



Trinity College Dublin

Coláiste na Tríonóide, Baile Átha Cliath

The University of Dublin

School of Computer Science and Statistics

Investigating Relation Between Five Factor Personality Traits and Facial Action Units

Ritika Sharma

Supervisor: Dr Carl Vogel

August 19, 2022

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Masters of Computer Science

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Ritika Sharma

August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Ritika Sharma

August 19, 2022

Investigating Relation Between Five Factor Personality Traits and Facial Action Units

Ritika Sharma, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Dr. Carl Vogel

Since the invention of Artificial Intelligence, researchers have aimed to create machines that can replicate human intelligence, which is not just confined to mathematics or analytical but is also emotionally aware and intelligent. Having emotionally aware systems is especially beneficial for Human-Computer-Interaction as they aim to enhance the user experience of using computers. Personality Computing is an example of such an effort made, based on the belief that the way users use technology relies heavily on their personalities.

This research is based on the branch of Personality Computing -Automatic Personality Recognition. It aims to find if there exists any relationship between Five-Factor Personality Traits and Facial Action Units and also if the above-mentioned relationship varies gender-wise. The Five Factor personality traits consist of Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism.

The facial action units of an individual are collected from the video files present in the MULTISIMO dataset. The Big Five Inventory (BFI, 44 items) generate the personality trait scores. The video files are processed through the PyFeat library to extract facial action unit features. Feature Engineering is applied to the dataset to make it more resilient before passing to different classifiers, decision trees and random forests. Random Forest generated the best accuracy out of the two. The data was non-parametric; thus, the Mann-Whitney U test was applied to verify the results. The results produced are promising. There is a clear association between each five personality traits and the 20 different action units considered in this research.

The approach used in this research can be applied to any real-time dataset to identify an individual's personality traits and leverage it to make the system adapt according to the user's personality traits or use it in career development applications.

Acknowledgements

Special thanks to my supervisor, Dr. Carl Vogel, for showing faith in my idea and taking me under his supervision to take the first steps towards achieving my goal, for guiding me along the way with utter patience. This work would not have reached its completion without his unconditional support.

I would also like to thank my second reader, Dr. Tim Fernando, for taking time out for my demonstration and showing an interest in my work.

I have the deepest gratitude towards my parents, Mr. Anil Sharma and Annu Sharma, and brother, Arpit for believing in me, financing me and sending me miles away from them to letting me fulfil my dreams. This work would not have been possible without them—a special remembrance to my Mumma and Tuntun. I wish you were with me to see this.

Also, to my Trinity friends - Yash, Faheim, Bhushan, Tolga and Oliver, you made my life easy in Dublin and helped me stay motivated, proofread my thesis in the weekly meetings. To Rajat and Karan, who are not part of the course but helped me throughout the process.

Last but not least, I would like to thank myself because I finally made it one step closer to where I always wanted to be.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Contributions	3
1.4	Dissertation Overview	3
2	Background and Literature Review	5
2.1	Personality Computing	5
2.2	Personality Assessment	7
2.2.1	Personality Assessment Methods	8
2.2.2	Personality Assessment Questionnaire	8
2.3	Five Factor Model	9
2.3.1	Openness to Experience	10
2.3.2	Conscientiousness	12
2.3.3	Extraversion	12
2.3.4	Agreeableness	13
2.3.5	Neuroticism	13
2.4	Are Facial Action Units and Personality Traits related	13
2.5	Conclusion	16
3	Methodology	17
3.1	Data	17
3.1.1	Corpus Overview	17
3.1.2	Participants - Players and Facilitators	18
3.1.3	Corpus Experimental Setup	18
3.1.4	Corpus Technical Setup	19
3.1.5	Personality Traits Assessment	19
3.2	PyFeat	20
3.2.1	Detector	20
3.2.2	Fex	21

3.3	Machine Learning Pipeline	22
3.3.1	Supervised Learning	22
3.3.2	Data Collection	23
3.3.3	Feature Engineering	24
3.4	Machine Learning Algorithms	27
3.4.1	Decision Trees	27
3.4.2	Random Forest	28
3.4.3	Variable Importance	29
3.5	Evaluation Methods	31
3.5.1	Mann Whitney U Test	31
3.5.2	Performance Measures	32
3.6	Conclusion	33
4	Implementation	35
4.1	Data Preparation	35
4.2	Feature Engineering	39
4.2.1	Imputation	39
4.2.2	Binning or Discretization	40
4.3	Machine Learning Models	43
4.3.1	Decision Trees	45
4.3.2	Random Forest	46
4.4	Conclusion	47
5	Evaluation	48
5.1	Random Forest	48
5.1.1	Openness	48
5.1.2	Conscientiousness	50
5.1.3	Extraversion	53
5.1.4	Agreeableness	55
5.1.5	Neuroticism	56
5.2	Decision Trees	57
5.2.1	Openness	59
5.2.2	Conscientiousness	60
5.2.3	Extraversion	61
5.2.4	Agreeableness	64
5.2.5	Neuroticism	64
5.3	Conclusion	67
6	Conclusion	68
6.1	Discussions	68

6.1.1	Openness	68
6.1.2	Conscientiousness	69
6.1.3	Extraversion	69
6.1.4	Agreeableness	69
6.1.5	Neuroticism	69
6.1.6	Research Finding	70
6.2	Challenges	71
6.3	Future Work	72

List of Figures

2.1	Different Personality Assessment Methods [Ass, 2016]	8
2.2	Five Factor Model [Sheerin, 2012]	11
2.3	Facial Action Units [Ko and Sim, 2010]	15
3.1	Machine Learning Pipeline [Lazzeri, 2019]	23
3.2	Random Forest Algorithm [Brandon Greenwell, 2020]	29
3.3	Confusion Matrix [Suresh, 2021]	32
4.1	Data snippet before Feature Engineering	39
4.2	BFI-44 data before preprocessing	39
4.3	Final Dataset Snippet	43
4.4	Skewed data distribution of action units	44
5.1	Error vs trees for Openness - Random Forest	49
5.2	Variable Importance Openness - Random Forest	51
5.3	Dependency of Gender on Openness	51
5.4	Dependency of AU01 on Openness	52
5.5	Variable Importance Conscientiousness - Random Forest	52
5.6	error vs trees for Extraversion - Random Forest	53
5.7	Variable Importance Extraversion - Random Forest	54
5.8	Variable Importance Agreeableness - Random Forest	56
5.9	Error vs trees for Neuroticism - Random Forest	57
5.10	Variable Importance Neuroticism - Random Forest	58
5.11	Cp vs error - Openness	59
5.12	Decision Tree - Openness	60
5.13	Cp vs error - Conscientiousness	61
5.14	Decision Tree - Conscientiousness	62
5.15	Decision Tree - Extraversion	63
5.16	Cp vs error - Agreeableness	64
5.17	Decision Tree - Agreeableness	65
5.18	Cp vs error - Neuroticism	65

5.19 Decision Tree - Neuroticism	66
--	----

List of Tables

3.1	Questions asked to all the participants	19
4.1	Gender distribution in original and train-test dataset	44
5.1	m_{try} values for Openness RF model	49
5.2	Confusion matrix for default and hypertuned RF model - Openness	50
5.3	Openness RF model Accuracy	50
5.4	m_{try} values for Conscientiousness RF model	51
5.5	Confusion Matrix for Extraversion Random Forest Default Model	53
5.6	m_{try} values for Extraversion RF model	54
5.7	Extraversion RF model Accuracy	54
5.8	m_{try} values for Agreeableness RF model	55
5.9	Agreeableness RF Model Accuracy	55
5.10	Confusion matrix for default and hypertuned RF model - Agreeableness	55
5.11	m_{try} values for Neuroticism RF model	56
5.12	Neuroticism RF Model Accuracy	57
5.13	Confusion Matrix - Neuroticism RF Model	58
5.14	Openness Decision Tree Accuracy	59
5.15	Conscientiousness Decision Tree Accuracy	60
5.16	Extraversion Decision Tree Accuracy	61
5.17	Agreeableness Decision Tree Accuracy	64
5.18	Neuroticism Decision Tree Accuracy	66
6.1	Action Units vs Openness - Mann Whitney U test	68
6.2	Action Units vs Conscientiousness - Mann Whitney U test	69
6.3	Action Units vs Extraversion - Mann Whitney U test	69
6.4	Action Units vs Agreeableness - Mann Whitney U test	70
6.5	Action Units vs Neuroticism - Mann Whitney U test	70
6.6	Actions Units and Associated Personality traits	71

1 Introduction

"The face is the mirror of the mind, and eyes without speaking confess the secrets of the heart." - St. Jerome

Famous sayings like the one above make a person wonder if these are just proverbs or aphorisms. This thesis is a segment of the series of studies done to discover the truthness and relevance of these quotes.

The introduction chapter first briefly discusses the motivation behind pursuing this line of research and the research problem. Then, it gives an overview of the layout of the remainder of the dissertation and the synopsis of the specific objectives attained in those respective chapters. Furthermore, a discussion of the challenges experienced during the research process is also present.

1.1 Motivation

Humans are curious creatures and tend to classify or relate unfamiliar things or people into their knowledgeable domains. According to the studies done in the field of social cognition, people spontaneously and involuntarily make social inferences (precisely characteristics, personality traits, and causes) of any person they meet [Uleman et al., 2008]. The personality traits that make us distinct from each other, even though being from the same species, can be explained through personality psychology. It deals with understanding stable individual qualities, usually quantifiable measures, that comprehend and predict observable behavioural actions of people.

The topic gained attention and has seen a drastic increase in the number of relevant research. Endeavours have been made to merge attributes of personality psychology with technology since the early 21st century. Research like [Guadagno et al., 2008, Butt and Phillips, 2008, Yeo, 2010, Qiu et al., 2012] explores the relationship between personality and computing by analysing the link between traits and technology usage. The main idea behind this type of research is that the way users use technology depends on their personality. Therefore, the personality of users can predict how they will act. The interest in this subject

rose due to three primary events in the technological landscape, leading to pioneer strategies designed to integrate personality psychology with human-computer interaction. The first is the rise in the availability of personal data, most of which is self-disclosing in nature [Kaplan and Haenlein, 2010] and can be gathered from social networking websites [Rainie and Wellman, 2012]. The second is the potential for using mobile technologies, particularly cellphones, to gather regular, spontaneous, beautifully behavioural evidence [Raento et al., 2009]. The third is an effort to give computers social and affective awareness to interact with people as people do [Vinciarelli et al., 2012].

All personality computing-related tasks can be broadly classified into three main problem areas - 1) Recognising the personality of a person (Automatic Personality Recognition or APR). 2) Predicting an individual's personality characteristics as perceived by others (Automatic Personality Perception or APP). 3) Creating artificial agents with human-like personalities (Automatic Personality Synthesis or APS) [Vinciarelli and Mohammadi, 2014]. Existing methods rely on specific questionnaires, which are time-consuming and cannot be repeated too frequently. A system capable of determining the personality qualities of a subject merely by studying their facial features will be able to monitor a subject's personality traits in real time. This research could benefit not only the field of human-computer interaction but also talent management, team collaboration and performance, determining personality shifts related to psychiatric disorders, personalised health assistance, and diagnosing physical diseases with personality trait changes as symptoms.

Automatic Personality Recognition is the main element of this research. The research question is to find if it is possible to predict an individual's personality traits through a series of experiments if their facial action units are being recorded.

1.2 Objectives

The major objectives of this research are -

- Recognise if personality traits can be associated with certain (or a combination of) facial action units.
- If yes, then predict personality traits through recorded facial action units of people engaged in a task
- Identify which machine learning classifier can predict personality traits if action units intensity values are given
- Analyse if gender has any role to be played in automatic personality recognition

While the null hypothesis of this research is -

There exists an association between personality traits and facial action units of an individual, thus making it possible to determine personality traits of a person if their facial action unit data is present.

1.3 Contributions

This work uses video files available in the MULTISIMO corpus and processes those video files to extract information like - facial landmarks, facial detection, action unit intensities and emotions corresponding to each second frame of the video. It is generated for each participant from the 18 sessions available on the MULTISIMO website. This novel dataset can be used in future work related to the corpus.

This research is also one of its kind on the MULTISIMO corpus that tries to find an association between the visuals and personality traits.

1.4 Dissertation Overview

1. Chapter 1 - Introduction

This chapter briefly discusses the motivation of the research problem and gives an overview of the layout of the dissertation.

2. Chapter 2 - Literature Review

This chapter thoroughly explains the fundamentals of Automatic Personality Recognition and what to consider to design an automatic personality recognition system. The chapter also discusses the basics of facial action units and details of personality psychology to set a foundation for the work done in this research. Later, the chapter reviews the literature on the work done in identifying the relation between facial action units and a person's personality traits.

3. Chapter 3 - Methodology

This chapter commences by discussing the data set requirements for the experiment. The source and details of the obtained data set are then explained, as well as how the obtained video data is processed into a dataset containing specific facial information to make it usable for project requirements. Finally, the chapter explains the different machine learning algorithms used to detect if any facial action units (AUs) are a predictor of the FIVE Factor Personality traits in the dataset collected.

