collection** | **5 days** | **Mon 09-10-23** | **Fri 13-10-23** | | **94%** |
| 39 | ✓ | Text and Image Scrapping from social media | 3 days | Mon 09-10-23 | Wed 11-10-23 | 37 | 100% |
| 40 | | Prepare Raw Dataset | 2 days | Thu 12-10-23 | Fri 13-10-23 | 39,37 | 85% |
| 41 | ✓ | **Data Preprocessing** | 4 days | Mon 16-10-23 | Thu 19-10-23 | 40 | 100% |
| 42 | ✓ | **Feature Extraction** | 2 days | Fri 20-10-23 | Mon 23-10-23 | 41 | 100% |
| 43 | ✓ | **Model Development** | **15 days** | **Tue 24-10-23** | **Mon 13-11-23** | | **100%** |
| 44 | ✓ | Algorithm Selection for Text-based and Image-based model | 3 days | Tue 24-10-23 | Thu 26-10-23 | 42 | 100% |



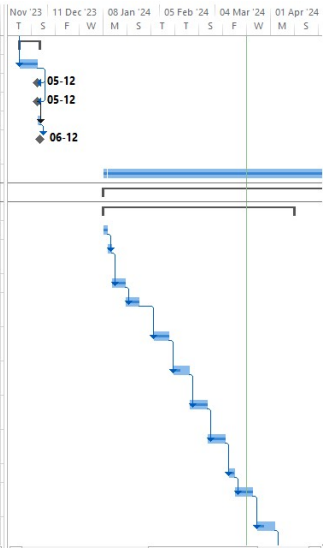

GANTT CHART

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | % Complete |
|---|---|---|---|---|---|---|---|
| 45 | ✓ | Model training for Text-based and Image-based models | 4 days | Fri 27-10-23 | Wed 01-11-23 | 44 | 100% |
| 46 | ✓ | Model Validation and Testing for Text-based and Image-based models | 5 days | Thu 02-11-23 | Wed 08-11-23 | 45 | 100% |
| 47 | ✓ | Personality trait predictions for Text-based and Image-based models separately | 3 days | Thu 09-11-23 | Mon 13-11-23 | 46 | 100% |
| 48 | ✓ | **Model Evaluation** | **1 day** | **Tue 14-11-23** | **Tue 14-11-23** | | **100%** |
| 49 | ✓ | Calculate model evaluation metrics | 1 day | Tue 14-11-23 | Tue 14-11-23 | 47 | 100% |
| 50 | ✓ | **Multimodal Fusion** | **2 days** | **Wed 15-11-23** | **Thu 16-11-23** | | **100%** |
| 51 | ✓ | Combining text-based prediction and image-based prediction using Model Averaging technique | 2 days | Wed 15-11-23 | Thu 16-11-23 | 49 | 100% |
| 52 | | **Test Plan Preparation** | **6 days** | **Fri 17-11-23** | **Fri 24-11-23** | | **98%** |
| 53 | | **Data Collection** | **1 day** | **Fri 17-11-23** | **Fri 17-11-23** | | **85%** |
| 54 | | Text and Image Scrapping from social media | 1 day | Fri 17-11-23 | Fri 17-11-23 | 51 | 85% |
| 55 | ✓ | **Model Development** | **3 days** | **Mon 20-11-23** | **Wed 22-11-23** | | **100%** |
| 56 | ✓ | Algorithm Selection | 1 day | Mon 20-11-23 | Mon 20-11-23 | 54 | 100% |
| 57 | ✓ | Model Validation and Testing | 1 day | Tue 21-11-23 | Tue 21-11-23 | 56 | 100% |
| 58 | ✓ | Text-based model and Image-based model personality trait predictions | 1 day | Wed 22-11-23 | Wed 22-11-23 | 57 | 100% |
| 59 | ✓ | **Multimodal Fusion module** | **2 days** | **Thu 23-11-23** | **Fri 24-11-23** | | **100%** |
| 60 | ✓ | Final averaged model Testing | 2 days | Thu 23-11-23 | Fri 24-11-23 | 58 | 100% |



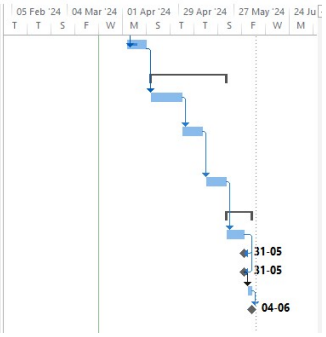
GANTT CHART

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | % Complete |
|---|---|---|---|---|---|---|---|
| 61 | ✓ | **Phase 1 Closure** | **8 days** | **Mon 27-11-23** | **Wed 06-12-23** | | **100%** |
| 62 | ✓ | Prepare 7th Semester Project Report | 7 days | Mon 27-11-23 | Tue 05-12-23 | 60 | 100% |
| 63 | ✓ | Updated Requirement Matrix | 0 days | Tue 05-12-23 | Tue 05-12-23 | 62FF | 100% |
| 64 | ✓ | **Updated Project Plan** | 0 days | Tue 05-12-23 | Tue 05-12-23 | 62FF | 100% |
| 65 | ✓ | Project Viva | 1 day | Wed 06-12-23 | Wed 06-12-23 | 64 | 100% |
| 66 | ✓ | **Approved Project Report - 7th Semester** | 0 days | Wed 06-12-23 | Wed 06-12-23 | 65 | 100% |
| 67 | ✓ | **Semester Gap** | 107.05 days | Mon 08-01-24 | Wed 05-06-24 | | 100% |
| 68 | | **Phase 2: 8th Semester Activities** | **107 days** | **Mon 08-01-24** | **Tue 04-06-24** | | **52%** |
| 69 | | **Coding & Unit Testing** | **70 days** | **Mon 08-01-24** | **Fri 12-04-24** | | **82%** |
| 70 | ✓ | Choosing a Notebook | 2 days | Mon 08-01-24 | Tue 09-01-24 | | 100% |
| 71 | | Collect cognitive scores through questionnaire | 2 days | Wed 10-01-24 | Thu 11-01-24 | 70 | 50% |
| 72 | ✓ | Training/Testing set-up | 5 days | Fri 12-01-24 | Thu 18-01-24 | 71 | 100% |
| 73 | ✓ | Logistic Regression based classification for text data | 5 days | Fri 19-01-24 | Thu 25-01-24 | 72 | 100% |
| 74 | ✓ | Random Forest based classification for text data | 6 days | Fri 02-02-24 | Fri 09-02-24 | 73 | 100% |
| 75 | | Gradient boosting classification for text data | 6 days | Mon 12-02-24 | Mon 19-02-24 | 74 | 65% |
| 76 | ✓ | ANN for image feature extraction and classification | 7 days | Tue 20-02-24 | Wed 28-02-24 | 75 | 100% |
| 77 | ✓ | CNN for image feature extraction and classification | 7 days | Thu 29-02-24 | Fri 08-03-24 | 76 | 100% |
| 78 | | Accuracy comparison of models | 3 days | Mon 11-03-24 | Wed 13-03-24 | 77 | 85% |
| 79 | ✓ | Prediction process using trained models in pickle format | 7 days | Thu 14-03-24 | Fri 22-03-24 | 78 | 100% |
| 80 | | Combining text and image based predictions | 7 days | Mon 25-03-24 | Tue 02-04-24 | 79 | 50% |




GANTT CHART

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | % Complete |
|---|---|---|---|---|---|---|---|
| 81 | | Correlating cognitive scores with dominant personality traits | 8 days | Wed 03-04-24 | Fri 12-04-24 | 80 | 40% |
| 82 | | **System Integration Testing** | 28 days | **Mon 15-04-24** | **Wed 22-05-24** | | **0%** |
| 83 | | Integrate all developed modules and components into a unified system | 12 days | Mon 15-04-24 | Tue 30-04-24 | 81 | 0% |
| 84 | | Test the interaction between the text and image data processing components | 8 days | Wed 01-05-24 | Fri 10-05-24 | 83 | 0% |
| 85 | | Check all dependencies and code are operational | 8 days | Mon 13-05-24 | Wed 22-05-24 | 84 | 0% |
| 86 | | **Project Closure** | **9 days** | **Thu 23-05-24** | **Tue 04-06-24** | | **0%** |
| 87 | | Prepare 8th Semester Project Report | 7 days | Thu 23-05-24 | Fri 31-05-24 | 85 | 0% |
| 88 | | **Updated Requirement Matrix** | 0 days | Fri 31-05-24 | Fri 31-05-24 | 87FF | 0% |
| 89 | | **Updated Project Plan** | 0 days | Fri 31-05-24 | Fri 31-05-24 | 87FF | 0% |
| 90 | | Review by Faculties | 2 days | Mon 03-06-24 | Tue 04-06-24 | 89 | 0% |
| 91 | | **Approved Project Report - 7th Semester** | 0 days | Tue 04-06-24 | Tue 04-06-24 | 90 | 0% |



## 5.3 Cost Analysis

**Function Point Worksheet**

| Measurement parameter | Count | | Weighting Factor: simple | average | complex | Choice | | |
|---|---|---|---|---|---|---|---|---|
| # of user inputs | 2 | X | 3 | 4 | 6 | 4 | = | 8 |
| # of user outputs | 2 | X | 4 | 5 | 7 | 7 | = | 14 |
| # of user inquiries | 0 | X | 3 | 4 | 6 | 4 | = | 0 |
| # of files | 1 | X | 7 | 10 | 15 | 10 | = | 10 |
| # of external interfaces | 1 | X | 5 | 7 | 10 | 5 | = | 5 |
| | | | | | Count-total (UFP)= | | | 37 |

| | | | |
|---|---|---|---|
| Rate each factor on a scale of 0 to 5: | 0 - No Influence | 1 - Incidental | 2 - Moderate |
| | 3 - Average | 4 - Significant | 5 - Essential |

| | |
|---|---|
| 1. Does the system require reliable backup and recovery? | 5 |
| 2. Are data communications required? | 5 |
| 3. Are there distributed processing functions? | 4 |
| 4. Is performance critical? | 5 |
| 5. Will the system run in an existing, heavily utilized operational environment? | 4 |
| 6. Does the system require on-line data entry? | 5 |
| 7. Does the on-line data entry require the input transaction to be built over multiple screens or operations? | 2 |
| 8. Are the master files updated on-line? | 3 |
| 9. Are the inputs, outputs, files, or inquiries complex? | 5 |
| 10. Is the internal processing complex? | 5 |
| 11. Is the code designed to be reusable? | 5 |
| 12. Are conversion and installation included in the design? | 3 |
| 13. Is the system designed for multiple installations in different organizations? | 3 |
| 14. Is the application designed to facilitate change and ease of use by the user? | 4 |
| Total Complexity Adjustment Value = | 58 |

| | | |
|---|---|---|
| Product Complexity Adjustment (PC) = | [.65+.01*CAV] | |
| | 1.23 | |
| Total Adjusted Function Point (FP) = | UFP * PC | |
| = | 45.51 | |
| Language Factor (LF) = | 60 | |
| Source Lines of Code (SLOC) = | FP * LF | |
| = | 2730.6 | |

As the team size is small, the problem is well understood and also the team members have a nominal experience regarding the problem. That's why we have chosen the BASIC COCOMO MODEL (ORGANIC).