4. Chapter 4 - Implementation

This chapter pans out the technical aspect of this research. It explains the code

written for this research, and the libraries or arguments passed to the library are mentioned clearly. This is done to make sure easy replication of this research is possible in the future. This also explains how machine learning models are built in this experiment.

5. Chapter 5 - Evaluation

This chapter evaluates the results generated by different machine learning models and their accuracy.

6. Chapter 6 - Conclusion

This chapter details the final findings of this research, the rejection or acceptance of the research hypothesis and any challenges faced or future work required to improve the results obtained.

2 Background and Literature Review

The Background and Literature Review chapter addresses prior studies done in Personality Computing centric on facial behaviours. Computer scientists have been conducting substantial research on personality computing in the last several decades to further bridge the social competence gap between humans and machines. In order to build a model for automatic personality recognition, some researchers have created new feature sets. In contrast, others proposed techniques and technologies to extract factors from individuals' behaviours, facial expressions, body movements, or speech. Furthermore, computational linguistics, psychology, and statistics researchers have started focusing on the relation between facial expression or emotions and personality detection and the broader topic of affective computing over the last decade.

This section will provide a deeper insight into personality computing, along with the work done to automate the task of personality recognition, the different models designed to categorize and recognize personality traits, and a critical review of the previous and pioneer work done in the field of detecting personality traits through facial expressions.

2.1 Personality Computing

The fundamental premise of personality psychology is that consistent individual traits lead to consistent behavioural patterns frequently exhibited by people, at least to some extent, regardless of the circumstance. Distinguishing the internal characteristics of the individual from overt behaviours and examining the causal linkages between them is one of the fundamental objectives of personality psychology [Matthews et al., 2009]. Aspects of life such as "happiness, physical and psychological health, quality of relationships with peers, family, and romantic others, career status, fulfilment, and success, community engagement, criminal activity, and political view" are successfully predicted by current personality models [Funder, 2001], in addition to "patterns of thought, emotion, and behaviour" [Ozer and Benet-Martínez, 2006]. Furthermore, how others perceive a person's personality significantly influences their attitude and social conduct toward them [Uleman et al., 2008]. This effectiveness by which an individual's personality traits can be captured is the main reason

behind the rising interest in assessing personality through technology, also known as personality computing. Combining artificial intelligence with personality psychology is the central vision of this topic. It is primarily concerned with three fundamental issues: automatic personality trait recognition, perception, and synthesis. The first aims to accurately identify or predict an individual's actual (self-assessed) personality. This method enables the creation of an evident personality (or first impressions) of a stranger. The study of automatic personality perception focuses on the various subjective factors that influence a person's personality perception. Finally, automatic personality trait synthesis aims to create artificial personalities using artificial agents and robots. This research solely focuses on Automatic Personality Recognition and is further discussed below.

According to the Lens Model [Brunswik, 2020], people tend to reflect their personality through subtle distal cues, i.e., any behaviour others can observe. Thus, even though personality is an abstract concept that cannot be seen directly, it leaves indicators in almost everything people do that can be seen [Scherer, 1979].

Automatic Personality Recognition (APR) is the process of deducing *self-assessed* individual personalities from computer-traceable distal cues by utilizing the concept of externalization of personality by users. Distal cues can comprise verbal and non-verbal behaviours. Non-verbal cues can be easily collected from the machines, and in principle, APR can perform personality predictions using them. Numerous works employ this idea and use non-verbal cues such as non-verbal communication [Mairesse et al., 2007, Ivanov et al., 2011], body movements, and multi-modal combinations extracted from speech (like intonation, pauses, pitch, laughter) [Pianesi et al., 2008, Batrinca et al., 2012] use of technology and interpersonal distances [Zen et al., 2010]. In most instances, APR techniques utilize methodology characteristic of Affective Computing, Social Signal Processing, sociolinguistics, adaptive applications, and other disciplines that capture users' social or emotional behaviour through machines (EEG sensors, mobiles, cameras). Co-variation of the distal cues captured and the personality traits are computed, also called Ecological Validity of the cues, to identify the distal cues that can lead to high APR performance [Vinciarelli and Mohammadi, 2014].

Automatic Personality Recognition works like any other Machine Learning problem statement. A research question is hypothesized first. Relevant data (in this case, distal cues) is collected and pre-processed; in the third stage, deep learning or machine learning techniques are used to attain the best prediction results. Hence, APR systems are classified into two categories based on the type of input data collected -

- 1) Single Modality - In this type of data, a single type of modality is extracted (eg - audio/ video/ text) for personality recognition
- 2) Multiple Modality - In this type, multiple modalities are merged (e.g., audio, text and visuals) in various ways for personality recognition.

Multiple modalities can be integrated into three ways: feature-level fusion, model-level fusion, and decision-level fusion [Zeng et al., 2009, Atrey et al., 2010]. This research uses a multi-modal dataset (audio and visual), and the type of fusion used for this research is model-level (all the modalities are assessed separately while also considering any correlation between the modalities) since this research focuses exclusively on visuals. More about the dataset is described in the section 3.1.

The APR relies on the traditional personality self-assessment methods for collecting the labelled data and assigning individuals' personality traits/types. These Personality models and approaches to assess them are discussed in the below section.

2.2 Personality Assessment

Researchers in personality psychology constantly try to figure out ways to capture or sort the differences present in individuals and the origins of these differences. Some theories suggest a biological outlook or cognitive aspect. In contrast, others suggest considering the environment or the inner states of a human being to identify what factors can classify all the variations in the identity of human beings [Funder, 2001]. But the most widely acknowledged construct is *traits* [Deary, 2009]. Thus, personality assessment can be done by characterizing a few generalized traits that can define the characteristics of people. Psychologists aim to find wide dimensions that successfully capture all the characteristics displayed by humans and are classified as *personality traits*. Although there is much debate on the validity of results obtained through traits, researchers argue they are hypothetical and do not predict the actual characteristics of an individual [Cloninger, 2009]. Nevertheless, years of experiments in a wide range of cultures and situations have validated the availability of these traits [Deary, 2009]. Over the years, researchers have devised multiple trait-based models to assess an individual's personality, such as 16-factor personality traits, Big Five Model, Five Factor Model, OPQ and Birkman.

The other sector of personality assessment model are "*type-based*" for example - Myers-Briggs Type Indicator (MBTI) [Boyle, 1995], FIRO-B [Schutz, 1992], Belbin Personality type [Aritzeta et al., 2007], DISC [Lykourantzou et al., 2016]. The main difference between type theory and trait theory is that type theory sees people's traits as discrete groups, while trait theory sees these traits as parts of a larger continuum. For example, a type theorist would say that extroverts and introverts are two different types of people. However, according to trait theorists, people fall between the range of introversion and extroversion [Psy].

2.2.1 Personality Assessment Methods

There are broadly four categories to assess an individual's personality, and each category contains various tools [Andrews, 1948]. The four categories are -

- 1) subjective methods - an individual is assessed from the first person's perspective through a variety of ways like personality inventories, interviews or writing their autobiography.
- 2) objective methods - an individual is assessed on how others perceive his/ her personality
- 3) Projective methods - the individual is encouraged to 'project' their ideas, feelings, wishes, and other reactions in offered settings. These approaches show the underlying qualities, moods, attitudes, and imaginations that govern an individual's behaviour.
- 4) Psycho-analytical methods - In this approach, an expert puts the individual in their subconscious state, helping the expert discover the individual's actual unmasked characteristics.

A summary is depicted in Fig 2.1. This research uses subjective methods, specifically personality inventories, as they provide self-assessment.

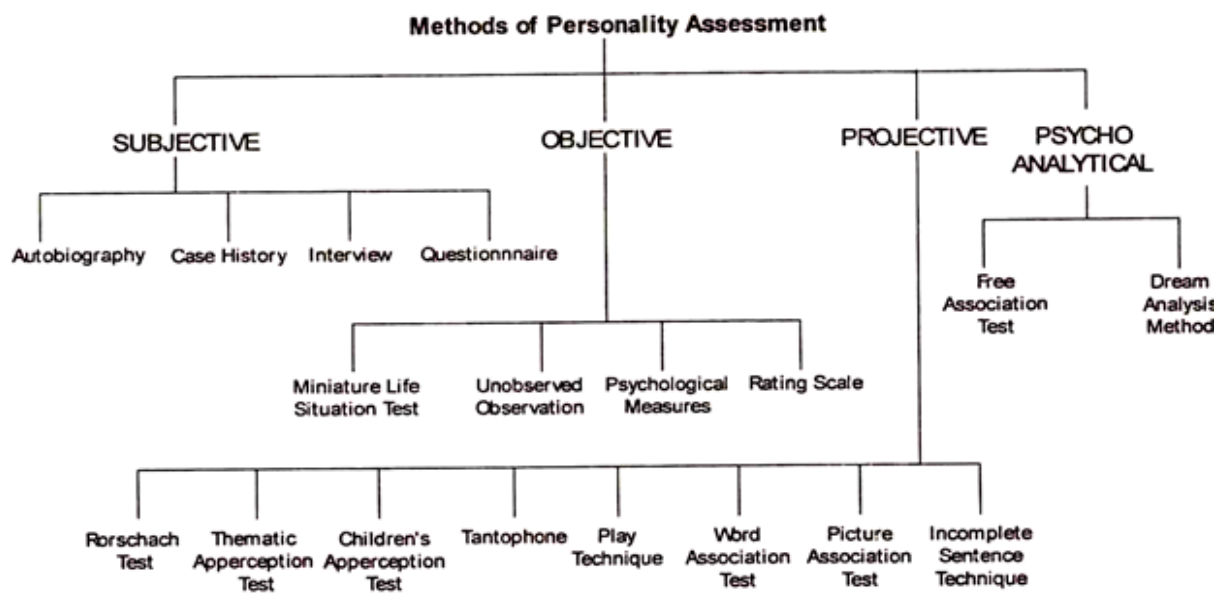


Figure 2.1: Different Personality Assessment Methods [Ass, 2016]

2.2.2 Personality Assessment Questionnaire

A personality assessment questionnaire or inventory is a document containing a list of questions with available options ranging from Strongly Disagree to Strongly Agree and is used to measure the intensity of personality traits exhibited by an individual. The most selected option helps predict the assessor if the individual has definite traits or not. The questions or phrases reflect the assessed individual's personality traits, feelings, attitudes, or

behaviours. One obstacle with this method is that the individual might not be interested in giving correct information about themselves. This assessment method is widely accepted and proven to yield great results [Boyle and Helmes, 2009]. There are two types of assessment questionnaires available -

- 1) Self-questionnaires are used for self-assessment (personality recognition). The individual being assessed fills out a questionnaire to reveal their true personality. They describe what they perceive about themselves. Are they social, lazy or focused.
- 2) Third-person questionnaires give informant assessments (personality perception). A set of assessors fill out a third-person questionnaire for the individual being assessed, rating them based on how they perceive the individual's personality. The questions are similar to first-person but changed to "Him/Her" rather than "I". These assessors must fill out questionnaires for all the participants of an experiment. The average rating is considered the personality rating of that individual.

Career professionals heavily use first-person questionnaires to predict the personality of job applicants. Although, chances are individuals might not opt for any negative answer since they do not want to create any negative impression and try to choose options that project a desirable personality. Nevertheless, substantial experiments on this topic showed a similarity between the results obtained by self-assessments and those given by others for that individual. Thus, proving the credibility of self-assessment questionnaires [Rammstedt and John, 2007]. The reliability of self-questionnaires carves the way towards the research of Automatic Personality Recognition, which is the main essence of this thesis.

Few commonly used inventories are - Neo-Personality-Inventory Revised (NEO-PI-R, 240 items) [Jr. and McCrae, 1995], Big-Five Inventory (BFI, 44 items) [John et al., 1991], BFI-10 (shorter version of BFI) [Rammstedt and John, 2007], Neo Five-Factor Inventory (NEO-FFI, 60 items) [McCrae and Costa, 2004]. This research uses BFI-44.

2.3 Five Factor Model

As mentioned, among the multiple ways to assess an individual's personality, one of the most famous is - The Five-Factor Model (FFM). The five-factor model (FFM) is a collection of five personality characteristics commonly known as Big Five Personality Traits: Extraversion, Conscientiousness, Openness to Experience, Agreeableness and Neuroticism, acronym as OCEAN. The FFM is one of the most extensively used models for personality structure [McCrae, 2020]. According to FFM research, each of the Five dimensions evolves according to environmental and biological stimuli throughout an individual's lifespan. Research such as one presented in [Roberts and Mroczek, 2008] demonstrates that compared to other approaches for assessing personality characteristics, FMM provided a greater degree of stability into adulthood, with the stability peak occurring four years after

the individual begins working. FMM has been utilized effectively in professional development and counselling, job cohesion, and training competence and has contributed to enhancing academic and learning performance [Komarraju et al., 2011]. In addition, FMM has demonstrated proficiency in identifying personality disorders, such as anxiety, depression, and substance abuse, and has been used as an indicator for various physical diseases, including cancer, hypertension, cardiovascular disease, respiratory diseases, stroke, and diabetes [Jokela et al., 2014]. Thus Five Factor can be employed to predict a wide range of significant social, vocational, and psychological outcomes [Soto et al., 2015]. Experiments were done to introduce more dimensions to the model, but the five-factor model proved more stable than those [Digman, 1996]. In contrast, experiments that tried to deduct a few traits in their model proved linear combinations of the five personality traits [Eysenck, 1991]. Five-Factor Model is commonly used for Automatic Personality Recognition. Therefore, it is used in this research as well. The Personality traits determined by the APR system are OCEAN, and the distal cue used is visuals collected from the multi-modal dataset.