Effort $=a*(KLOC)^b$ PM

T dev $=c*(efforts)^d$ Months

Where, KLOC is the estimated size of the software product indicate in Kilo Lines of

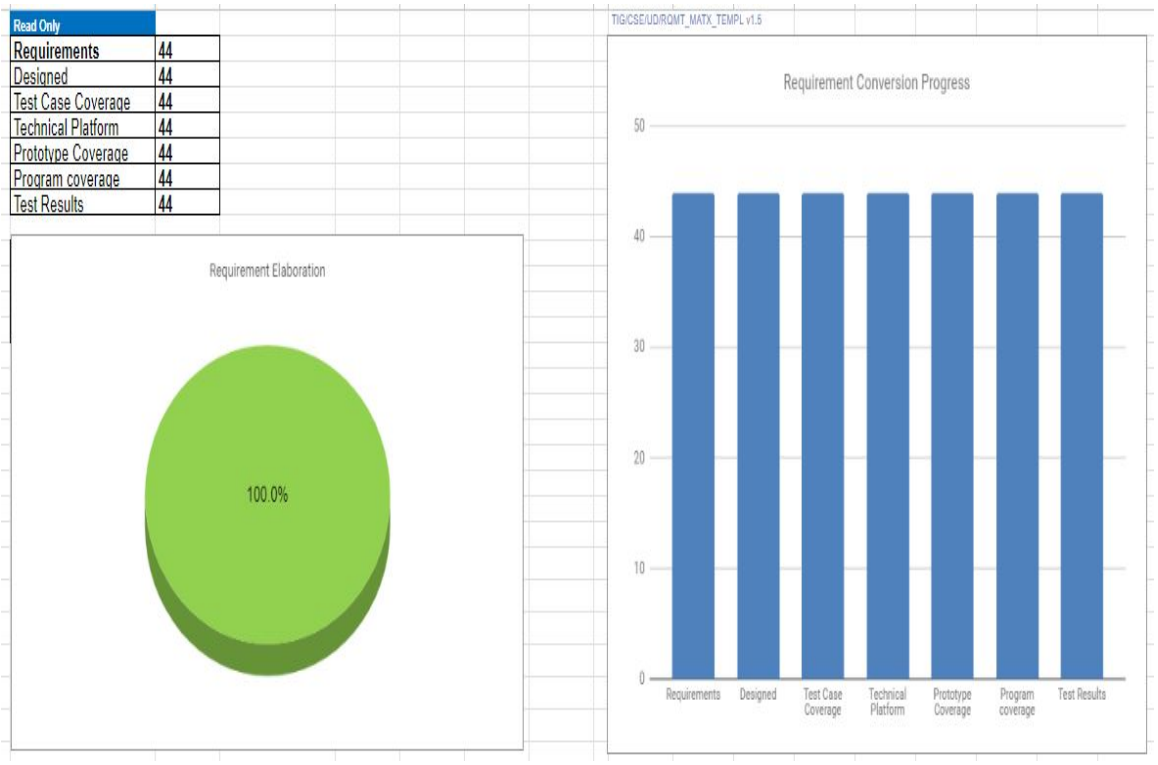
Code, a1, a2, b1, b2 are constants for each group of software products, T dev is the estimated time to develop the software, expressed in months, Effort is the total effort required to develop the software product, expressed in person months (PMs).

| BASIC COCOMO MODEL (ORGANIC) | | | | | | | |
|---|---|---|---|---|---|---|---|
| LOC | 2730.6 | | | | | | |
| KLOC | 2.7306 | | Cost per month (in Rs) | 50000 | | a | 2.4 |
| | | | | | | b | 1.05 |
| | | | | | | c | 0.38 |
| EFFORT (in PM) | | 6.890999714 | | | | | |
| | | | | | | | |
| $T_{dev}$ (in M) | | 5.205797336 | | | | | |
| | | | | | | | |
| STAFF SIZE (in P) | | 1 | | | | | |
| | | | | | | | |
| Cost to develop (in Rs) | | 260289.8668 | | | | | |

# 6 Requirement Analysis

## 6.1 Requirement Matrix

| Rqmt ID | Requirement Item | Requirement Analysis Status | Design Module | Design Reference (section# under project Report) |
|---|---|---|---|---|
| **QBIG5-1** | **Big 5 Questionnaire** | Completed | QBIG5 | 7.3.1 |
| QBIG5-1.1 | Collect responses for IPIP-Big 5 Questionnaire | Completed | QBIG5 | 7.3.1.1 |
| QBIG5-1.2 | Calculate Big 5 personality trait scores | Completed | QBIG5 | 7.3.1.2 |
| **QCOG-1** | **Cognitive Scores Questionnaire** | Completed | QCOG | 7.3.2 |
| QCOG-1.1 | Calculate cognitive scores for questionnaire responses | Completed | QCOG | 7.3.2.1 |
| **TXTDAT-1** | **Data Collection** | Completed | TXTDAT | 7.3.3 |
| TXTDAT-1.1 | Collect social media credentials for scrapping | Completed | TXTDAT | 7.3.3.1 |
| TXTDAT-1.2 | Scrap textual data from social media | Completed | TXTDAT | 7.3.3.2 |
| TXTDAT-1.3 | Scrap images from social media | Completed | TXTDAT | 7.3.3.3 |
| **DATPRO-1** | **Data Preprocessing** | Completed | DATPRO | 7.3.4 |
| DATPRO-1.1 | Textual Data Preprocessing | Completed | DATPRO | 7.3.4.1 |
| DATPRO-1.2 | Image Data Preprocessing | Completed | DATPRO | 7.3.4.2 |
| **FEXT-1** | **Feature Extraction** | Completed | FEXT | 7.3.5 |
| FEXT-1.1 | Textual Feature Extraction | Completed | FEXT | 7.3.5.1 |
| FEXT-1.2 | Image Feature Extraction | Completed | FEXT | 7.3.5.2 |
| **MOD-1** | **Post-Text based Model** | Completed | MOD | 7.3.6 |
| MOD-1.1 | Algorithm Selection | Completed | MOD | 7.3.6.1 |
| MOD-1.2 | Model Training | Completed | MOD | 7.3.6.2 |
| MOD-1.3 | Model Validation and Evaluation | Completed | MOD | 7.3.6.3 |
| MOD-1.4 | Model based personality trait scores prediction | Completed | MOD | 7.3.6.4 |
| **MOD-2** | **Comment-Text based Model** | Completed | MOD | 7.3.7 |
| MOD-2.1 | Algorithm Selection | Completed | MOD | 7.3.7.1 |
| MOD-2.2 | Model Training | Completed | MOD | 7.3.7.2 |
| MOD-2.3 | Model Validation and Evaluation | Completed | MOD | 7.3.7.3 |
| MOD-2.4 | Model based personality trait scores prediction | Completed | MOD | 7.3.7.4 |
| **MOD-3** | **Post-Image based Model** | Completed | MOD | 7.3.8 |
| MOD-3.1 | Algorithm Selection | Completed | MOD | 7.3.8.1 |
| MOD-3.2 | Model Training | Completed | MOD | 7.3.8.2 |
| MOD-3.3 | Model Evaluation | Completed | MOD | 7.3.8.3 |
| MOD-3.4 | Model based personality trait scores prediction | Completed | MOD | 7.3.8.4 |
| **MOD-4** | **Profile-Image based Model** | Completed | MOD | 7.3.9 |
| MOD-4.1 | Algorithm Selection | Completed | MOD | 7.3.9.1 |
| MOD-4.2 | Model Training | Completed | MOD | 7.3.9.2 |
| MOD-4.3 | Model Evaluation | Completed | MOD | 7.3.9.3 |
| MOD-4.4 | Model based personality trait scores prediction | Completed | MOD | 7.3.9.4 |
| **MODFUS-1** | **Multimodal Fusion and Final Personality Trait Scores Prediction** | Completed | MODFUS | 7.3.10 |
| MODFUS-1.1 | Calculate weightage for each model | Completed | MODFUS | 7.3.10.1 |
| MODFUS-1.2 | Combine models to get personality trait scores prediction | Completed | MODFUS | 7.3.10.2 |
| **MOD-5** | **Cognitive Scores vs Personality Trait Scores Model** | Completed | MOD | 7.3.11 |
| MOD-5.1 | Algorithm Selection | Completed | MOD | 7.3.11.1 |
| MOD-5.2 | Model Training | Completed | MOD | 7.3.11.2 |
| MOD-5.3 | Model Validation and Evaluation | Completed | MOD | 7.3.11.3 |
| MOD-5.4 | Model based cognitive scores prediction | Completed | MOD | 7.3.11.4 |
| **COGPRED-1** | **Predict cognitive scores from predicted personality trait scores** | Completed | COGPRED | 7.3.12 |



| Read Only | |
|---|---|
| Requirements | 44 |
| Designed | 44 |
| Test Case Coverage | 44 |
| Technical Platform | 44 |
| Prototype Coverage | 44 |
| Program coverage | 44 |
| Test Results | 44 |

## 6.2 Requirement Elaboration

### 6.2.1 Big 5 Questionnaire

#### 6.2.1.1 Collect responses for IPIP-Big 5 Questionnaire

Created a google form for users to collect responses to the Big Five Personality Traits (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism) questionnaire.