Two theories can back up the concept of Five Factor traits. 1) One is the research done in phonetic personality and attempts to classify all the personality-related adjectives used in various languages into a broader range of personality dimensions [Goldberg, 1993], leading to the invention of the Big Five Model (BFM) Personality Traits. 2) The Five Factor Model was invented to shorten the personality questionnaires. Moreover, most personality traits found by other inventories can be easily classified into Big Five traits. [John et al., 2008]. However, now, FFM and BFM are used interchangeably.

There is a detailed description of the personality traits and what adjectives can describe the traits in respective subsections [Soto et al., 2015]. Fig 2.2 depicts how personality traits and their presence influence an individual's behaviours.

2.3.1 Openness to Experience

Openness to Experience signifies an individual's artistic, intellectual, and experiential depth. Essential components of Openness comprise aesthetic sensibility (as opposed to insensitivity), imagination (as opposed to a lack of creativity), and intelligence (as opposed to a lack of intellectual curiosity). Individuals with a high level of Openness are likely to have a broad spectrum of hobbies and love learning or exploring new things. In contrast, those with a low level of Openness typically have fewer interests and value routine and familiarity above novelty and variation. Nevertheless, there are inevitable disagreements with the definition of Openness. In BFM, Openness is called Intellect, as intelligence is included along with intellectual curiosity and interests as a component of this trait. Compared to their less receptive peers, those who are more open-minded tend to score higher on Intellect and creative exams and spend more time in school. They excel and are predisposed toward

Personality Trait	Low Scorer	High Scorer
Openness	Favours conservative values Judges in conventional terms Is uncomfortable with complexities Moralistic	Values intellectual matters Rebellious, non-conforming Has an unusual thought process Introspective
Conscientiousness	Unable to deny gratification Self-indulgent Engages in daydreams	Behaves ethically Dependable, responsible Productive Has high aspiration level
Extraversion	Emotionally bland Avoids close relationships Over-control of impulses Submissive	Talkative Gregarious Socially poised Behaves assertively
Agreeableness	Critical, skeptical Behaviour is condescending Tries to push limits Expresses hostility directly	Sympathetic, considerate Warm, compassionate Likeable Behaves in a giving way
Neuroticism	Calm, relaxed Satisfied with self Clear-cut personality Prides self on objectivity	Thin-skinned Anxious Irritable Guilt-prone

Source: McCrae and Costa (2003: 53).

Figure 2.2: Five Factor Model [Sheerin, 2012]

creative, analytical, and technological fields. Compared with their less open colleagues, they are likely to indulge in drug use, define themselves as spiritual (although not exclusively religious), and have liberal political and social opinions.

2.3.2 Conscientiousness

The aptitude of an individual to arrange things, perform activities, and strive toward long-term objectives is characterized by Conscientiousness. Key characteristics include orderliness (as opposed to disorder), self-discipline (as opposed to inefficiency), and dependability (vs inconsistency). Individuals with high Conscientiousness, like structure and order, are effective workers, tend to adhere to rules and conventions, and are better able to defer pleasure. In contrast, those with low Conscientiousness struggle to control their impulses and are quickly diverted from their jobs. Out of the Five Factor Personality traits, Conscientiousness is the highest predictor of combined academic and vocational success. Conscientious students obtain comparatively higher marks, and conscientious employees do better in various jobs. In contrast, those with low conscientiousness people engage in unproductive work activities. In addition, Conscientiousness is a significant, strong indicator of physical health, mental well-being and lifespan. The connections of high Conscientiousness with general health extend to many health-related practices, such as a better diet, regular exercise, less consumption of cigarettes, alcohol, and drug consumption, and less hazardous sexual conduct. In addition to being less likely to participate in antisocial and illegal activity, conscientious people seem more religious and possess conservative political opinions.

2.3.3 Extraversion

Extraversion suggests if a person is outgoing and chatty in social circumstances. Its fundamental characteristics are friendliness (against shyness), assertiveness (versus submission), and activity (vs lack of energy). Extroverts prefer to speak frequently, take leadership in group settings, and show positivity, whereas introverts typically feel uncomfortable in social circumstances and keep their ideas and emotions to themselves. Extraversion is a good predictor of social outcomes like acceptance and friendship from peers, social status, people dating, and relationship satisfaction. Extraverts tend to like and do better in jobs that require them to interact with people and take risks. They are more likely to take on leadership responsibilities at work and community. Psychologically, extraverts generally have higher self-esteem and more excellent contextual mental health, particularly intense positive affect. Compared to introverts, they have more emotional stability and better-coping skills when bad things happen.

2.3.4 Agreeableness

Agreeableness is an essential component of social conduct. It refers to a person's pro-social behaviour and ability to sustain satisfying, harmonic interpersonal relationships. Essential components of agreeableness are compassion, courtesy, and trust. Those having high agreeableness are more inclined to assist and forgive others and to treat others with respect. In contrast, those with a low level of agreeableness are more likely to look down on others, start fights, and harbour grudges. Agreeableness is an effective predictor of social outcomes like extraversion. Agreeable people are more likely to be accepted and liked by their peers and be happy in their dating and relationship lives. In contrast, people with low agreeableness are often more likely to be bullied and rejected by their peers. People who are easy to get along with tending to look for jobs where they can work with others and do well. There are high chances that such people are religious, volunteer, and take on leadership roles in their communities. They are less inclined to commit crimes. A low level of agreeableness is linked to health problems like heart disease and a shorter life span.

2.3.5 Neuroticism

Neuroticism (Emotional Stability in the BFM) refers to the propensity for unpleasant feelings and moods. Its primary characteristics are anxiety (against tranquillity), sadness (vs satisfaction), and emotional instability (vs stability). Individuals with a high level of neuroticism feel more frequent and powerful negative emotional states, such as fear, despair, and frustration, as well as frequent mood changes. Those with low neuroticism maintain composure and optimism, even under challenging circumstances, and find it simpler to manage their emotions. High levels of neuroticism are strongly correlated with poorer levels of happiness and life satisfaction and lower levels of self-esteem. This overall discontent spreads to more-specific life arenas. For instance, those who are neurotic are more likely to experience marital problems like fighting or being abused by a significant other or even getting a divorce. They also have a lower work satisfaction, loyalty, and achievement rate. Neurotic people are more likely to develop mood and anxiety disorders, including clinical depression and generalized anxiety, because they frequently feel unpleasant emotions and have trouble coping with bad experiences.

2.4 Are Facial Action Units and Personality Traits related

Facial expressions have garnered a great deal of attention as they are one of the primary nonverbal cues and emotional state-expressing methods. Techniques for automating the interpretation of facial expressions [Anderson and McOwan, 2006, Pantic and Patras, 2006]

are essential to implementing more realistic and successful human-computer interactions, which may complement current research on human emotion and affective computing and improve widespread applications. However, exploring associations between facial behaviours and personality traits is not new. Research has been going on since the era of Aristotle. He was investigating if appearance and characteristics have any relationship [Gloor et al., 2021]. Along with his colleagues, Paul Ekman performed since the 1970s and found evidence that shows certain facial expressions are universal. These represent happiness, disgust, anger, fear, and surprise. The study on these expressions was carried out in various cultures, and the common thread was the emotions expressed through facial expressions, whether it was Japanese faces or American faces. Among kids, the researchers found that babies show various expressions without being taught, which led to them concluding that the expressions being shown by the babies were innate. A FACS (Facial action coding system) was developed by Friesen and Ekman [Ekman and Friesen, 1978]. In this system, facial expressions were coded, and the movements on the face were described by Action Units(AUs). Each facial expression has a different combination of Action units. The work by Friesen and Ekman paved the way for many researchers to use techniques like image processing and video processing to study and categorize various facial expressions. The list of action units coded is shown in the Fig 2.3.

Multiple research have been done which are focused on facial action units [Pantic and Patras, 2006, Gloor et al., 2021, Lee et al., 2012, Sayette et al., 2001]. As compared to the vast use of facial action units to identify emotions (Facial Expression Recognition), the research on using Facial action units is relatively less. The study done by [Biel et al., 2012] focuses on identifying emotions of people playing video games using their facial expression. Another study done by [Gavrilescu, 2015] makes use of FACS to determine the 27 facial action units. the architecture is customized version of [Gavrilescu, 2014] and is divided into three blocks - 1) Action unit detection 2)Behavioral Map Building and 3) Personality traits forward feed neural network. The results showed that the system identified Openness to experience, Extraversion, and Neuroticism but could not perform well for agreeableness and conscientiousness, but this can be explained to the type of dataset (emotion-based) and these traits are action-induced.

Though, often facial expressions can be misleading as well, as the user might be masking their genuine emotions, mainly when users are being recorded or performing online activities. Although, in a real task-based engaging scenario, masking emotions can be challenging. [Hill and Craig, 2002, Littlewort et al., 2009]. However, this study suggests a fresh way of exploring the correlations between different facial action units and personality traits through a three-party collaborative task.










































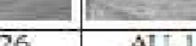
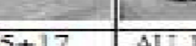






Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
					
Inner Brow Raiser *AU 41	Outer Brow Raiser *AU 42	Brow Lowerer *AU 43	Upper Lid Raiser AU 44	Cheek Raiser AU 45	Lid Tightener AU 46
					
Lid Droop	Slit	Eyes Closed	Squint	Blink	Wink
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
					
Nose Wrinkler AU 15	Upper Lip Raiser AU 16	Nasolabial Deepener AU 17	Lip Corner Puller AU 18	Cheek Puffer AU 20	Dimpler AU 22
					
Lip Corner Depressor AU 23	Lower Lip Depressor AU 24	Chin Raiser *AU 25	Lip Puckerer *AU 26	Lip Stretcher *AU 27	Lip Funneler AU 28
					
Lip Tightener	Lip Pressor	Lips Part	Jaw Drop	Mouth Stretch	Lip Suck
AU 1+2	AU 1+4	AU 4+5	AU 1+2+4	AU 1+2+5	
					
AU 1+6	AU 6+7	AU 1+2+5+6+7	AU 23+24	AU 9+17	
					
AU 9+25	AU 9+17+23+24	AU 10+17	AU 10+25	AU 10+15+17	
					
AU 12+25	AU 12+26	AU 15+17	AU 17+23+24	AU 20+25	
					

그림 4. Facial action unit 및 action unit 조합.

Figure 2.3: Facial Action Units [Ko and Sim, 2010]

2.5 Conclusion

This chapter discussed the background of the research and went in depth into the different topics that this research is a part of. In the next chapter, a description of the different approaches used to convert the background discussed into an experiment is discussed,

3 Methodology

This chapter explains the different methodologies used to achieve the desired results. The chapter starts with the Data section, which gives an insight into the MULTISIMO dataset, its collection process, information and personality assessments of the participants. Next, the research uses the PyFeat toolkit to extract action units from the video, which is explained in The instrumentation section. The Machine Learning section covers all the statistical methods and machine learning models used to study the relationship between various Facial Action Units and Personality Traits. Later in the Evaluation Methods, different parameters to evaluate these ML models are discussed.

3.1 Data

This section talks about the MULTISIMO [Koutsombogera and Vogel, 2018] corpus, its experimental setup and technical details, ethical concerns and the features used. The corpus was designed as part of the MULTISIMO project to investigate and model collaborative characteristics of multimodal behaviours while performing simple tasks as a group. This corpus attempts to bridge the gap in investigating factors contributing to collaborative behaviours and tools measuring the group's success in a three-party activity.

3.1.1 Corpus Overview

MULTISIMO is a multimodal, multiparty corpus comprised of collaborative group interactions wherein a facilitator directs two participants to deliver answers to a quiz. The corpus consists of video and audio recordings from 23 different sessions involving 49 participants, of which 46 act as players and 3 act as facilitators. Each session recorded was of an average of 10 minutes (max = 16 mins, min = 6 mins) time duration, thus making it a complete recording of roughly 4 hrs.

Each session had 3 participants, with 2 participants acting as players and the other acting as facilitator of the session. Participants collaborated while the facilitator kept track of their progress and gave comments and recommendations as required. The facilitator poses a series of questions, and participants should respond with three answers that they predict will

be the most prevalent replies. In this instance, collaboration refers to players collaborating to identify and rank the appropriate solutions. Collaboration is quantified using communicative aspects that demonstrate verbal and non-verbal behaviour during three-party task-based dialogues.