#### 6.2.1.2 Calculate Big 5 personality trait scores

Used the IPIP Scale Scoring instructions to calculate scores for each of the Big Five Personality Traits based on the user's questionnaire responses.

### 6.2.2 Cognitive Scores Questionnaire

#### 6.2.2.1 Calculate cognitive scores for questionnaire responses

Create tables for questions, responses, and scores and stored data in csv. Implement algorithms for calculating and normalizing cognitive scores.

### 6.2.3 Data Collection

#### 6.2.3.1 Collect social media credentials for scrapping

Designed a google form to collect social media credentials like Facebook and LinkedIn profile links from the participants.

#### 6.2.3.2 Scrap textual data from social media

The post and comment data from Facebook and LinkedIn is taken. For scrapping some libraries like nltk and numpy should be installed.

- Google chrome should be installed, and some library like WebDriver,BeautifulSoup for HTML parsing, should be installed to open the Facebook and LinkedIn in chrome.

#### 6.2.3.3 Scrap images from social media

The post images and the profile images of the users are scrapped from Facebook. For scrapping some libraries like numpy, pickle and os need to be installed.

- Google chrome should be installed, and some library like WebDriver,BeautifulSoup for HTML parsing, should be installed to open the Facebook and LinkedIn in chrome.

### 6.2.4 Data Preprocessing

Required library installed in the system like numpy, NLTK.

#### 6.2.4.1 Textual Data Preprocessing

Required libraries installed like GoogleTranslator, stopwords, word-tokenize, nltk, WordNetLemmatizer.

#### 6.2.4.2 Image Data Preprocessing

Required library installed in the system like MTCNN, CV2

### 6.2.5 Feature Extraction

Different feature extraction need to be applied.

#### 6.2.5.1 Textual Feature Extraction

Fast text is needed to convert the prepossessed text in to vector form and as Fast text gives 300 sized vectors for each word, so mean of that is taken.

#### 6.2.5.2 Image Feature Extraction

Different technique like Gaussian blur, canny edge detection ,threshold and data augmentation is needed to get different featured images.

#### 6.2.6 Post-Text based Model

Post text vector is required.

##### 6.2.6.1 Algorithm Selection

Different regression models needed to be installed.

##### 6.2.6.2 Model Training

Model needed to be trained on the selected algorithms.

##### 6.2.6.3 Model Validation and Evaluation

Trained model is need to run the test input data and compare with output test data.

##### 6.2.6.4 Model based personality trait scores prediction

Model is supplied with a new user post text vector to get the personality trait scores.

#### 6.2.7 Comment-Text based Model

Comment text vector is required.

##### 6.2.7.1 Algorithm Selection

Various algorithms such as linear regression, random forest, and support vector regression need to be considered for the comment-text based model.

##### 6.2.7.2 Model Training

Train the selected algorithms on the preprocessed textual data obtained from comments on social media platforms.

#### 6.2.7.3 Model Validation and Evaluation

Validate the trained models using techniques like k-fold cross-validation and evaluate their performance using metrics such as mean squared error (MSE) and R-squared.

#### 6.2.7.4 Model based personality trait scores prediction

Utilize the trained models to predict personality trait scores based on the textual data extracted from comments on social media.

### 6.2.8 Post-Image based Model

#### 6.2.8.1 Algorithm Selection

Consider convolutional neural networks (CNNs) and transfer learning techniques such as VGG16 for the post-image based model

#### 6.2.8.2 Model Training

Train the selected algorithms on the preprocessed image data obtained from posts on social media platforms.

#### 6.2.8.3 Model Evaluation

Evaluate the trained models using techniques like mean squared error (MSE) and other relevant evaluation metrics to assess their performance.

#### 6.2.8.4 Model based personality trait scores prediction

Utilize the trained models to predict personality trait scores based on the image data extracted from posts on social media.

### 6.2.9 Profile-Image based Model

#### 6.2.9.1 Algorithm Selection

Consider similar algorithms as the post-image based model for the profile-image based model, such as CNNs and transfer learning techniques.

#### 6.2.9.2 Model Training

Train the selected algorithms on the preprocessed profile image data obtained from social media platforms.

#### 6.2.9.3 Model Evaluation

Evaluate the trained models using relevant evaluation metrics to assess their performance and generalization capability.

#### 6.2.9.4 Model based personality trait scores prediction

Utilize the trained models to predict personality trait scores based on the profile image data extracted from social media profiles.

### 6.2.10 Multimodal Fusion and Final Personality Trait Scores Prediction

#### 6.2.10.1 Calculate weightage for each model

Determine the relative importance or weightage of each model .

#### 6.2.10.2 Combine models to get personality trait scores prediction

Fuse the predictions from different models using appropriate fusion techniques .

### 6.2.11 Cognitive Scores vs Personality Trait Scores Model

#### 6.2.11.1 Algorithm Selection

Consider regression algorithms suitable for predicting cognitive scores based on personality trait scores, such as linear regression or support vector regression.

#### 6.2.11.2 Model Training

Train the selected algorithms on the dataset consisting of personality trait scores as features and cognitive scores as labels.

#### 6.2.11.3 Model Validation and Testing

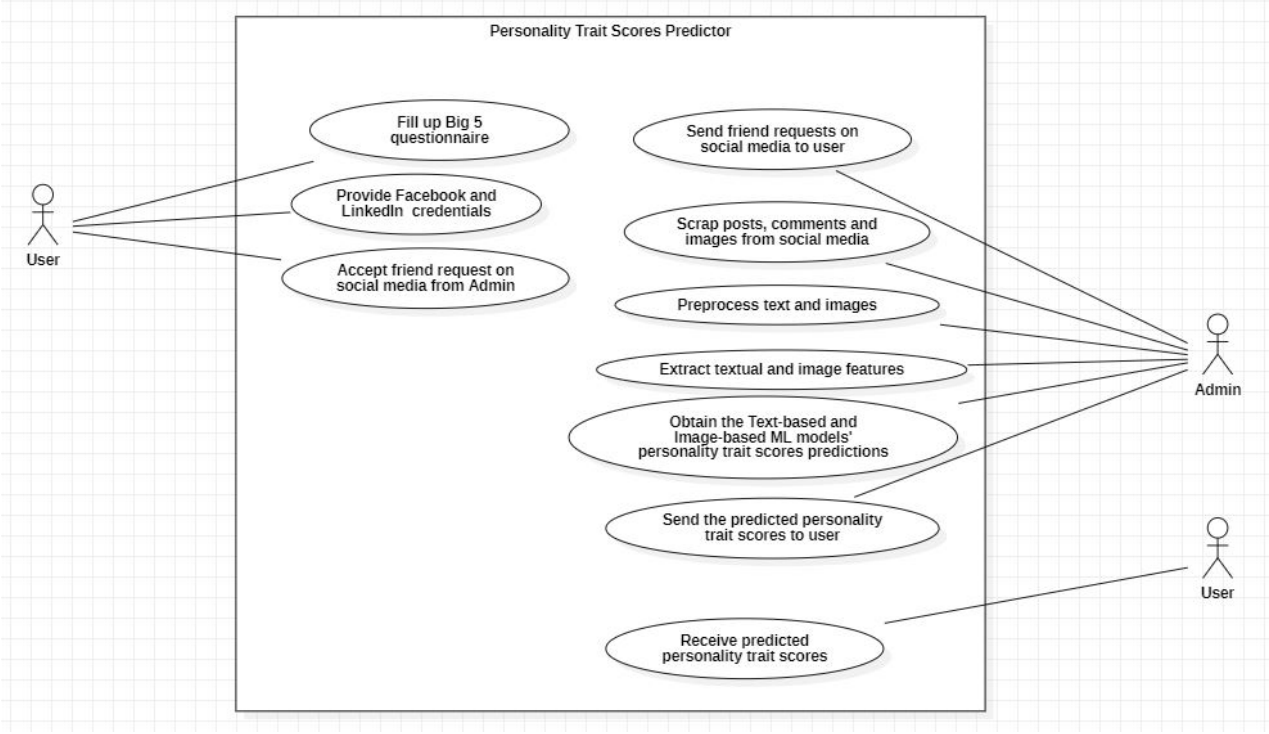
Validate the trained models using standard validation techniques and evaluate their performance using relevant evaluation metrics.

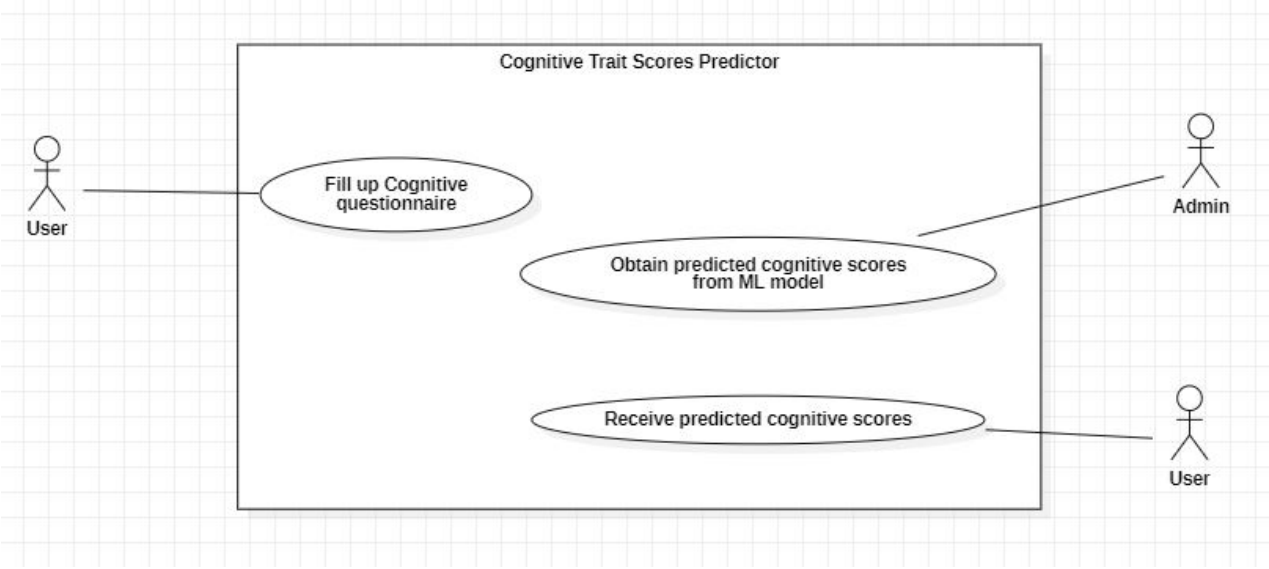
#### 6.2.11.4 Model based cognitive scores prediction

Utilize the trained models to predict cognitive scores based on the predicted personality trait scores.