3.1.2 Participants - Players and Facilitators

Participants were Trinity College Dublin students or researchers recruited via online applications. A total of 49 participants were recruited, 46 were later paired for the 23 sessions, and the remaining three were selected as facilitators. The players were randomly paired depending on their availability to join the experiment. In most sessions, the participants did not know one another; however, there were four groups where the members were either friends or coworkers. The median age of the participants is 30 (ranging from 19 to 44). The gender distribution is balanced, with 25 female and 24 male players. In addition, there are eighteen different nationalities among the participants, with one-third being natural English speakers.

Those selected to play the facilitator position were educators or tutors with expertise in directing group activities. Furthermore, to avoid variability, certain variables were kept constant while choosing facilitators, like gender, English proficiency, nationality and career (female Greek English teachers). The role of facilitators was highly crucial as this role would later be modelled as an embodied conversational agent that can organise group interaction and assist members in achieving their objectives. Thus, the facilitator function was created to provide the extraction of behavioural clues for the construction of the agent mentioned above.

3.1.3 Corpus Experimental Setup

All the sessions were conducted in English. The task was to have an interactive conversation with the motive of giving the 3 most popular answers to each of the 3 questions asked. Therefore, the players teamed up, came up with personal opinions, discussed the most probable answers, and after coming to 3 mutual options and rankings, they informed the facilitator.

The answers were decided based on responses in a survey of 3 questions given to 100 people. The questions were extracted from a Family Feud game [Fam, 2022] related database as described in the table 3.1. The questions were chosen so that they would be simple to answer for both native and non-native English speakers, prompt conversation among the participants, and stimulate multimodal behaviour, such as using gestures while describing an item or a concept. Following the conclusion of each session, participants completed a brief questionnaire to reflect their opinion of the experiment.

Since the aim of this setup is for group collaboration along with successful task completion, a successful task evaluation will be measured on several characteristics -

- correct answers submitted
- amount of time taken
- the extent of involvement of both players, which can be measured by speech activity, mutual gazes shared and direct interactions.

Table 3.1: Questions asked to all the participants

Questions
name a public place where it's likely to catch a cold or a flu bug
name 3 instruments you can find in a symphony orchestra
name something that people cut

3.1.4 Corpus Technical Setup

High-quality video files with a resolution of 1920x1080 px were used for the three participants. Two cameras shot at 29.97 frames per second were placed opposite the players to record their frontal view. In addition, there was a third camera that recorded the whole scene. Audio is captured by an Omni-directional microphone and the head mics of each participant. Since in this dissertation, the main focus is on the players' facial action units and emotions, the video generated from the two cameras capturing the players' frontal view is considered.

3.1.5 Personality Traits Assessment

As stated earlier, the purpose of creating the MULTISIMO corpus is to explore the impact of personality factors not only on task completion but also on the collaborative behaviour of participants, including their interest, attentiveness, and initiate or handle conversational disagreements since personality factors are an essential instrument for interpreting social behaviour. Therefore, participants might come up with answers that might seem very probable, but if they are not the precise, most common three responses, the correct solution is still lacking. Although the expected responses are collected through a survey, so they are not factually correct, the Correctness criteria, even though subjective, help promote a variety of emotions in the participants and encourage them to continue to collaborate and interact with their partners. Thus, providing a plethora of cues needed for this experiment.

The participants filled out the BFI-44 questionnaire before the recordings. The results of the

BFI personality assessments were documented per person. For further evaluation purposes, the percentile rank for every person was measured using local norms.

3.2 PyFeat

This section discusses PyFeat and summarises its functionalities used for feature extraction (Facial Action Units and Emotions). The resultant features are used as a feature set to train the Machine learning models.

PyFeat [Cheong et al., 2021] is an open-source tool which facilitates facial expression analysis. Based loosely on the idea of Openface [Baltrušaitis et al., 2016], it provides various features such as the face (or multiple faces) captured from an image or video, detection of facial features (landmarks, action units and emotions) along with additional features like preprocessing, analysis and visualisation of the data.

The Py-Feat tool is divided into two modules - Detector and Fex. The Detector module works on the detection of features from any image or video and returns the result as a *Fex* data class. Fex data class is further helpful in the preprocessing, analysis and visualising of facial expressions from the dataset. These two modules will be used heavily in the feature extraction for the research and are described briefly in the below section -

3.2.1 Detector

The Detector modules help detect the following facial features a) facial landmarks, b) a face from an image or video, c) facial action units movements, and d) emotions.

The detector features used in this research and the related algorithms used are described as follows -

1. Face Detector - This detector is used to identify the location of the face in an image or video frame if there exists any. The face detection values are stored in the form of the face's bounding box.

Py-feat used three models to train on multiple datasets - RetinaFace [Deng et al., 2019], MTCNN [Zhang et al., 2019] and Faceboxes [Zhang et al., 2019].

The default face detector is Feat-RetinaFace as of 0.3.7 and performs better than others. Furthermore, the results recorded were best for the easy detection task (where the subject was in close proximity of the camera) for the WIDER [Yang et al., 2016] dataset.

2. Action Unit Detectors - PyFeat provides 20 different AUs - [AU01, AU02, AU04, AU05, AU06, AU07, AU09, AU10, AU11, AU12, AU14, AU15, AU17, AU20, AU23,

AU24, AU25, AU26, AU28, AU43]. To detect these AUs, Pyfeat includes four AU detectors trained on different datasets - JAA-Net Neural Network [[Shao et al., 2021], [?]], Random Forest Classifier (Feat-RF), Linear SVM classifiers (Feat-SVM) and Logistic Regression Classifier (Feat-Logistic).

Feat-RF gave the best results for AUs 5, 25, and 26; overall, Feat-SVM performed the best out of all four detectors.

Feat-SVM provides the results in set (0,1), indicating if the particular AU was detected or not. Whereas, Feat-RF gives results in detection probability ranging from [0-1], making it convenient for analysis. Feat-RF is the default AU detector as of version 0.3.7.

3. Emotion Detectors - Pyfeat emotion detectors can detect seven types of emotions - *Anger, Happiness, Neutral, Fear, Disgust, Sadness, Surprise*. One can choose any detector among the four - ResMaskNet [Pham et al., 2021], Feat-FerNet, RandomForest and SVM to detect these emotions. Random Forest and SVM are statistical learning models and use the concept of HOG features to generate values of emotions. ResMaskNet has the best accuracy rate among all. However, a HOG-based action unit detector (Feat-RF, Feat-SVM, Feat-Logistic) should be paired with a HOG-based emotion detector (Feat-SVM, Feat-RF) and vice-versa.

To detect facial characteristics from an image - `detect_image()` function is used whereas to detect from video `detect_video()` function is used. The parameters are - input video/image file name, the output file name(optional), `skip_frames` - tells how many frames to skip.

3.2.2 Fex

The Fex data class inherits from the Panda data frame [pandas development team, 2020, Wes McKinney, 2010] and is used for the facial expression analysis. In addition, this data class provides basic functionalities like slicing, sampling, and data summary and advanced functionalities to process the dataframe returned from the Detector module. Finally, the fex data class facilitates data manipulation (select different data segments - facial landmarks, faceboxes, emotions and action units), preprocessing facial expression time series data, aggregating facial action units and visualising the data.

The functionalities mentioned above used in this research are -

- `read_feat()` - This is used to read any feat detector file and treats it like a Fex data class
- `aus()` - retrieves all Action Unit columns
- `emotions()` - retrieves all emotion columns

- *sessions* - If any index value is passed to this argument, then the dataset will be grouped according to that particular instance for further processing
- *baseline()* - normalizes the facial action units with mean or median and if provided, normalization can be done on session basis.

3.3 Machine Learning Pipeline

This dissertation uses several machine learning algorithms to predict an individual's personality traits and compare their outcomes. This section overviews the Machine Learning algorithms utilised in this dissertation.

Machine learning is a field of study responsible for making machines smart enough to carry out any specific task without being explicitly programmed. It uses statistical methods on existing data to train the machine. Machine learning algorithms are of two types- 1) Supervised learning (learning from the labelled data) and 2) Unsupervised Learning. Unsupervised learning has no output values, and the algorithm attempts to infer patterns from the data. Regression and classification are two further subcategories of supervised learning tasks. This dissertation uses supervised learning; hence it will be discussed in the next part.

3.3.1 Supervised Learning

Supervised learning is a category of machine learning algorithms that employ labelled data for model training. The algorithm's objective is to deduce a function that links input parameters to an output variable. Once the mapping function has been extrapolated, predictions on new inputs may be made. Input variables are called independent factors, and output variables are dependent factors. Algorithms employ the dependent variable to learn or infer associations with the independent variables, enabling them to anticipate the output for unobserved data. The dependent variable can be of type categorical, such as Boolean, or continuous. Before passing the data to the ML algorithm, it needs to be processed. The method for preparing raw data for a machine learning algorithm combined with the application of domain expertise to develop new features is known as feature engineering. The necessity for feature engineering stems from the fact that raw data frequently produces poor forecasts.

Classification and regression are the two types of supervised learning tasks. A classification problem is one in which dependent variables are categorical. The method aims to develop a model that can classify test data into one of the categories. Regression problems, on the other hand, have continuous values as dependent variables. Since Five Factor personality traits are divided into two categories ("High" and "Low"), this research approaches the personality evaluation problem as a classification problem, employing different classification

algorithms to predict personality traits.

Fig 3.1 is the pipeline referenced in this research and, in general, any machine learning-based project. The remainder of the section details the steps taken and machine learning algorithms used in this research.

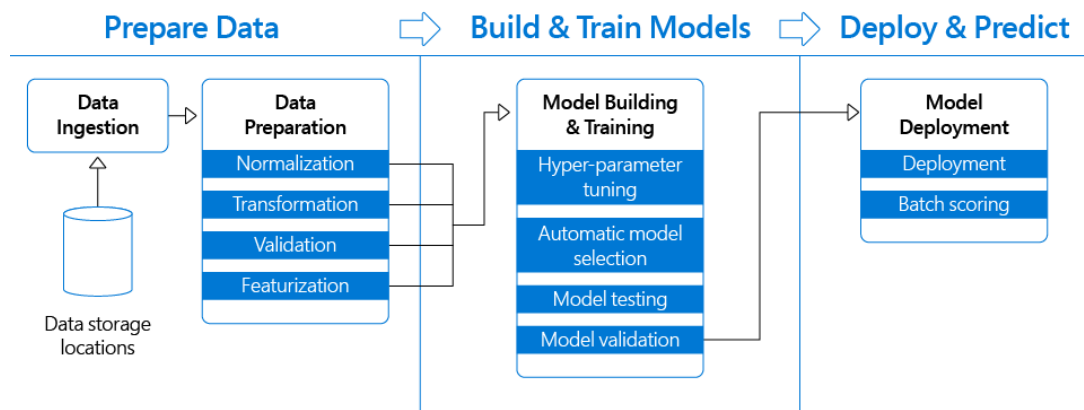


Figure 3.1: Machine Learning Pipeline [Lazzeri, 2019]

3.3.2 Data Collection

The first phase is to analyse the problem and collect relevant data. The Data Preparation method covers Data Collection and Data Preprocessing. Machine learning model prediction accuracy relies on the quality of data given to it. Therefore, the data collection process becomes even more significant.

ML models are given several distinct datasets, each serving a unique purpose. Training datasets are entered into the ML algorithm before validation datasets (or test datasets) are used to validate the model's interpretation of the training datasets. Once these training and validation sets have been loaded into the system, future datasets may be utilised to refine the machine learning model. Data passed can be of the following types - categorical data, continuous data, time series data, and text corpus [Dat].

Below are some problems that might arise during the data collection process that needs to be taken care of -

- The data used is not complete or relevant to the problem statement.
- There are many missing values in the data, thus making that column (or set of columns) unreliable.
- All the columns in the dataset are not standardised and have varying ranges.
- Data can be biased

The solution to these issues is Data Preprocessing, a part of the Feature Engineering step discussed in the following subsection.

3.3.3 Feature Engineering

The most effective approach for developing machine learning models is feature engineering. Feature Engineering is an umbrella term for various actions done on variables (features) to make them utilisable for an algorithm. This step boosts the model's precision, improving the forecasts' accuracy. Feature engineering has two objectives - 1) Preparing an input dataset suitable to the requirements of the machine learning algorithm. 2) Enhancing the effectiveness of the model. For this reason, this research gives immense attention and time to the process surrounding classification problems. Furthermore, this research does almost all the feature engineering in Python. Although, the same approach can be found in R or any other programming language with different function names. Thus, it is easy to recreate the results by following the same approach. The different feature engineering approaches used in this research are discussed below.

Imputation

In preparing data for machine learning, missing data is a common obstacle. Possible causes of the missing values include human mistakes, data flow outages, and privacy concerns. Regardless of the cause, missing values negatively impact the performance of machine learning models.

Some machine learning tools automatically eliminate rows with missing data during model training, which reduces the model's performance due to the reduced training size. In contrast, most algorithms reject datasets with null values and return an error.