### 6.2.12 Predict cognitive scores from predicted personality trait scores

Utilize the trained models to predict cognitive scores from the predicted personality trait scores obtained .




Cognitive Trait Scores Predictor
User
Fill up Cognitive questionnaire
Obtain predicted cognitive scores from ML model
Admin
Receive predicted cognitive scores
User

# 7 Design

## 7.1 Technical Environment

Hardware Requirements

- CPU
- RAM 2 GB
- HDD 200 GB
- Network adapter
- UPS

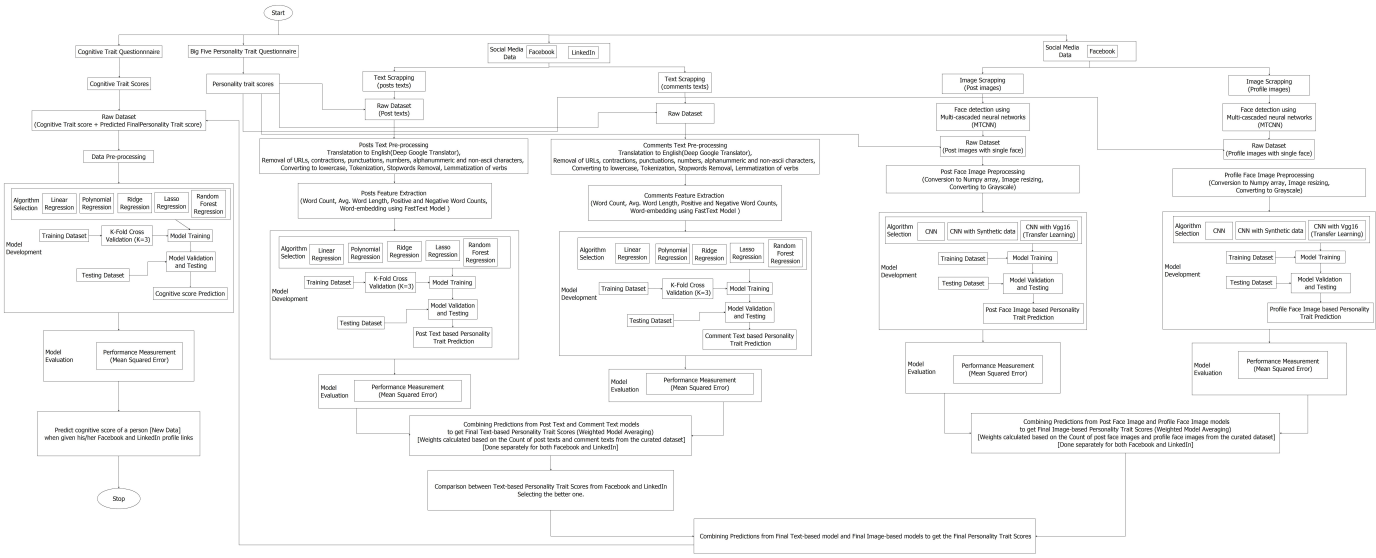
Operating System Requirements

- Windows OS – latest version
- Any Linux OS – latest version
- MAC OS

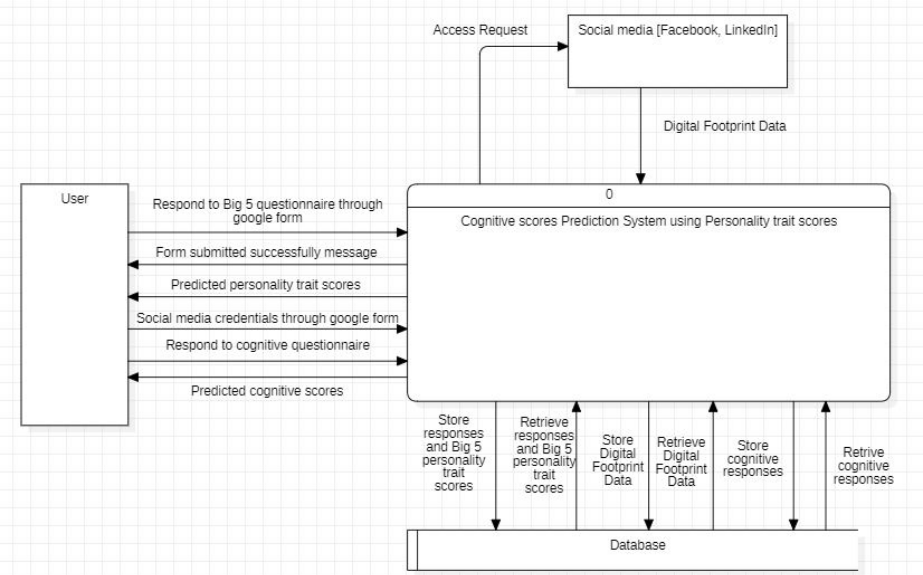
Software Requirements

- Web Scraping: BeautifulSoup, requests
- Text Processing: NLTK, unidecode
- Image Processing: OpenCV
- Machine Learning: Keras, Scikit-learn, numpy, pandas
- Deep Learning Framework: TensorFlow, VS Code
- Version Control: Git and GitHub

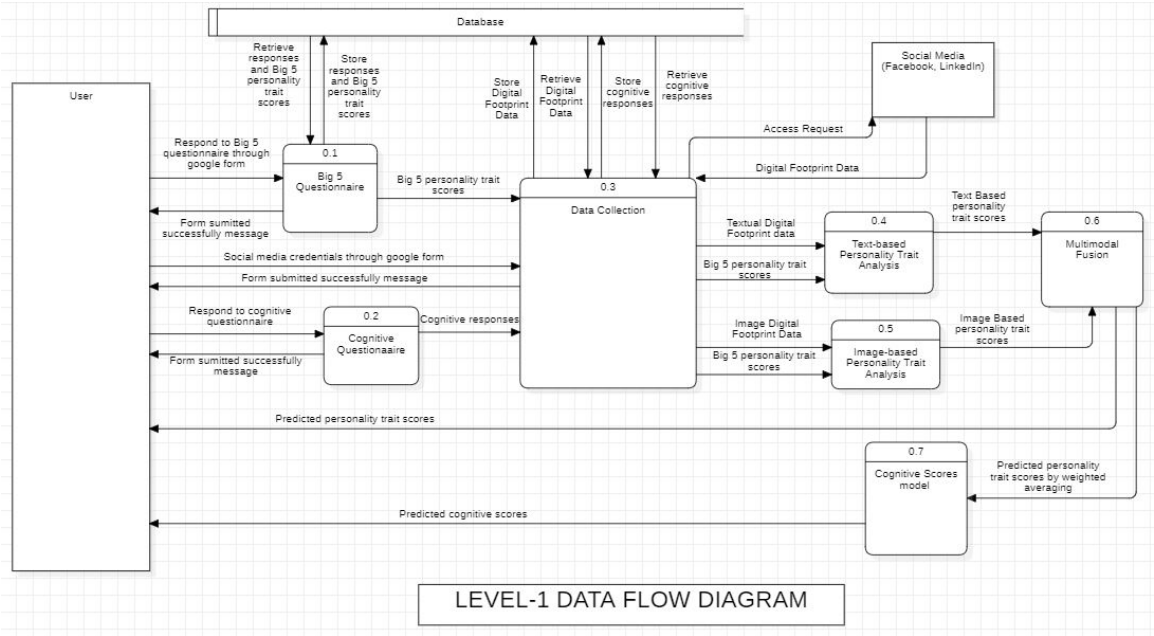
## 7.2 Hierarchy of Modules




## 7.3 Detailed Design




LEVEL-0 DATA FLOW DIAGRAM




LEVEL-1 DATA FLOW DIAGRAM

### 7.3.1 Big 5 Questionnaire

#### 7.3.1.1 Collect responses for IPIP-Big 5 Questionnaire

Created a google form for users to collect responses to the Big Five Personality Traits (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism) questionnaire.

#### 7.3.1.2 Calculate Big 5 personality trait scores

Used the IPIP Scale Scoring instructions to calculate scores for each of the Big Five Personality Traits based on the user's questionnaire responses.

### 7.3.2 Cognitive Scores Questionnaire

#### 7.3.2.1 Calculate cognitive scores for questionnaire responses

Using bloom's taxonomy which provides a framework for understanding the levels of cognitive complexity involved in learning and thinking, the cognitive questions were framed. Based on these we calculate a cognitive score for each individual.

### 7.3.3 Data Collection

#### 7.3.3.1 Collect social media credentials for scrapping

Designed a google form to collect social media credentials like Facebook and LinkedIn profile links from the participants.

#### 7.3.3.2 Scrap textual data from social media

To scrap textual data (posts, comments) from social media handles like Facebook and LinkedIn, the following steps were followed:

* **Initialize Browser using Selenium WebDriver**

  It allows to programmatically control a web browser, useful for web scraping and automated testing.

  - **Import Necessary Libraries:**Import the required modules from Selenium.

- **Set Browser Options:**Configure the browser options (e.g., headless mode, disable images, set window size).
- **Initialize the WebDriver:**Create an instance of the WebDriver with the specified options.

* **Install and Configure Chrome Extension with Selenium WebDriver**

  It allows for enhanced web scrapping.

  - **Download the Chrome Extension:**Obtain the .crx file of the Chrome extension needed. It can be downloaded directly from the Chrome Web Store or other sources.
  - **Set Up Chrome Options:**Use Selenium's Options class to add the extension and configure any additional settings.

* **Navigate to Profile Page (Using Browser Automation)**

  Either the profile URL is directly entered in the browser or it is located and the link that navigates to the profile page is clicked.

* **Handle Pop-ups and Notifications (If any)**

  Essential for maintaining smooth automation workflows, especially when dealing with websites that frequently display alerts and modal dialogs.

  - **Check for Pop-ups:**Detect the presence of pop-ups (such as alert boxes, confirmation boxes, or modal dialogs).
  - **Handle Notifications:**Manage browser notifications that may appear. Adjust the element locators and waiting times as needed based on the specific behavior of the website.

* **Scroll to Load More Content using Selenium WebDriver**

  This can be especially useful for websites that load content dynamically as you scroll down the page like social media platforms.

  - **Bring the Browser to the Top:**Ensure that the browser starts at the top of the page.
  - **Automate Scrolling:**Use JavaScript to scroll the browser window down in intervals to load more content.

* **Extract HTML Data (Using BeautifulSoup)**

  This method automates the process of navigating to a webpage, retrieving its HTML content, and parsing the data to extract relevant information.

  - **Set Up and Retrieve HTML with Selenium:**Use Selenium to navigate to the webpage and retrieve the HTML content.
  - **Parse HTML with BeautifulSoup:**Use BeautifulSoup to parse the retrieved HTML content.
  - **Extract Relevant Data:**Finally, use BeautifulSoup methods to find and extract the desired data.

  Adjust the data extraction logic based on the specific structure of the webpage and data needed.

* **Extract Post Text and Comment Text (Separately)**

  Identify the HTML structure for posts and comments and extract the text.

* **Store Extracted Textual Data**

  Store the textual data extracted from posts and comments under separate columns in CSV.

#### 7.3.3.3 Scrap images from social media

To scrap images (posts, profiles) from social media handles like Facebook and LinkedIn, the following steps were followed:

* **Initialize Browser using Selenium WebDriver**

  It allows to programmatically control a web browser, useful for web scraping and automated testing.

  - **Import Necessary Libraries:** Import the required modules from Selenium.
  - **Set Browser Options:**Configure the browser options (e.g., headless mode, disable images, set window size).
  - **Initialize the WebDriver:**Create an instance of the WebDriver with the specified options.

* **Install and Configure Chrome Extension with Selenium WebDriver**

  It allows for enhanced web scrapping.

  - **Download the Chrome Extension:**Obtain the .crx file of the Chrome extension needed. It can be downloaded directly from the Chrome Web Store or other sources.
  - **Set Up Chrome Options:**Use Selenium's Options class to add the extension and configure any additional settings.

* **Navigate to Profile Page (Using Browser Automation)**

  Either the profile URL is directly entered in the browser or it is located and the link that navigates to the profile page is clicked.

* **Handle Pop-ups and Notifications (If any)**

  Essential for maintaining smooth automation workflows, especially when dealing with websites that frequently display alerts and modal dialogs.

  - **Check for Pop-ups:**Detect the presence of pop-ups (such as alert boxes, confirmation boxes, or modal dialogs).
  - **Handle Notifications:**Manage browser notifications that may appear. Adjust the element locators and waiting times as needed based on the specific behavior of the website.

* **Scroll to Load More Content using Selenium WebDriver**

  This can be especially useful for websites that load content dynamically as you scroll down the page like social media platforms.

  - **Bring the Browser to the Top:**Ensure that the browser starts at the top of the page.
  - **Automate Scrolling:**Use JavaScript to scroll the browser window down in intervals to load more content.

* **Extract HTML Data (Using BeautifulSoup)**

  This method automates the process of navigating to a webpage, retrieving its HTML content, and parsing the data to extract relevant information.

- **Set Up and Retrieve HTML with Selenium:**Use Selenium to navigate to the webpage and retrieve the HTML content.
- **Parse HTML with BeautifulSoup:**Use BeautifulSoup to parse the retrieved HTML content.
- **Extract Relevant Data:**Finally, use BeautifulSoup methods to find and extract the desired data.

Adjust the data extraction logic based on the specific structure of the webpage and data needed.

* **Extract Post Images and Profile Images (Separately)**

  Libraries like 'requests' and 'numpy' are used to achieve this.

  - **Extract Image URLs:**Identify the HTML elements for post and profile images and extract their URLs.
  - **Convert URLs to Image Byte Arrays:**Download the images from the URLs and convert them to byte arrays.
  - **Store as 3D Arrays:**Convert the byte arrays to 3D arrays (numpy arrays) representing the images.

* **Store Extracted Image Data**

  Before storing the images, a pickle file must be created.

  - The images extracted from posts and profiles are saved to a Python dictionary using different key-value pairs.
  - Then these dictionaries are serialized to bytes using Pickle.
  - The pickled bytes are encoded using Base64 encoding and the encoded data is written to the pickle file already created.

* **View the stored images data**

  - The encoded data is read from the pickle file.
  - Then using Base64 decoding, the pickled bytes is decoded.
  - Finally, the stored dictionary containing images is obtained after deserialization of the decoded data from pickle file.
  - Using the appropriate key-value pairs and libraries like Matplotlib, the images can be viewed.

### 7.3.4 Data Preprocessing

#### 7.3.4.1 Textual Data Preprocessing

The steps of textual data processing are as follows:

* **Translate non-English text to English**
  - Install deep-translator tool.
  - Import GoogleTranslator library from deep-translator.
  - Specify the source language as 'auto' and target language as 'en' in the translate function.
* **Remove URLs**

  Regular expressions (regex) patterns specific to URLs are used to identify and remove URLs from the text. Python function using this regex pattern takes a list of texts and returns a list of cleaned texts without URLs.
* **Replace Contractions**

  To expand contractions in the text (e.g., " I'm " to "I am"), install the contractions library in Python. This library helps to expand contractions in English text efficiently.
* **Remove Punctuations**

  Regular expressions (regex) patterns specific to punctuations are used to identify and remove punctuations from the text. Python function using this regex pattern takes a list of texts and returns a list of cleaned texts without punctuations.
* **Tokenize Words**

  Install nltk library and use word_tokenize function to split the text into individual words for further preprocessing.
* **Convert To Lowercase**

  Convert all text to lowercase to ensure uniformity. For example, "Apple", "apple", and "APPLE" will all be considered the same word. This prevents duplicate representations of the same word due to case differences.

* **Remove Numbers**
  - **Install inflect:**Install the inflect library.
  - **Initialize inflect engine:**Create an instance of the inflect engine.
  - **Check and Remove Numbers:**Iterate through the words, checking if each word is a digit using isdigit(). If it is, remove it from the text.
* **Remove Alphanumerics**

  Regular expressions (regex) patterns specific to alphanumeric characters are used to identify and remove words containing both alphabets and numbers from the text.
* **Remove Non-Ascii Characters**

  Filters out characters that are not part of the ASCII character set using Python's 'unicodedata' library.
* **Remove Stopwords**

  Stopwords are common words like "and", "the", "is", etc., that are often removed from text data during preprocessing to focus on more meaningful words. These can be removed using the 'nltk' library in Python, which provides a list of common stopwords in various languages.
* **Lemmatize Verbs**

  To lemmatize a verb means to reduce it to its base or dictionary form, which is called the lemma. For example: "Running" becomes "run". Used WordNetLemmatizer to lemmatize verbs.

#### 7.3.4.2 Image Data Preprocessing

The steps of textual data processing are as follows:

* **Face Detection using MTCNN Model**

  To perform face detection using the MTCNN (Multi-Task Cascaded Convolutional Neural Network) model, following are the steps:

  1. **Install the required libraries:**Install tensorflow, keras, and mtcnn libraries.

2. **Load the MTCNN model:**Use the MTCNN library to load the model.
3. **Load the image:**Load the image in which you want to detect faces.
4. **Detect faces:**Use the MTCNN model to detect faces in the image.
5. **Draw bounding boxes: Optionally, can draw bounding boxes around the detected faces to visualize the results.**

* **Filter Images with Detected Faces**
  - **Define the function to detect faces:**Define a function that takes a list of image paths and detects faces in each image.
  - **Filter single-face detected images:**Retains only those images that have been detected as containing a single face.

### 7.3.5 Feature Extraction

#### 7.3.5.1 Textual Feature Extraction

First, we need to load the preprocessed textual data stored in CSV format. Then the following steps take place:

* **Vectorize Text using FastText Model**

  Text vectorization involves converting text data into numerical representations that can be used for various natural language processing (NLP) tasks.

  FastText, developed by Facebook's AI Research (FAIR) lab, is an efficient and effective method for word representation and provides accurate word-embeddings.

  It extends the Word2Vec model by considering subword information, which improves the representation of rare words and out-of-vocabulary words. This is particularly useful in languages with rich morphology or for domain-specific jargon and can handle transliterated texts also.

  - **Install the Required Libraries:**Ensure that the Fasttext library installed or use gensim which also supports FastText.

- **Load a Pre-trained FastText Model:**Use a pre-trained FastText model, as pre-trained models (like those provided by Facebook) are sufficient and save a significant amount of time.
- **Dimensionality Reduction:**By default, the word-vectors of pre-trained model are of 300 dimensions. These can be reduced to 100 dimensions by loading the appropriate model.
- **Obtain word-embeddings:**Once the FastText model is ready, we can vectorize the text data to get the word-embeddings. The 'wv' attribute of the FastText model provides access to the word vectors.
- **Store word-embeddings:**It is stored in a dictionary and acts as textual feature since it captures the semantic context from the text.