The simplest remedy for missing values is to delete the entire row or column. Although, imputation is preferable to dropping since it preserves the amount of the dataset. However, there is a significant selection of missing value replacements. The optimal method for imputation is to use the column medians. Since column averages are vulnerable to outlier values, medians are more stable. Mean, and standard deviation are also possible options for replacing NaN. Categorical columns can be effectively managed by substituting missing values with the most frequent value. If the values in the column are distributed equally, and there is no dominating value, it may be more prudent to impute a category such as "Other".

[Dat]

Binning or Discretization

Binning takes data values and categorises them into bins (or buckets). It is essential in data exploration endeavours. Typically, it is used to convert continuous variables into categorical

variables. It makes the model more resilient and prevents overfitting, although it comes at the expense of performance. Information is lost, and the data becomes more standardised. The main element of the binning procedure is the trade-off between performance and overfitting. Given its impact on model performance, binning may be unnecessary for several numerical column techniques, except in a few situations of evident overfitting.

However, for categorical columns, labels with low frequencies likely harm the robustness of statistical models. Assigning a generic category to these less common variables maintains the model's resilience.

The data can be categorised as follows:

- Clustering of identical intervals
- Grouping based on frequency parity (of observations in the bin)
- Clustering based on a decision tree (to establish a relationship with target dependent variable)

Grouping Operations

In most machine learning methods, each entity is represented by a dataset row, with each column representing a specific entity property. This type of data is known as "Tidy." Although few datasets might have many rows in an instance and not fit the notion of "tidy" datasets, in such a scenario, the data can be grouped by instances, with each instance represented by a single row.

The primary objective of the *group by* operations is to determine the aggregation functions of the attributes. Typically, sum and mean are efficient solutions for numerical characteristics. However, categorical characteristics are more complex. [Rençberoğlu, 2019]

Feature Split

Splitting features is an excellent method to make non-significant features useful in machine learning. Most of the time, the dataset comprises string columns that contradict the criteria of tidy data. By separating the useful portions of columns into new attributes -

- it is possible for machine learning algorithms to understand them.
- Make it feasible to sort and categorise them.
- Improve the performance of the model by identifying potential information. [Fea, 2020]

Scaling

In most instances, the numerical characteristics of the dataset lack a predetermined range and differ from one another. Scaling resolves the issue. After a scaling procedure, the continuous attributes have the same range. This procedure is optional for the majority of algorithms. However, it may be desirable to implement. There are two methods of scaling:

1. Normalisation (or min-max normalisation) scales all values between 0 and 1 in a given range. However, this adjustment has no effect on the feature distribution, and because the standard deviations have been reduced, the influence of outliers has increased. Thus, outliers need to be handled prior to normalisation.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

2. Standardisation (also known as z-score normalisation) adjusts values while accounting for standard deviation. If the standard deviation of the characteristics differs, their range will likewise differ. It lessens the impact of outliers in the features.

$$z = \frac{x - mean}{std_dev} \quad (2)$$

Feature Selection

Feature selection is the selection of necessary independent features. Selecting significant independent variables that are more related to the dependent feature can aid in developing a good model. There are several approaches for selecting features [Fea, 2020] and making them model ready:

- Find the relation between the independent variable and dependent variable by plotting the correlation matrix using a heat map and determine only highly related independent variables using statistical tests
- Convert categorical string data to factors
- Add and delete relevant columns

Correlation Analysis

Correlation is a measure of the relation between the given variable. In correlation analysis, a sample correlation coefficient is calculated based on data; in the case of normal distributions, the Pearson correlation coefficient is utilised to detect the associations. The correlation coefficient, varying from -1 to +1, measures the direction and intensity of the

linear relationship between the two variables. The relation between two variables can be either positive or negative. The sign of the correlation coefficient indicates the direction of the link. Whereas the value reflects the strength of the association.

3.4 Machine Learning Algorithms

This section thoroughly discusses the various Machine Learning Algorithms used in this research.

3.4.1 Decision Trees

This article describes the theoretical implementation of decision trees. In addition, it discusses decision tree-related terms used in the research.

A decision tree is a visual depiction of potential decisions taken by the algorithm depending on particular parameters. It is termed a decision tree because, like a tree, it begins with a single variable and then branches out into various possible answers. Decision Trees are a flexible Machine Learning method capable of regression and classification. Moreover, they are capable of fitting complex datasets. A decision tree is made of three major elements:

- The topmost node is referred to as the Root Node. It denotes the most accurate prediction (independent variable).
- Decision / Internal Node: The nodes where independent features are tested, with each branch representing a test outcome.
- Leaf / Terminal Node: It contains the final prediction outcome.

The Decision Tree operates as follows:

1. Choose the variable that produces the best split
2. Divide the data according to the value of this variable.
3. Steps 1 and 2 must be repeated. When no additional gain can be gained, or when some pre-set stopping conditions are reached, splitting ceases. (Alternatively, the data is separated as far as feasible before the tree is trimmed.)

The split is based on Gini Index, which quantifies impurity in a given node. It ranges from 0 to $(1-1/n)$, where n represents the total classes in the variable. The error rate of misclassification of the tree can be corrected by improving the *Complexity Parameter* (cp) of the tree. The hyperparameters that can be tuned are 1) the Maximum depth of the tree and 2) the Minimum data in terminal nodes.

Decision trees have many advantages but also a few disadvantages. The advantages are that the decision trees are simple to understand and work even when variables have nonlinear connections and outliers have no effect on the results. However, they tend to overfit. As a result, decision trees do not achieve well on the validation sample. Furthermore, it presupposes that all independent features interact with one another, which is not always the case. However, the disadvantage can be fixed. For example, the overfitting issue can be fixed by pruning the tree. Pruning shrinks decision trees by deleting parts of the tree with minimal capacity to categorise cases—pruning results in less complicated trees. In addition, it eliminates abnormalities in training data caused by noise or outliers.

3.4.2 Random Forest

Random forests are premised on the concept of "collective intelligence." The aggregate of numerous predictors provides a more accurate forecast than the most acceptable individual predictor. The collective of forecasters (decision trees) is known as an ensemble. Consequently, this method is known as Ensemble Learning. This technique is used to improve the results obtained through decision trees. Several Decision Tree classifiers are trained on a distinct random subset of the training data. The predictions of each tree are collected, and then the class label or value that receives the most votes is predicted. This method is known as Random Forest. Random forests can be used for classification and regression. The random forest can handle a vast number of attributes and aids in identifying the most significant characteristics. It prevents overfitting. Although, it becomes hard to interpret random forests.

Random forests lower tree correlation by introducing more unpredictability into tree growth. Specifically, random forests employ split-variable randomisation while developing a decision tree throughout the bagging process. Gini Impurity decides the best split. Each tree works is - 1) Each tree takes 66% of the random data with replacement and trains on it. 2) m variables are selected out of the total features available for doing the split. 3) The error is calculated on the misclassification of the leftover (Out of Bag) data. The average error from all the trees is calculated as the Out of Bag error. This error is considered while hyperparameter tuning as well. 4) For final prediction, prediction from each tree is considered, and the value with the maximum number of votes is used. Although several hyper-parameters may be tweaked, the default settings yield above satisfactory outcomes [Brandon Greenwell, 2020].

Discussed below are a few of the hyper-parameters that can be tuned for better model accuracy and are used in this research -

```

1. Given a training data set
2. Select number of trees to build (n_trees)
3. for i = 1 to n_trees do
4. | Generate a bootstrap sample of the original data
5. | Grow a regression/classification tree to the bootstrapped data
6. | for each split do
7. | | Select m_try variables at random from all p variables
8. | | Pick the best variable/split-point among the m_try
9. | | Split the node into two child nodes
10. | end
11. | Use typical tree model stopping criteria to determine when a
    | tree is complete (but do not prune)
12. end
13. Output ensemble of trees

```

Figure 3.2: Random Forest Algorithm [Brandon Greenwell, 2020]

m_{try}

m_{try} handles the split-variable randomisation functionality of random forest models. It establishes a compromise between low tree correlation and good prediction power. For classification and regression issues, the default value of $m_{try} = p/3$ for regression problems and for classification $m_{try} = \sqrt{p}$. Nonetheless, where there are fewer relevant predictors (such as noisy data), a greater value of m_{try} is preferred as it selects strongly related features, and when there are plenty of relevant features, then a lower m_{try} is preferred. If $m_{try} = p$, then it becomes the case of bagging.

Number of trees - ntrees

Another hyperparameter is the number of trees. The number of trees should be high enough to stabilise the error rate. However, other hyperparameters should be modified first because the value of trees might change accordingly. Also, the influence on computing time grows linearly with the number of trees.

3.4.3 Variable Importance

After running the model, the next step is to find which features influence the model most, i.e. which independent features are most important in predicting the dependent feature. Variables with high importance impact the result, whereas variables with low importance can be omitted from the model to make the model simple and fast. In a random forest classifier,

variable importance can be identified using Mean Decrease Accuracy (MDA) and Mean Decrease Impurity (MDI) or Mean Decrease Gini. Each variable is assigned the measures mentioned above of significance. The first metric is based on how much the accuracy drops when the variable is removed. The second measurement is based on the decline of impurity when a variable is selected to split a node, in this case, the Gini Impurity. Both methods are explained below.

Mean Decrease Accuracy (MDA)

As mentioned earlier, random forests train each tree on a subset of the training data. The remaining data employs as Out-of-bag data. First, the particular tree accuracy is calculated on the out-of-bag. Then, the values of the feature whose importance is to be calculated are randomly shuffled in the out-of-bag dataset. Shuffling ensures that any existing correlations for the feature in the dataset are destroyed, thereby nullifying the importance of the particular variable. After shuffling, the model accuracy is calculated again. Finally, the mean decrease in accuracy is noted for all the trees, giving the value of mean decrease accuracy. In layman's terms, mean decrease accuracy is how much the model accuracy will decrease if the variable is removed. Thus, the higher the value of MDA, the more influential the variable. This process is also called permutation feature importance.

According to the *random forest* library in *R*, the classification error rate is calculated. Nevertheless, the metric of MDA is not important. The relevant measure to the other variables is considered while calculating the feature importance.

Although, this method is known not to produce good results if the variables are highly correlated as one variable is shuffled rather than completely removed. Also, if the variable has significantly less importance, shuffling the values might increase the accuracy due to the introduction of random noise, thus leading to false importance, but this can be discarded during the evaluation process.

Mean Decrease Gini (MDG)

Gini impurity records the chances of misclassifying a record in the training dataset. It is a decisive measure while splitting the tree at each node. The variable with the lowest Gini impurity is selected to split the tree further. Every time any particular feature is used to make the split, the decrease in gain is calculated. For calculating feature importance, the sum of all gini decreases because the feature is calculated and divided by the total trees that used the feature. This mean is known as MDG. The mean decrease Gini scale is not significant, and only relative values matter while evaluating the feature importance.

Since the Gini index can be calculated while training the model, it is faster to calculate. However, MDG is biased towards numerical or multiple categorical independent variables

compared to the dichotomous variables because of the many splits they offer.

3.5 Evaluation Methods

This section outlines the tools to evaluate the efficacy of the machine learning algorithms presented in section 3.4. These metrics assist in selecting the optimal model from the available options.

3.5.1 Mann Whitney U Test

Statistical significance tests are applied to the results obtained from machine learning algorithms to confirm whether a relationship exists or where the results are generated by chance. This research will be making use of the Mann-Whitney U test. It is a non-parametric rank-based test used to identify if there exists any difference between two groups on an ordinal or continuous variable. It is also known as the Wilcoxon-Mann-Whitney [Mann and Whitney, 1947]. The null test hypothesis is that the two groups populations are equal. The test requires two independents (dichotomous) variables and a dependent (continuous/ ordinal) variable. The test ranks all the values (scores) of the dependent variable, with the smallest value having the lowest rank. The independent variable has only two groups, and the ranks attached to each are computed and averaged out. This gives a mean rank for each of the groups. If the mean rank is the same for the groups, then their distribution is also the same. Therefore, the null hypothesis can be accepted. However, if one group got assigned higher rank values, then that group will have a higher mean rank, the distributions will be different, and the null hypothesis is not accepted. For statistical significance, the difference in mean rank is calculated.

A few assumptions should be present to use this test and are stated below -

1. There exists one continuous or ordinal dependent variable
2. There exists one dichotomous categorical variable
3. The observations present in each group or among the group should have no relationship
4. The distribution shape must be determined of the two groups present in the independent variable. If the shape of the distribution of the two groups are same, then the test checks if any difference exists in the medians of the two groups. Otherwise, differences in the distribution are checked.

3.5.2 Performance Measures

Assessing the effectiveness of any given machine learning algorithm is critical. It helps in evaluating the performance of the model and deciding if the hyper-parameters should be tuned or not. Furthermore, it helps collect accuracy measures data for each ML algorithm and compare it to determine which model performs better for the task. Since this research approaches automatic personality detection as a classification issue, the discussion in this section is focused on metrics used for evaluating classifiers.