* **Other Textual Features**

  (Word Count, Avg Word Length, Positive and Negative Count)

  - **Install the necessary libraries:**Install Python's Spacy library.
  - **Word Count:**Calculate the number of words in the text and then take an average of it. It can act as textual feauture as it may provide more information about the individual's personality. Eg. Extroverted individuals might use more words as they are often outgoing and expressive.
  - **Avg. Word Length:**Calculate the length of all words in the text and then take an average of it. It can act as textual feauture as it may provide more information about the individual's personality. Eg. Conscientious individuals might use words of larger lengths highlighting their carefulness facet.
  - **Positive and Negative Count:**Use a predefined list of positive and negative words with sentiment lexicons like the NRC Word-Emotion Association Lexicon. E.g. People with high conscientiousness express their prudence by restricting themselves from negative emotions. Also, Neuroticism is associated with use of less positive emotion words while frequent use is an indicator for extraversion, agreeableness and conscientiousness.

* **Store the textual features**

  Create a dictionary where each key is a feature name and the corresponding value is a list of feature values.

#### 7.3.5.2 Image Feature Extraction

First, we need to load the images from the pickle file where they are stored as Numpy arrays. Then the following steps take place:

* **Apply Gaussian Blur:** It can help reduce noise and detail, making detection of facial features more robust.
* **Canny edge detection:**It helps in identifying the edges in an image, which is useful for accurate detection of facial features.
* **Thresholding Images:**It is a technique used to convert an image to a binary image based on pixel intensity. This process involves setting a threshold value, and any pixel intensity above this value is set to the maximum value (white), while those below it are set to the minimum value (black).
* **Data Augmentation:**It is a technique used to increase the diversity of the dataset by applying various transformations. This can help improve the robustness and generalizability of the model. The ImageDataGenerator from Keras is used to apply data augmentation techniques such as rotation, width shift, height shift, shear, zoom, horizontal flip, and fill mode.
* **Reshaping Images:**Reshaping images to a consistent size is often necessary for image processing tasks, especially when using neural networks like CNN in our case.

### 7.3.6 Post-Text based Model

It is a regression model that takes textual features obtained from users' posts and their calculated personality trait scores (from self-reported Big 5 questionnaire responses) as input parameters.

#### 7.3.6.1 Algorithm Selection

Different machine learning algorithms are tried out to find the best one suitable for the collected dataset. They are as follows:

* **Multiple Linear Regression**
* **Polynomial Regression**
* **Ridge Regression**
* **Lasso Regression**
* **Random Forest Regression**

#### 7.3.6.2 Model Training

The model is trained using the following steps:

* **Train-Test Split:** The dataset is split into training dataset (90%) and testing dataset (10%). This splitting depends upon the size of the total dataset collected.
* **Model Fitting:** Based on the textual features extracted from posts and the target labels i.e. personality trait scores, the model gets trained.
* **Model Testing:** The trained model is used to obtain the predictions based on the testing dataset.

#### 7.3.6.3 Model Validation and Evaluation

– **K-Fold Cross-Validation:** It is a powerful technique for assessing the performance of a machine learning model, particularly its ability to generalize to unseen data.

**1. Splitting the Data:** The data is randomly partitioned into k equally sized folds where k is a hyper-parameter.

**2. Training and Validation:**

· For each fold i (where i ranges from 1 to k):

· Use the i-th fold as the validation set.

· Use the i-th fold as the validation set.

**3. Model Training:** Train the model on the training set and evaluate it on validation set.

**4. Performance Averaging:** Compute the performance metric (e.g. mean squared error) for each of the k folds and average them to get the overall performance estimate.

– **Calculate model evaluation metrics:** Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

#### 7.3.6.4 Model based personality trait scores prediction

The model is tested with textual data from posts of new users and their corresponding personality trait scores predicted by model are stored.

### 7.3.7 Comment-Text based Model

It is a regression model that takes textual features obtained from users' comments and their calculated personality trait scores (from self-reported Big 5 questionnaire responses) as input parameters.

#### 7.3.7.1 Algorithm Selection

Different machine learning algorithms are tried out to find the best one suitable for the collected dataset. They are as follows:

* Linear Regression
* Polynomial Regression
* Ridge Regression
* Lasso Regression
* Random Forest Regression

#### 7.3.7.2 Model Training

The model is trained using the following steps:

* **Train-Test Split:** The dataset is split into training dataset (90%) and testing dataset (10%). This splitting depends upon the size of the total dataset collected.
* **Model Fitting:** Based on the textual features extracted from posts and the target labels i.e. personality trait scores, the model gets trained.
* **Model Testing:** The trained model is used to obtain the predictions based on the testing dataset.

#### 7.3.7.3 Model Validation and Evaluation

– **K-Fold Cross-Validation:** It is a powerful technique for assessing the performance of a machine learning model, particularly its ability to generalize to unseen data.

**1. Splitting the Data:** The data is randomly partitioned into k equally sized folds where k is a hyper-parameter.

**2. Training and Validation:**

· For each fold i (where i ranges from 1 to k):

· Use the i-th fold as the validation set.

· Use the i-th fold as the validation set.

**3. Model Training:** Train the model on the training set and evaluate it on validation set.

**4. Performance Averaging:** Compute the performance metric (e.g. mean squared error) for each of the k folds and average them to get the overall performance estimate.

- **Calculate model evaluation metrics:** Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

#### 7.3.7.4 Model based personality trait scores prediction

The model is tested with textual data from posts of new users and their corresponding personality trait scores predicted by model are stored.

### 7.3.8 Post-Image based Model

It is a regression model that takes single-face detected image features obtained from users' posts and their calculated personality trait scores (from self-reported Big 5 questionnaire responses) as input parameters.

#### 7.3.8.1 Algorithm Selection

Different deep learning algorithms are tried out to find the best one suitable for the collected dataset. They are as follows:

* **Convoluted Neural Networks (CNNs)**

* **Convoluted Neural Networks (CNNs) with Synthetic Data**
* **Convoluted Neural Networks (CNNs) with VGG16 [Transfer Learning]**

#### 7.3.8.2 Model Training

The model is trained using the following steps:

* **Train-Test Split:** The dataset is split into training dataset (90%) and testing dataset (10%). This splitting depends upon the size of the total dataset collected.
* **Model Fitting:** Based on the textual features extracted from posts and the target labels i.e. personality trait scores, the model gets trained.
* **Model Testing:** The trained model is used to obtain the predictions based on the testing dataset.

#### 7.3.8.3 Model Evaluation

– **K-Fold Cross-Validation:** It is a powerful technique for assessing the performance of a machine learning model, particularly its ability to generalize to unseen data.

**1. Splitting the Data:** The data is randomly partitioned into k equally sized folds where k is a hyper-parameter.

**2. Training and Validation:**

· For each fold i (where i ranges from 1 to k):

· Use the i-th fold as the validation set.

· Use the i-th fold as the validation set.

**3. Model Training:** Train the model on the training set and evaluate it on validation set.

**4. Performance Averaging:** Compute the performance metric (e.g. mean squared error) for each of the k folds and average them to get the overall performance estimate.

– **Calculate model evaluation metrics:** Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

#### 7.3.8.4 Model based personality trait scores prediction

The model is tested with single-face detected images from posts of new users and their corresponding personality trait scores predicted by model are stored.

### 7.3.9 Profile-Image based Model

It is a regression model that takes single-face detected image features obtained from users' profile pictures and their calculated personality trait scores (from self-reported Big 5 questionnaire responses) as input parameters.

#### 7.3.9.1 Algorithm Selection

Different deep learning algorithms are tried out to find the best one suitable for the collected dataset. They are as follows:

* **Convoluted Neural Networks (CNNs)**
* **Convoluted Neural Networks (CNNs) with Synthetic Data**

* **Convoluted Neural Networks (CNNs) with VGG16 [Transfer Learning]**

#### 7.3.9.2 Model Training

The model is trained using the following steps:

* **Train-Test Split:** The dataset is split into training dataset (90%) and testing dataset (10%). This splitting depends upon the size of the total dataset collected.
* **Model Fitting:** Based on the textual features extracted from posts and the target labels i.e. personality trait scores, the model gets trained.
* **Model Testing:** The trained model is used to obtain the predictions based on the testing dataset.

#### 7.3.9.3 Model Evaluation

– **K-Fold Cross-Validation:** It is a powerful technique for assessing the performance of a machine learning model, particularly its ability to generalize to unseen data.

**1. Splitting the Data:** The data is randomly partitioned into k equally sized folds where k is a hyper-parameter.

**2. Training and Validation:**

· For each fold i (where i ranges from 1 to k):

· Use the i-th fold as the validation set.

· Use the i-th fold as the validation set.

**3. Model Training:** Train the model on the training set and evaluate it on validation set.

**4. Performance Averaging:** Compute the performance metric (e.g. mean squared error) for each of the k folds and average them to get the overall performance estimate.

– **Calculate model evaluation metrics:** Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

#### 7.3.9.4 Model based personality trait scores prediction

The model is tested with single-face detected images from posts of new users and their corresponding personality trait scores predicted by model are stored.

### 7.3.10 Multimodal Fusion and Final Personality Trait Scores Prediction

#### 7.3.10.1 Calculate weightage for each model

$$\text{Weightage of Post-text model} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\text{Count of post-texts per user})}{(\text{Count of total texts per user})}$$

$$\text{Weightage of Comment-text model} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\text{Count of comment-texts per user})}{(\text{Count of total texts per user})}$$

$$\text{Weightage of Post-image model} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\text{Count of post-images per user})}{(\text{Count of total images per user})}$$

$$\text{Weightage of Profile-image model} = \frac{1}{N}\sum_{i=1}^{N}\frac{(\text{Count of profile-images per user})}{(\text{Count of total images per user})}$$

#### 7.3.10.2 Combine models to get personality trait scores prediction

Using the principle of linear combination, multiply the predictions of each of the personality trait models with their corresponding weights and the add them to get the final predicted personality trait scores.

### 7.3.11 Cognitive Scores vs Personality Trait Scores Model

Social media credentials of new users are taken as input and their corresponding cognitive scores are predicted by cognitive model utilizing the predictions of the personality trait models.

#### 7.3.11.1 Algorithm Selection

Different machine learning algorithms are tried out to find the best one suitable for the collected dataset. They are as follows:

* Linear Regression
* Polynomial Regression
* Ridge Regression
* Lasso Regression
* Random Forest Regression
* Support Vector Regression

#### 7.3.11.2 Model Training

The model is trained using the following steps:

* **Train-Test Split:** The dataset is split into training dataset (90%) and testing dataset (10%). This splitting depends upon the size of the total dataset collected.
* **Model Fitting:** Based on the textual features extracted from posts and the target labels i.e. personality trait scores, the model gets trained.
* **Model Testing:** The trained model is used to obtain the predictions based on the testing dataset.

#### 7.3.11.3 Model Validation and Evaluation

- **K-Fold Cross-Validation:** It is a powerful technique for assessing the performance of a machine learning model, particularly its ability to generalize to unseen data.

  **1. Splitting the Data:** The data is randomly partitioned into k equally sized folds where k is a hyper-parameter.

  **2. Training and Validation:**

  · For each fold i (where i ranges from 1 to k):

  · Use the i-th fold as the validation set.

  · Use the i-th fold as the validation set.

  **3. Model Training:** Train the model on the training set and evaluate it on validation set.

  **4. Performance Averaging:** Compute the performance metric (e.g. mean squared error) for each of the k folds and average them to get the overall performance estimate.

- **Calculate model evaluation metrics:** Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

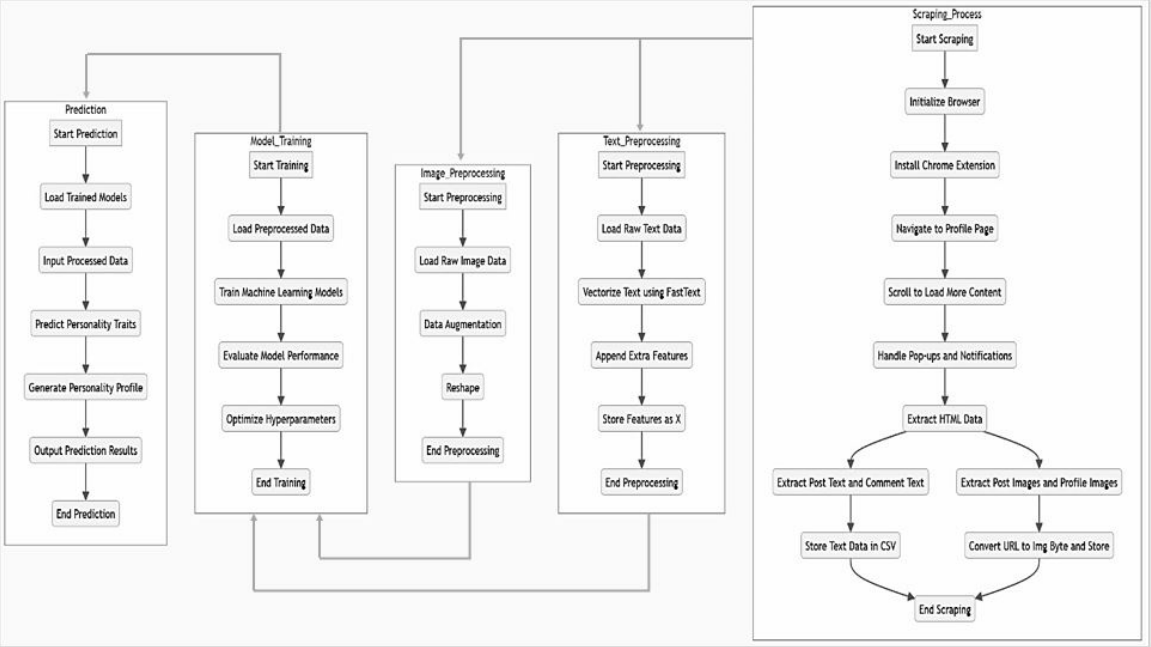
The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

#### 7.3.11.4 Model based Cognitive scores prediction

The model is tested with personality trait scores of new users and their corresponding cognitive scores predicted by model are stored.

### 7.3.12 Predict cognitive scores from predicted personality trait scores

Social media credentials of new users are taken as input and their corresponding cognitive scores are predicted by cognitive model utilizing the predictions of the personality trait models.




# 8 Implementation

## 8.1 Implementation Details

The following is the implementation details:

* **Big 5 Questionnaire:**
  - Collect responses for the IPIP-Big 5 Questionnaire.
  - Calculate Big 5 personality trait scores based on the questionnaire responses.

* **Cognitive Scores Questionnaire:**
  - Calculate cognitive scores for questionnaire responses.
* **Data Collection:**
  - Collect social media credentials for scraping.
  - Scrap textual data from social media.
  - Scrap images from social media.
* **Data Preprocessing:**
  - Preprocess textual data for removing unwanted data.
  - Preprocess image data includes change image link to img array.
* **Feature Extraction:**
  - Use text vectorizer technique to change text to vector.
  - Consider different features of image like edge, horizonatal flip, etc.
* **Post-Text based Model:**
  - Select algorithms for the model.
  - Train, validate, and test the model.
  - Predict personality trait scores based on textual data.
  - Calculate model evaluation metrics.
* **Comment-Text based Model:**
  - Select algorithms for the model.
  - Train, validate, and test the model.
  - Predict personality trait scores based on comment-text data.
  - Calculate model evaluation metrics.
* **Post-Image based Model:**
  - Select algorithms for the model.
  - Train, validate, and test the model.
  - Predict personality trait scores based on image data.
  - Calculate model evaluation metrics.
* **Profile-Image based Model:**
  - Select algorithms for the model.
  - Train, validate, and test the model.

- Predict personality trait scores based on profile images.
- Calculate model evaluation metrics.

* **Multimodal Fusion and Final Personality Trait Scores Prediction:**
  - Calculate weightage for each model.
  - Combine models to predict personality trait scores.
* **Cognitive Scores vs Personality Trait Scores Model:**
  - Select algorithms for the model.
  - Train, validate, and test the model.
  - Predict cognitive scores from obtained final personality trait scores.
  - Calculate model evaluation metrics.

## 8.2 System Installation Steps

* **Basic Setup:**
  - Install Python, Jupyter Notebook, Google Colab, GPU (optional).
* **For Scraping Part:**
  - Install Chrome browser and ChromeDriver.
  - Install Selenium and BeautifulSoup libraries.
  - Configure the ChromeDriver path.
  - Initialize the WebDriver.
* **For Text Preprocessing:**
  - Install necessary libraries like deep-translator, nltk, and inflect.
  - Download NLTK resources.
  - Implement text preprocessing techniques.
* **For Image Preprocessing:**
  - Install OpenCV, NumPy, and mtcnn libraries.
  - Implement image preprocessing techniques.
* **For Text Model:**
  - Install necessary libraries like scikit-learn and TensorFlow.
  - Implement algorithms like k-fold cross-validation, linear regression, etc.

* **For Image Model:**
  - Install TensorFlow and Keras libraries.
  - Implement CNN architectures or use pre-trained models like VGG16.

### 8.3 System Usage Instructions

* **Data Collection:**
  - Provide social media credentials.
  - Run scraping script to collect data.
* **Data Preprocessing:**
  - Run scripts for text and image preprocessing.
* **Feature Extraction:**
  - Execute scripts for feature extraction.
* **Model Training:**
  - Train models using preprocessed data.
* **Model Evaluation:**
  - Evaluate models using evaluation metrics.
* **Final Prediction:**
  - Combine model predictions to generate final personality trait scores and cognitive scores.

## 9 Test Results and Analysis

Performance metrics or evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms. When evaluating a ma- chine learning model, it is crucial to assess its predictive ability, generalization capability, and overall quality. Evaluation metrics provide objective criteria to measure these aspects. The choice of evaluation metrics depends on the specific problem domain, the type of data, and the desired outcome.

Metric used to measure the performance of the trained regression model is Mean Squared Error (MSE). It takes the values of the actual target labels (those present in the testing dataset) and the target labels predicted by the model as input. MSE is computed as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where $N$ is the number of data points,
$f_i$ the value returned by the model and
$y_i$ the actual value for data point $i$.

The MSE values of the model when trained using different ML algorithms are compared and the best one i.e. most nearer to zero is selected.

| Text Model | | |
|---|---|---|
| **Model Type** | **Post text MSE** | **Comment Text MSE** |
| **K-Fold Cross Validation on Ridge Regression** | 0.000864 ± 0.000158 | 0.000864 ± 0.000071 |
| **Ridge Regression** | 0.00095547 | 0.000981727 |
| **Lasso Regression** | 0.0009554 | 0.000981701 |
| **Random Forest Regressor** | 0.001100625 | 0.001362798 |
| **MultiLinear Regression** | 0.187740545 | 0.147041598 |
| **Polynomial Regression** | 0.228348583 | 0.253920377 |

| Image Model | | |
|---|---|---|
| **Model Type** | **Post face images MSE** | **Profile face images MSE** |
| CNN | 0.00087 | 0.039 |
| CNN with Synthetic data | 0.00073 | 0.0023 |
| CNN with Vgg16 (transfer Learning) | 0.0023 | 0.0021 |

| Cognitive Score Model | |
|---|---|
| **Model Type** | **Cognitive scores MSE** |
| Support Vector Regression (SVR) | 0.009234669 |
| Ridge Regression | 0.015347428 |
| Lasso Regression | 0.015563659 |
| MultiLinear Regression | 0.017522713 |
| Random Forest | 0.020050536 |
| Polynomial Regression | 0.033331121 |
| K-Fold Cross Validation on Ridge Regression | 0.039512 ± 0.019227 |

# 10 Conclusion

## 10.1 Project Benefits

This project holds immense usefulness across diverse fields:

- Personal Growth and Development: Individuals can gain self-awareness, identify strengths, and work on areas for improvement, leading to personal growth and better life decisions.
- Interpersonal Relationships: Better understanding of one's own and others traits enhances communication, conflict resolution, and relationship-building skills.
- Organizational Productivity: Businesses can optimize team dynamics, job-role fit, and leadership development for enhanced productivity and employee satisfaction.
- Education: Educational institutions can adapt teaching methods to students learning styles, leading to improved engagement and academic outcomes.