One of the first stages in analysing a supervised machine learning algorithm is creating a confusion matrix and a table used to summarise an algorithm's prediction outcomes. The table summarises correct and unsuccessful predictions with count values and categorises them. The table is a 2X2 matrix for binary classification, with each column carrying the count of four qualities - True Positive, False Positive, True Negative, and False Negative. Next, the accuracy, precision, F1, and ROC AUC are calculated using these four metrics.

For example, an ML algorithm is trained to predict a binary dependent variable with two categories, True and False. The percentage of cases when both the actual and predicted outcomes are true is known as the true positive rate (TPR). The number of False labelled predictions that were classified correctly is the true negative. Similarly, the number of properly anticipated False occurrences is known as the False positive, whereas the number of wrongly predicted False instances is known as the False negative. The measurements stated above are computed using these four values. After getting these values, the important task is to determine which of the two, i.e. True negative or False negative, has to be minimised to acquire better results. This choice is based on the conditions and environment in which the model will be used [Suresh, 2021]. Fig 3.3 explains the concept explained in a graphic manner.

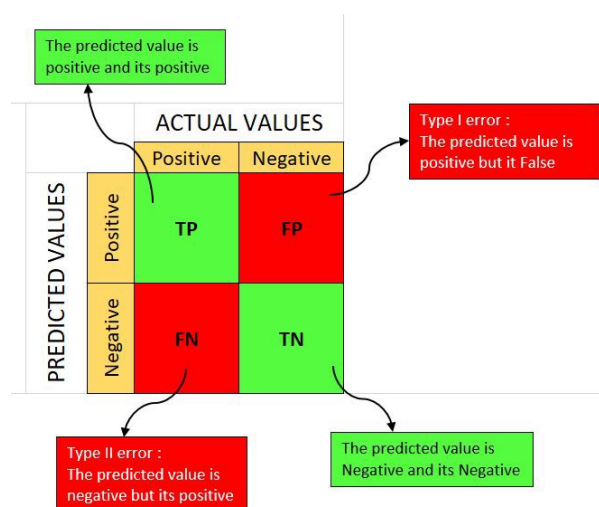


Figure 3.3: Confusion Matrix [Suresh, 2021]

$$\text{CorrectPrediction} = \text{TruePositive}(TP) + \text{TrueNegative}(TN)$$

Accuracy

Accuracy quantifies how frequently the classifier predicts correctly. It is the ratio of correct classifications to the total classifications made. Accuracy metric is not advised for imbalanced classes. For unbalanced data, the accuracy will be high when the model predicts all the points belonging to the majority class label, thus attaining a high accuracy rate. The model, however, is inaccurate.

$$\text{Accuracy} = \text{CorrectPredictions} / \text{TotalPredictions} = \tag{3}$$

$$(TP + TN) / (TP + TN + FP + FN) \tag{4}$$

Precision

Precision is the total number of successfully classified positive classes to the total predicted positive classes. The value of precision should be high (ideally 1). When False Positives are more problematic than False Negatives, precision is a valuable indicator.

$$\text{Precision} = \text{CorrectPositivePredictions} / \text{TotalPredictedPositive} \tag{5}$$

$$= (TP) / (TP + FP) \tag{6}$$

Recall

It measures how many positive observations are accurately predicted as positive. It also goes by the name Sensitivity. When the aim is to catch as many positives as possible, recall is a good option for evaluation statistics. Therefore, recall ought to be high (ideally 1).

$$\text{Precision} = \text{CorrectPositivePredictions} / \text{TotalActualPositive} \tag{7}$$

$$= (TP) / (TP + FN) \tag{8}$$

3.6 Conclusion

This chapter discussed the theoretical concepts of different machine learning algorithms used in this research and the different performance strategies to evaluate their performance. It

gives overview of the different features of the dataset. The exact values and parameters used to generate the results and to preprocess the dataset are discussed in the next section.

4 Implementation

This chapter implements the different approaches mentioned in the Methodology section on the MULTISIMO dataset to solve the research question, - *"is automatic detection of personality traits through facial action units possible and if yes, then understanding which feature sets are indicative of the different personality traits"*. The experiments are broadly divided into three categories - 1) data collection, 2) feature engineering 3) applying machine algorithms on the final dataset generated to predict personality traits. The chapter examines the machine learning pipeline and the generation of results. Steps taken to generate features from video files are presented. The feature engineering code is in Python and uses existing toolkits and libraries. Data visualisation tasks and machine learning algorithms are implemented in R. The code can be accessed from the GitHub repository <https://github.com/ritika24-s/PredictBIG5>.

4.1 Data Preparation

Since the research question revolves around facial action units, the dataset used for this research is video recordings. The dataset was downloaded from the MULTISIMO website. Two formats are available for the videos - *High Quality and Low Quality*. The initial plan was to consider both these folders to check if the quality of videos impacted the results, but processing the HIGH QUALITY videos was time-consuming and computationally heavy. Hence could not be performed through the PyFeat toolkit. Thus, only VIDEO_LOW_QUALITY is used in this research. Furthermore, among all the videos, the research focuses only on the participants' facial expressions. Therefore, videos concerning frontal views are used (ends with *"front-video_Z_S_L.mov"*)[refer lines from 14 in code 4.1]. The videos available are from the following 18 sessions out of 23 - [S02, S03, S04, S05, S07, S08, S09, S10, S11, S13, S14, S17, S18, S19, S20, S21, S22, S23] for both High and Low Quality. Since the data collected is in video format, the videos need to be processed through a library to perform any feature engineering on the dataset. The facial action units and emotion features required for analysis can be directly extracted by processing the videos through the toolkit - PyFeat. Code is written to search each session folder, look for frontal videos, and process them through the PyFeat `video_detect()`

function. The main directory has to be passed in the parameter *root*. The steps followed are explained thoroughly in the below subsection.

Personality assessment scores are also collected from the same website in zip format. There are two types - BFI-44 and BFI-10. The respective excel files (*BFI-44-assessment.xlsx* and *BFI-10-assessment.xlsx*) are extracted from the zip folder.

During the start of this research, Pyfeat was at version release 0.3.7 and later was updated to 0.4.0. Nevertheless, for this research, version 0.3.7 has been used. Furthermore, a python script, *data_collect.py*, is created to use PyFeat features, so the functionalities of the Detector class and Fex class modules can be used.

The code written is modular, reusable and readable, following the Object Oriented Programming approach. Most variables are passed as an argument, thus making it easy to change the values with a different set of options in case a batch run is required.

First, a class named FACS is created with the following attributes -

- *root*: root directory which contains all the videos
- *dest*: destination directory, which will contain all the output CSV files but in the same order as the root directory
- *frames* (default = 30) : defines the number of frames to skip in the function *detect_video()*
- *au_model* (default = "rf") : defines the type of model to run for action units
- *emotion_model* (default = "rf") : defines the type of model to run to detect emotions

A function named *get_all_videos(loc)* (refer code 4.1) defined in the class FACS is called to get all the video file names from the root location passed while instantiating FACS class. These video file names are later passed to the *detect_video()* function.

```
1 # loc = "VIDEO_LOW_QUALITY"
2 def get_all_videos(self, loc):
3     # creates a dictionary with a key as all the videos
4     # that need to be processed
5     # value is the destination location of the .csv file
6
7     for file in os.listdir(loc):
8         subdir = os.path.join(loc, file)
9
10        if os.path.isdir(subdir):
11            self.get_all_videos(subdir)
```

```

12
13 elif os.path.isfile(subdir) and
14     subdir.endswith('front-video_Z_S_L.mov'):
15     output = file.replace('.mov', '.csv')
16     # add desired destination to output file name
17     dir = os.path.join(self.dest, (os.path.dirname(subdir) +
18     "_"+str(self.frames)+ "_frames"))
19
20     # check if the current directory exists or not
21     self.create_directory(dir)
22     output = os.path.join(dir, output)
23
24     # add filename to the videos dictionary
25     self.videos[subdir] = output

```

Source Code 4.1: Video Collection from different folders

A detector class instance is created with the following arguments -

- face_model (default value: *retinaface*) - The recordings in the MULTISIMO dataset can be considered close proximity, therefore, can be classified under easy detection task. Thus for this research, the Feat-RetinaFace model is selected for face detection.
- au_model (default value : *rf*) - In this research, results are generated using rf (Random Forest) model as it generates AU intensity
- emotion_model (default : *resmasknet*) - Since au_models used is rf, emotion_model also uses the same value.

After creating the object for the Detector class, the function detect_video() is called, as can be seen in the code shown below 4.2. The videos were captured at 29.97 frames per second. Thus, to capture frames from every second, the skip_frame is set as 30. This step also speeds up the processing of the videos. The initial plan was to check if not skipping any frame and retaining complete information impacts the results. However, this was not done in the final research stage as it was challenging to preprocess the resultant dataset and perform any operations.

Source Code 4.2: Video Processing using PyFeat

```

1 au_model = "rf"
2 emotion_model = "rf"
3 self.detector = Detector(au_model = au_model, emotion_mode = emotion_model)
4 self.frames = 30
5 for video, output in self.videos.items():
6     # check if the output file already exists

```



```

7         if os.path.exists(output):
8             print("Already processed ", video)
9         else:
10            # assert os.path.isfile(output)
11
12            with open(output, "w") as file:
13                print("Processing ", video)
14                self.detector.detect_video(video, outputFname= output,
15                skip_frames=self.frames)

```

Fig 4.1 shows the dataset generated after videos were processed through Pyfeat. The list of all the columns generated is-

```

['frame', 'FaceRectX', 'FaceRectY', 'FaceRectWidth', 'FaceRectHeight',
'FaceScore', 'x_0', 'x_1', 'x_2', 'x_3', 'x_4', 'x_5', 'x_6', 'x_7',
'x_8', 'x_9', 'x_10', 'x_11', 'x_12', 'x_13', 'x_14', 'x_15', 'x_16',
'x_17', 'x_18', 'x_19', 'x_20', 'x_21', 'x_22', 'x_23', 'x_24', 'x_25',
'x_26', 'x_27', 'x_28', 'x_29', 'x_30', 'x_31', 'x_32', 'x_33', 'x_34',
'x_35', 'x_36', 'x_37', 'x_38', 'x_39', 'x_40', 'x_41', 'x_42', 'x_43',
'x_44', 'x_45', 'x_46', 'x_47', 'x_48', 'x_49', 'x_50', 'x_51', 'x_52',
'x_53', 'x_54', 'x_55', 'x_56', 'x_57', 'x_58', 'x_59', 'x_60', 'x_61',
'x_62', 'x_63', 'x_64', 'x_65', 'x_66', 'x_67', 'y_0', 'y_1', 'y_2',
'y_3', 'y_4', 'y_5', 'y_6', 'y_7', 'y_8', 'y_9', 'y_10', 'y_11', 'y_12',
'y_13', 'y_14', 'y_15', 'y_16', 'y_17', 'y_18', 'y_19', 'y_20', 'y_21',
'y_22', 'y_23', 'y_24', 'y_25', 'y_26', 'y_27', 'y_28', 'y_29', 'y_30',
'y_31', 'y_32', 'y_33', 'y_34', 'y_35', 'y_36', 'y_37', 'y_38', 'y_39',
'y_40', 'y_41', 'y_42', 'y_43', 'y_44', 'y_45', 'y_46', 'y_47', 'y_48',
'y_49', 'y_50', 'y_51', 'y_52', 'y_53', 'y_54', 'y_55', 'y_56', 'y_57',
'y_58', 'y_59', 'y_60', 'y_61', 'y_62', 'y_63', 'y_64', 'y_65', 'y_66',
'y_67', 'AU01', 'AU02', 'AU04', 'AU05', 'AU06', 'AU07', 'AU09', 'AU10',
'AU11', 'AU12', 'AU14', 'AU15', 'AU17', 'AU20', 'AU23', 'AU24', 'AU25',
'AU26', 'AU28', 'AU43', 'anger', 'disgust', 'fear', 'happiness', 'sadness',
'surprise', 'neutral', 'input']

```

The data for BFI44 and BFI10 is collected using `pd.read_excel()` as the datasets are available in excel sheets. The code 4.3 and Fig 4.2 are attached below for reference.

```

1     bfi_44 = pd.read_excel("Personality test\BFI-44-assessment.xlsx",
2                          sheet_name="Raw and Percentile scores-ALL")
3     bfi10 = pd.read_excel("Personality test\BFI-10-assessment.xlsx",
4                          sheet_name=1)

```

Figure 4.1: Data snippet before Feature Engineering

Source Code 4.3: BFI data collection

Unnamed: 0	Unnamed: 1	Unnamed: 2	EXTRAVERSION	Unnamed: 4	AGREEABLENESS	Unnamed: 6	CONSCIENTIOUSNESS	Unnamed: 8	NEUROTICISM	Unnamed: 10	OPENNESS	Unnamed: 12
0	Session	ID	AGE	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	raw score	percentile	raw score	percentile	raw score	percentile	raw score	percentile	raw score
2	NaN	M001	33	31	89	42	91	41	93	23	42	35
3	NaN	M002	38	19	18	42	91	39	86	25	58	34
4	NaN	M003	31	21	29	33	25	40	90	34	98	34
5	S02	P006	23	24	50	44	97	36	71	19	15	34
6	S02	P007	24	20	24	32	18	31	36	26	66	47
7	S03	P008	23	16	8	28	3	26	10	29	85	32
8	S03	P009	32	29	82	37	59	30	29	19	15	35
9	S04	P010	34	15	5	34	32	38	82	22	34	47
10	S04	P011	19	30	86	30	9	26	10	28	79	33

Figure 4.2: BFI-44 data before preprocessing

4.2 Feature Engineering

This section covers all the steps to apply feature engineering on the datasets in a step-wise layout as discussed in the 3.3.3.