## 10.2 Future Scope for improvements

- Real-time Analysis: Develop tools that provide real-time personality traits analysis, adapting to individuals changing behaviors and life circumstances.
- Platform Independent: Make the system capable of working on Android and iOS.

# References

Azucar, D., Marengo, D. & Settanni, M. (2018), 'Predicting the big 5 personality traits from digital footprints on social media: A meta-analysis', *Personality and individual differences* **124**, 150–159.

Golbeck, J., Robles, C., Edmondson, M. & Turner, K. (2011), Predicting personality from twitter, *in* '2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing', IEEE, pp. 149–156.

Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H. & Eichstaedt, J. C. (2017), 'Detecting depression and mental illness on social media: an integrative review', *Current Opinion in Behavioral Sciences* **18**, 43–49.

Jin, S. V., Muqaddam, A. & Ryu, E. (2019), 'Instafamous and social media influencer marketing', *Marketing Intelligence & Planning* **37**(5), 567–579.

Kavakci, E. & Kraeplin, C. R. (2017), 'Religious beings in fashionable bodies: the online identity construction of hijabi social media personalities', *Media, Culture & Society* **39**(6), 850–868.

Li, C., Wan, J. & Wang, B. (2017), Personality prediction of social network users, *in* '2017 16th international symposium on distributed computing and applications to business, engineering and science (DCABES)', IEEE, pp. 84–87.

Liu, L., Preotiuc-Pietro, D., Samani, Z. R., Moghaddam, M. E. & Ungar, L. (2016), Analyzing personality through social media profile picture choice, *in* 'Proceedings of the International AAAI Conference on Web and Social Media', Vol. 10, pp. 211–220.

Park, G., Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Kosinski, M., Stillwell, D. J., Ungar, L. H. & Seligman, M. E. (2015), 'Automatic personality assessment through social media language.', *Journal of personality and social psychology* **108**(6), 934.

Praphulla, G., Kishore, I. B., Venkatesh, B., Praveen, B. & Rao, P. S. (2023), 'Personality prediction using machine learning techniques', **2794**(1).

Pratama, B. Y. & Sarno, R. (2015), Personality classification based on twitter text using naive bayes, knn and svm, *in* '2015 international conference on data and software engineering (ICoDSE)', IEEE, pp. 170–174.

Seidman, G. (2013), 'Self-presentation and belonging on facebook: How personality influences social media use and motivations', *Personality and individual differences* **54**(3), 402–407.

Sorokowska, A., Oleszkiewicz, A., Frackowiak, T., Pisanski, K., Chmiel, A. & Sorokowski, P. (2016), 'Selfies and personality: Who posts self-portrait photographs?', *Personality and Individual Differences* **90**, 119–123.

Valanarasu, R. (2021), 'Comparative analysis for personality prediction by digital footprints in social media', *Journal of Information Technology and Digital World* **3**(2), 77–91.

Walsh, P., Clavio, G., Lovell, M. D. & Blaszka, M. (2013), 'Differences in event brand personality between social media users and non-users.', *Sport Marketing Quarterly* **22**(4).

Wan, D., Zhang, C., Wu, M. & An, Z. (2014), Personality prediction based on all characters of user social media information, *in* 'Social Media Processing: Third National Conference, SMP 2014, Beijing, China, November 1-2, 2014. Proceedings', Springer, pp. 220–230.

# APPENDIX

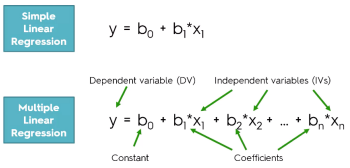
– **Multiple Linear Regression**

Linear regression predicts the relationship between two variables by assuming a linear connection between the independent and dependent variables. It seeks the optimal line that minimizes the sum of squared differences between predicted and actual values. The goal of the linear regression algorithm is to get the best values for b0 and b1 to find the best-fit line. To calculate best-fit line linear regression uses a traditional slope-intercept form which is given below:

$$Y_i = b_0 + b_1 X_i$$

Where,
$Y_i$ = Dependent variable,
$b_0$ = constant/Intercept,
$b_1$ = Slope/Intercept,
$X_i$ = Independent variable.

It can extend to multiple linear regression involving several independent variables as in our case since more than one textual features are extracted from the users' posts. Each of these act as an independent variable.

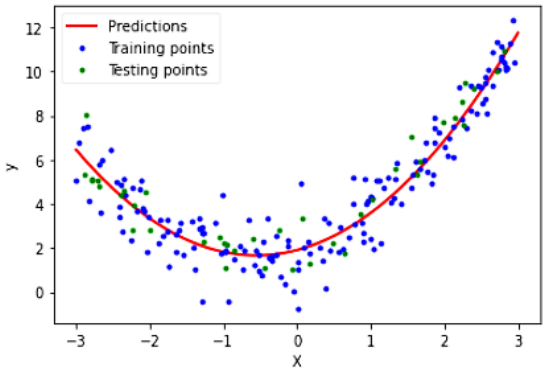



– **Polynomial Regression**

It is a form of Linear regression where only due to the Non-linear relationship between dependent and independent variable, some polynomial terms are added to linear regression to convert it into Polynomial regression. The equation of polynomials becomes something like this:

$$y = a_0 + a_1x_1 + a_2x_1^2 + \ldots + a_nx_1^n$$

The relationship between the dependent variable and the independent variable is modelled as a nth-degree polynomial function. When the polynomial is of degree 2, it is called a quadratic model; when the degree of a polynomial is 3, it is called a cubic model, so on.




These models are usually fitted with the method of least squares.

– **Ridge Regression**

Ridge regression, also known as Tikhonov regularization, adds a penalty equal to the square of the magnitude of the coefficients to the loss function of ordinary least squares (OLS) regression. This technique can be used to reduce the effects of multicollinearity in ridge regression, which may result from high correlations among predictors or between predictors and independent variables.The cost function for ridge regression is given by:

$$RSS_{L2} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \lambda \sum_{j=1}^{P} B_j^2$$

Where,

$y_i$ are the actual values,

$y_i\hat{}$ are the observed values,

$\beta_i$ are the coefficients,

$\lambda$ is the regularization parameter.

– **Lasso Regression**

Lasso regression, short for Least Absolute Shrinkage and Selection Operator, is another type of linear regression that incorporates a regularization term to prevent overfitting and encourage sparsity in the model. It adds a penalty term to the ordinary least squares (OLS) loss function. This penalty term is the absolute value of the coefficients multiplied by a regularization parameter.

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\left|\hat{\beta}_j\right|$$

Where,

$y_i$ are the actual values,

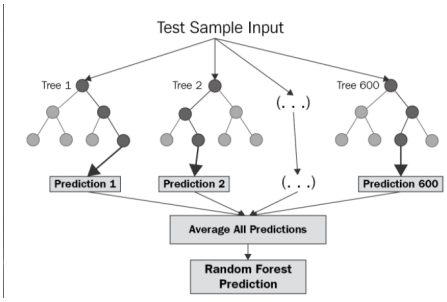
$y_i\hat{}$ are the observed values,

$\beta_i$ are the coefficients,

$\lambda$ is the regularization parameter.

– **Random Forest Regression**

It is a machine learning algorithm that utilizes an ensemble of decision trees to improve predictive performance and control overfitting. It introduces an extra layer of randomness by selecting a random subset of features at each split in the decision tree leading to a more robust model. averages the predictions of multiple decision trees to produce a final prediction. This helps in improving the generalization ability of the model.

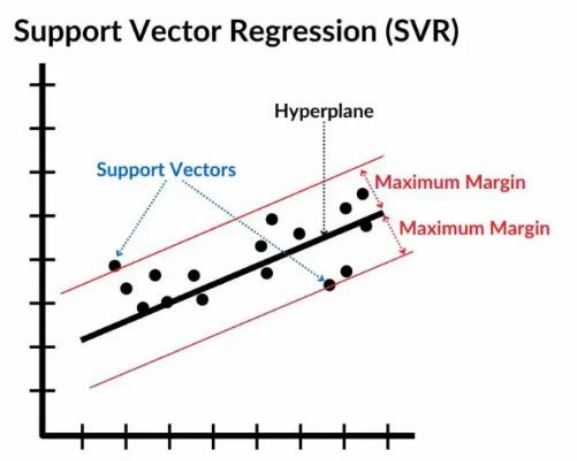


**Algorithm Steps:**

- **Bootstrap Sampling:** Randomly sample the training data with replacement to create multiple datasets.
- **Train Decision Trees:** For each sampled dataset, train a decision tree. At each node, use a random subset of features to determine the split.
- **Aggregate Predictions:** For a new data point, pass it through each of the trained decision trees and average their predictions to obtain the final prediction.

– **SVR (Support Vector Regression)**

Support Vector Regression (SVR) with a linear kernel is a powerful method for handling linear regression problems. It aims to find a linear function that approximates the relationship between the input features and the target values while maintaining a margin of tolerance (epsilon) where deviations are not penalized. The linear kernel simplifies the model by assuming a linear relationship between the input variables and the target, making it computationally efficient and effective for datasets where the underlying relationship is indeed linear. By optimizing the regularization parameter (C), SVR balances the trade-off between minimizing errors and avoiding overfitting, resulting in a robust linear regression model.



– **CNN (Convolutional Neural Network):**

This model is used for various tasks such as image preprocessing, feature extraction, and model training for personality prediction. It plays a crucial role in analyzing and processing image data collected from social media platforms.

– **CNN with Synthetic Data:**

Synthetic data generation using CNNs is employed to augment the training dataset for improving model performance. By creating artificial images resembling those from social media platforms, this approach enhances the diversity and quantity of data available for training the CNN model.

– **CNN with VGG16 (Transfer Learning):**

Utilizing VGG16 as a pre-trained CNN model allows leveraging its learned features for image understanding. Transfer learning with VGG16 involves fine-tuning the model on your specific dataset, adapting it to extract relevant features from social media images for personality trait prediction. This approach benefits from the already learned features in VGG16, enhancing the efficiency and effectiveness of the model training process.