4.2.1 Imputation

The processed video data and personality assessment data were checked for any row or column of NaN data by passing it through `describe()` function available with Pandas dataframe. Rows which had NA values were removed using `dropna(inplace = True)` as the rows did not contain any useful information.

Source Code 4.4: Imputation of data

```

1 bfi_44.describe()
2 # check if any NaN values exist and determine what to do with them
3 bfi_44.dropna(inplace=True)

```

4.2.2 Binning or Discretization

The self-personality assessment scores in the dataset *BFI-Assessment-44.xlsx* were stored in percentiles. Although the score percentile data is not normalised and the continuous values are not required, the scores are converted into dichotomous categorical variables [DeCoster et al., 2009]. Furthermore, this makes the model more resilient. Thus following the guidelines by [Mønsted et al., 2018], the features are classified based on quantiles. The new class labels are - "High" for percentile score greater than or equal to 50 and "Low" for percentile score less than 50 for each personality trait and these values are stored in new respective columns, as can be seen in the code below 4.5 -

```

1 # convert percentile scores to categorical variable
2 def binary(self, row):
3     if isinstance(row,float):
4         return None
5     if int(row)>=50:
6         return 'High'
7     else:
8         return 'Low'
9
10 # get binary values for each trait
11 bfi_44['extraversion_score'] = bfi_44['Unnamed: 4'].apply(self.binary)
12 bfi_44['agreeableness_score'] = bfi_44['Unnamed: 6'].apply(self.binary)
13 bfi_44['conscientious_score'] = bfi_44['Unnamed: 8'].apply(self.binary)
14 bfi_44['neuroticism_score'] = bfi_44['Unnamed: 10'].apply(self.binary)
15 bfi_44['openness_score'] = bfi_44['Unnamed: 12'].apply(self.binary)

```

Source Code 4.5: Binning of Personality Traits

Feature Split

Currently, in the dataset, there does not exist any column that gives the information about the participant, except the *input column* which contains the filename, i.e., person id does not exist. Thus, this information is extracted from the *input* column. Another column is

added from the input column - *opposite_person*. The input column is dropped after this step. The related code can be seen 4.6

```
1     # fetch person and opposite person name string from the input column
2     self.data['person'] = self.data['input'].apply(lambda x :
3     re.findall(r'P\d+', x)[0])
4     self.data['opposite_person'] = self.data['person'].
5     apply(lambda x: x[0]+str(int(x[2:])+1).zfill(3))
6     self.data.drop("input", axis=1, inplace=True)
```

Source Code 4.6: Adding person and opposite_person column

Grouping Operations

The video dataset does not confine to the definition of a "tidy" dataset since it has multiple rows for the same instance - "person". Thus few operations like removing baseline and scaling need to be done on each instance as a group rather than applying it on each row individually. For this purpose, the *sessions* argument of the Fex data class is used to group the rows per person, and the rest of the feature engineering steps are performed on the dataset generated.

$$df.sessions = df['person'] \tag{1}$$

Scaling

According to a study, in real-life scenarios where expressions are being recorded, people are not making any facial expressions most of the time. Thereby neutral face expressions predominate their interactions [Afzal and Robinson, 2009]. Two effective ways to factor for this are - 1) either by recording a neutral expression of the participants and then subtracting it from the result or 2) finding the median face value, which is a representative of the neutral face and subtracting it from the feature descriptor, thus giving normalised facial features [Baltrušaitis et al., 2015].

To accommodate this factor, the *baseline()* function of PyFeat is used to calculate the median value of facial action units and is subtracted from the original values. The median values are taken by grouping data per person using the *sessions* argument mentioned above. The resultant values are in the range [-0.4, 0.4] thus have to be further scaled to the range of [0, 1] using *MinMaxScaler* from *sklearn* [Buitinck et al., 2013] [Pedregosa et al., 2011]. This is done as all the other features scale from [0, 1]. The code 4.7 has been added for reference.

```

1  from sklearn import preprocessing
2  # remove baseline median face value from the facial expressions
3  def baseline(self, data):
4      # remove baseline using Fex Function
5      new_data = data.ous().baseline(baseline = "median")
6
7      # scale the data using MinMax
8      scaler = preprocessing.MinMaxScaler()
9      names = new_data.columns
10     d = scaler.fit_transform(new_data)
11     scaled_df = pd.DataFrame(d, columns=names)
12     # copy scaled au columns to original dataset
13     data[names] = scaled_df[names]
14     return data

```

Source Code 4.7: Normalising facial features by subtracting median value

Feature Selection

The last step in the feature engineering process is to find the most significant features and remove the rest before passing the dataset to any Machine Learning algorithm. This is done in R. The dataset is imported, and any *NA* values are removed. Since to process character categorical data in R, it is crucial to first convert those columns into *factor* type. Thus, all the string type columns are converted to factors. All the columns involving facial landmarks are removed.

From the dataset BFI-10-assessment, the column *gender* is extracted to ensure the data distribution is kept non-biased while performing any operations.

The final list of columns taken from each dataset -

- video datasets - ['person', 'frame', 'AU01', 'AU02', 'AU04', 'AU05', 'AU06', 'AU07', 'AU09', 'AU10', 'AU11', 'AU12', 'AU14', 'AU15', 'AU17', 'AU20', 'AU23', 'AU24', 'AU25', 'AU26', 'AU28', 'AU43', 'opposite_person']
- BFI 10 - ['gender', 'person']
- BFI 44 - ['person', 'extraversion_score', 'agreeableness_score', 'conscientious_score', 'neuroticism_score', 'openness_score']

These datasets are merged into one using "inner" join on the "person" column and are saved as `final_dataset_rf.csv`. Fig 4.3 is shared for column reference.

Data distribution is checked for all action units by plotting the histogram, and most of the variable distribution is found to be non-normalised. Since the data size was greater than 5000, the shapiro-wilk test could not be applied. Nevertheless, the histograms plotted

	person	frame	AU01	AU02	AU04	AU05	AU06	AU07	AU09	AU10	AU11	AU12	AU14	AU15	AU17	AU20	AU23	A
0	P006	0	0.248337	0.217471	0.258003	0.083125	0.924531	0.789383	0.402193	0.96683	0.4151	0.950347	0.777628	0.131515	0.081602	0.222436	0.310025	
1	P006	30	0.223148	0.29745	0.169897	0.083353	0.909069	0.833007	0.270995	0.979998	0.4151	0.97353	0.667554	0.064222	0.073369	0.228959	0.185201	
2	P006	60	0.224658	0.230322	0.259241	0.099721	0.839753	0.761308	0.322349	0.950242	0.400628	0.931421	0.724425	0.114359	0.070765	0.236306	0.272533	
3	P006	90	0.184534	0.212434	0.28612	0.106616	0.734603	0.730403	0.292113	0.924321	0.395809	0.880223	0.721564	0.119814	0.093031	0.223883	0.276058	
4	P006	120	0.19614	0.188003	0.317503	0.188923	0.478493	0.638945	0.144112	0.739603	0.387028	0.685015	0.619226	0.076358	0.145321	0.225102	0.332674	
5	P006	150	0.168376	0.213073	0.314119	0.12114	0.559732	0.690419	0.25973	0.833312	0.390851	0.801275	0.656116	0.100105	0.139159	0.21273	0.289142	
6	P006	150	0.413127	0.299312	0.434515	0.460343	0.073807	0.160059	0.022694	0.270911	0.376991	0.172448	0.340431	0.488082	0.496853	0.051178	0.397629	
7	P006	180	0.186502	0.199765	0.341011	0.091933	0.555688	0.651419	0.258416	0.81394	0.390851	0.719364	0.641232	0.105078	0.147692	0.21483	0.329669	
8	P006	180	0.388661	0.274231	0.36642	0.653323	0.053578	0.141138	0.028753	0.243011	0.376991	0.117382	0.297746	0.427438	0.41814	0.041347	0.36852	
9	P006	210	0.23506	0.21971	0.230792	0.309408	0.44807	0.606401	0.112713	0.805387	0.395809	0.73905	0.568871	0.097687	0.168487	0.17946	0.329745	
10	P006	210	0.38952	0.279646	0.385225	0.584766	0.050407	0.124901	0.021611	0.25595	0.376991	0.118291	0.303436	0.416978	0.417809	0.041377	0.383358	
11	P006	240	0.35202	0.454326	0.332738	0.060904	0.417345	0.562155	0.343057	0.771456	0.387028	0.53016	0.616324	0.18556	0.227058	0.284629	0.38638	
12	P006	240	0.41296	0.28463	0.528878	0.402723	0.106887	0.230065	0.051545	0.261566	0.376991	0.177537	0.267902	0.504968	0.554381	0.05236	0.382863	
13	P006	270	0.281112	0.186119	0.407277	0.35437	0.040919	0.219972	0.021202	0.22611	0.376991	0.038907	0.492897	0.182302	0.215902	0.077379	0.499252	
14	P006	270	0.39717	0.291568	0.350741	0.642635	0.037162	0.134444	0.01289	0.23923	0.376991	0.107199	0.297838	0.404316	0.390693	0.041347	0.36972	
15	P006	300	0.309515	0.180777	0.410395	0.347753	0.044363	0.302649	0.032576	0.225245	0.376991	0.034475	0.453548	0.315961	0.244477	0.040226	0.440115	

Figure 4.3: Final Dataset Snippet

showed a skewed distribution and a few are shared below in Fig 4.4.

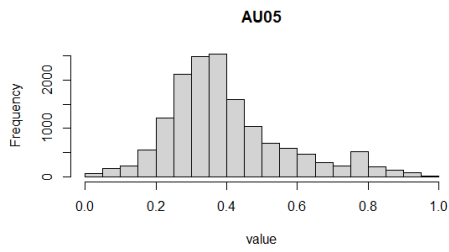
4.3 Machine Learning Models

This section thoroughly describes all the Machine Learning algorithms used and parameters and hyperparameters tuning done to improve accuracy. The basic steps to take while building any machine learning models are as follows -

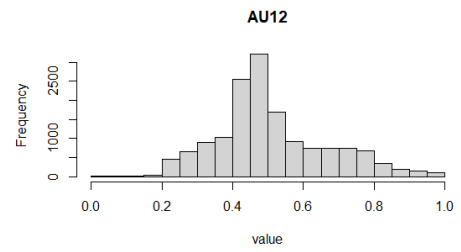
1. Step 1: Import the libraries
2. Step 2: Data Preparation (measures mentioned in 3.3.3)
3. Step 2a): Split the dataset into train and test dataset
4. Step 3: Build the model on the training dataset
5. Step 4: Look at the accuracy measures
6. Step 5: Tune hyperparameters to find best accuracy results
7. Step 5a): Visualise the error rate vs hyperparameters
8. Step 6: Predict values of the test dataset
9. Step 7: Check the accuracy rate of the tuned model

The libraries for each model used are different and will be discussed in the respective subsections. The data is split into two sets - train and test before applying any ML models, and the seed is set to 101 to compare models on the same dataset. Library 'dqrng' is used to split the dataset, as it provides unbiased data sampling. Data is distributed to keep gender distribution similar to the original dataset so as not to introduce gender bias. The gender distribution can be seen in the table 4.1.

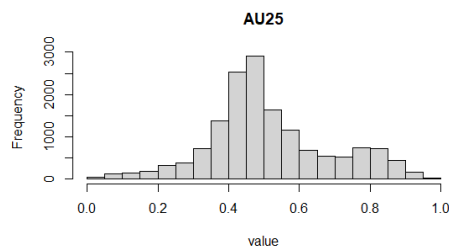
All the models used are classification models and predict *High or Low* categories for each personality trait.



(a) Data Distribution of AU05



(b) Data Distribution of AU12



(c) Data Distribution of AU25

Figure 4.4: Skewed data distribution of action units

Dataset	Female	Male
Original	0.4683602	0.5316398
Train	0.4648396	0.5351604
Test	0.4730635	0.5269365

Table 4.1: Gender distribution in original and train-test dataset

4.3.1 Decision Trees

For decision trees two libraries are used - *rpart* and *rpart.plot*. As mentioned above, the dataset is cleaned, preprocessed, and distributed into train and test datasets. The *rpart()* function is used to create the decision trees and takes the following arguments - 1) formula (function that needs to be predicted - dependent variable ~ set of independent variable), 2) data, in this case it is *train*, 3) method, in this case it is "class" (for classification). Rpart uses Gini Index to split the node and keeps growing the tree until the stopping criteria are reached. Furthermore, *rpart* by default applies 10-fold cross-validation to find optimal complexity parameter, *cp*, values. The value of *cp* is investigated to determine how to prune the tree. Since smaller *cp* values tend to increase the tree size, the *cp* value is checked for the default *cp* =0.01 to make a simple but powerful tree. The *cp* values and the errors are captured in a plot, using *plotcp()*, and the best *cp* value is decided for pruning. The fitted model is plotted using *rpart.plot()* to see the decision tree generated. *gender* feature is also used to check if gender plays any role along with the Action Units in determining Personality traits. Below is the code 4.8 for reference.

```
1 # apply decision tree on the training dataset
2 e_dtree <- rpart(formula = extraversion_score~AU01+AU02+AU05+
3                 AU06+AU07+AU09+AU10+AU11+AU12+AU14+AU15+AU17+
4                 AU20+AU23+AU24+AU25+AU26+AU28+AU43+gender,
5                 data = train,
6                 method = 'class')
7 rpart.plot(e_dtree, box.palette = "RdBu", digits = -3)
8
9 # calculate the lowest error for cp
10 bestcp<- e_dtree$cptable[which.min(e_dtree$cptable[,
11 "xerror"]), "CP"]
```

Source Code 4.8: Random Forest model with default parameters for extroversion personality traits

After getting the trained model, predictions are made on the test dataset, and a confusion matrix is generated to measure the performance of the tree. These values are stored in a table and will later be compared with the tuned model to measure model accuracy performance.

As discussed in the 3.4.1, multiple hyperparameters in decision trees can be controlled to improve model accuracy. Rpart offers the functionality via *rpart.control()* which takes on argument for *minsplit* (default =20), *minbucket* (default = round(*minsplit*/3)), *cp* (default=0.01), *maxdepth*. These argument values are tweaked and the above process is repeated to check whether the accuracy increased. The results are discussed in the 5.2.

4.3.2 Random Forest

The Random Forest model is generated using "*randomForest*" library of R [Liaw and Wiener, 2002] is used. The model is passed train dataset with default settings as generally random forest performs very well even on default settings.

The default model is created using *randomForest* function. The parameters passed are the *formula* (function that needs to be predicted - dependent variable ~ set of independent variable). In the current scenario, dependent variables are personality traits, factor variables; thus, the model runs as classification. The independent variables passed are all the action units and gender. The training dataset is passed to the *data* argument, and the *importance* argument to assess the importance of independent features is set as TRUE. Since the features passed are 18, thus the default value taken for $m_{try} = 4$ and $ntrees = 500$. The code is shown in 4.9. The same code is run for other personality traits, respectively, with the same values - (neuroticism, openness, agreeableness, conscientious)

```
1 # random forest default model
2 rf_extraversion1 <- randomForest(extraversion_score~AU01+AU02+AU05+
3     AU06+AU07+AU09+AU10+AU11+AU12+AU14+AU15+AU17+
4     AU20+AU23+AU24+AU25+AU26+AU28+AU43 + gender,
5     data = train,
6     proximity = TRUE,
7     importance =TRUE)
```

Source Code 4.9: Random Forest model with default parameters for extroversion personality traits

The accuracy results of each personality trait's default models are stored in a table and used to compare the model performance after hyperparameter tuning.

There are two variables for hyperparameter tuning - $ntrees$ and m_{try} . For finding best m_{try} . To tune the m_{try} *tuneRF()* function is used and is given below 4.10 for reference. The parameters passed are the independent and dependent variables, along with *stepFactor* indicating value to inflate m_{try} , *improve* indicates the min improvement required in OOB error and *plot* is used to plot the m_{try} error graph. The value for $ntrees$ can be selected by plotting the model graph. The columns from 4:23 represent action units, and 32 indicates gender feature.

```
1 # Hyper-parameter tuning
2 t <- tuneRF(train[,c(4:23, 32)], train[,33],
3     stepFactor = 0.8,
4     plot=TRUE,
```

```
5     ntree = 400,  
6     trace = TRUE,  
7     improve = 0.05)
```

Source Code 4.10: Tuning m_{try}

After selecting the best value of hyperparameters, the model is trained again, and predictions are made on the test dataset. Finally, the exact process is applied to each personality trait. The results generated and the final values of hyper-parameters taken are explained in the Evaluation chapter 5.1.

4.4 Conclusion

This chapter went into depth describing the implementation of the work done for this research. It details the reason for choosing specific algorithms to give the freedom and opinion to change any part of that algorithm in future related work. Finally, it describes the procedure in sufficient detail for Python and R language replication. The next chapter will thoroughly evaluate the results generated from the experiments discussed in this chapter.

5 Evaluation

This chapter reviews the findings of the tests conducted in the previous chapter to comprehend their efficacy. It starts by analyzing the produced data set that is tested using a variety of metrics. In addition, it will describe the cross-validation and assessment of the machine learning models provided in the preceding chapter following industry-standard machine learning techniques described in 3.5.2

5.1 Random Forest

The model generated for Random Forest is run for each personality trait and is tuned for improving accuracy. The results are recorded and discussed in respective subsections. Variable importance and accuracy of the model are also recorded. The statistical significance test, Mann-Whitney, is also applied to the final results generated for verification. As discussed in section 3.4.3 for variable importance, both the parameters are considered, MDA and MDG. As both measures use different approaches to calculate feature importance, the list of variable importance generated by both differs. As a general thumb of rule for this research, the number of top variables selected for each personality trait is kept close to the final hyper-tuned value of m_{try} and an attempt is made to select variables that are important in both MDA and MDG. However, since the independent features are correlated with all the variables being numeric and the research aims to determine the most potent predictors, thus MDG is given more preference over MDA.

5.1.1 Openness

To produce results for the Personality Trait Openness through Random Forest Model, a default random forest model (refer Code 4.9) is run on the train dataset with default values of $m_{try} = 4$ and $ntrees = 500$. The OOB error generated is - 4.01% indicating an accuracy rate in the OOB data of 95.99%. The model is then hyper tuned to decrease the class error generated through this model. A plot is drawn against the number of trees vs error generated to find optimal $ntrees$ value and can be seen in Fig 5.1. Error is lowest at values around 500, and thus $ntrees=500$ is selected. To hyper tune m_{try} , different values of m_{try} are tried by changing the *stepFactor* argument in the code 4.10. The values that increase

the accuracy by *improve* are only generated. The OOB error is recorded in the table 5.1. The lowest value = 5.99 is produced by $m_{try} = 5$, and thus the same is taken to generate the final model. The confusion matrix generated by both the models is presented in table 5.2 for better comparing the differences. As can be observed in the table, the class error decreases for both the classes in the hyper-tuned model.

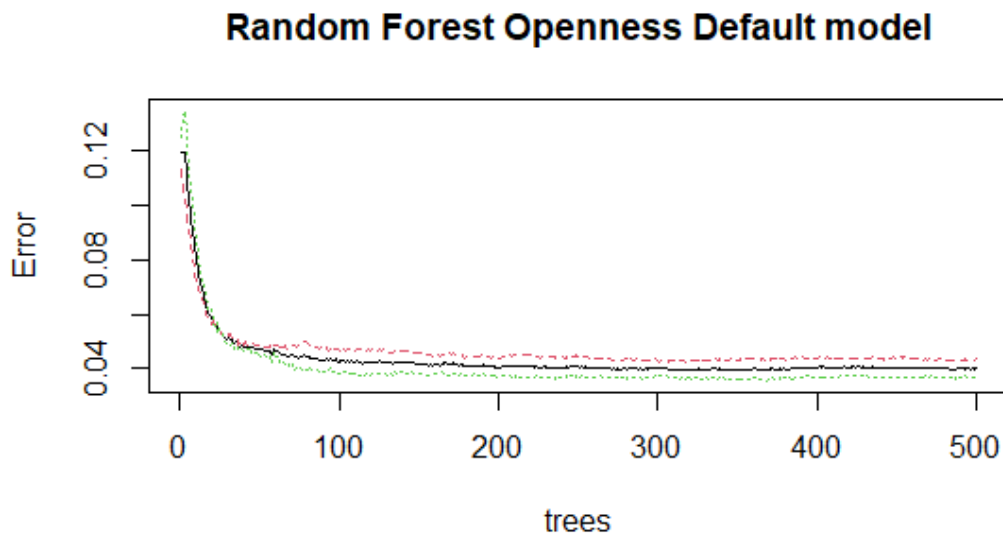


Figure 5.1: Error vs trees for Openness - Random Forest

m_{try}	OOB error (in %)
1	6.69
2	6
3	6.06
4	6.09
5	5.99
8	6.03

Table 5.1: m_{try} values for Openness RF model

The final model predictions are run on the test dataset, and the accuracy increases from 91.25% to 91.41%. The sensitivity for "High" class also increases (refer Table 5.3). Variable importance is plotted and can be seen in Fig 5.2. The MDA and MDG for *Gender* are the highest, thus making *Gender* the strongest predictor for Openness for this dataset. AU01 is approximately 0.8 times less significant than Gender, according to MDA and MDG. The following important feature suggested in the plot is AU28, which is 0.5 times as significant as gender w.r.t. MDA, whereas only 0.3 times less than w.r.t. MDI. Feature AU02 hardly contributes to the accuracy of the model but is important w.r.t. MDG and thus considered. Although AU43 is the next important feature according to MDG, it contributes comparatively less to the gini decrease and the accuracy decrease. The same goes for AU15

Confusion matrix for default model:			
	High	Low	class.error
High	5831	265	0.043471
Low	224	5867	0.036745
Confusion matrix for hypertuned model:			
High	5836	260	0.042651
Low	221	5870	0.036253

Table 5.2: Confusion matrix for default and hypertuned RF model - Openness

Parameters	RF Default	RF Tuned
ntrees	500	500
m_{try}	4	5
OOB error	4.01	3.95
Accuracy on train	95.99	96.05
Accuracy on test	91.25	91.41
Sensitivity "High"	90.62	90.77

Table 5.3: Openness RF model Accuracy

in the MDA plot. Thus the final list of features considered for Openness is - ["Gender", "AU01"]. The other variables contribute to the decrease of gini and accuracy comparatively less.

Partial dependency plots are generated for gender and AU01 to understand what values help the split and improve the accuracy. The male class was found to have high Openness personality traits as compared to "Female" in this multimodal dataset (refer fig 5.3). For AU01, values less than 0.5 are a good predictor of Openness.

5.1.2 Conscientiousness

The same procedures as above are repeated to create results for Conscientiousness. A default model is first trained on train dataset with $m_{try} = 4$ and $ntrees = 500$. These combinations generated an OOB error of 4.95%, yielding an accuracy of 95.05% on the OOB set. The model is hyper tuned, and m_{try} with the least value of OOB is selected from the table 5.4. As can be seen, the least OOB value (highlighted) is the default value, 4. Thus the same model will be used to make the predictions on the test dataset.

The model receives an accuracy of 89.95% on the test dataset with a sensitivity score of 92.1% for the "High" class. The variable importance is plotted for the variables (ref Fig 5.5). Following the same concept used for Openness, the plot of MDG is observed, and the top features are selected. The respective MDA value is also noted. The variables having both comparatively high values are selected. Thus the features selected following the above mentioned concept are - ["AU28", "AU14", "AU02", "AU05", "AU01", "AU07"].

Variable importance for Openness - RF

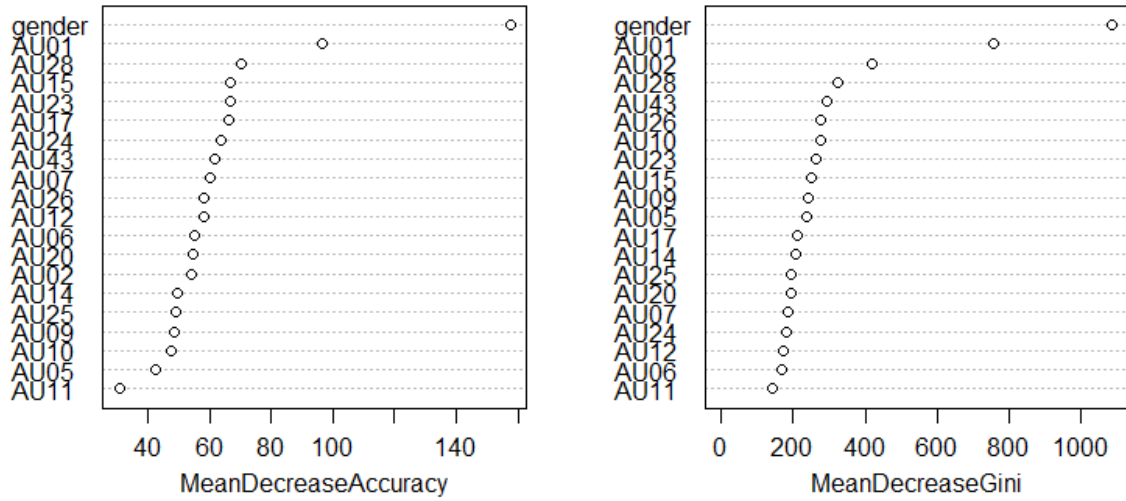


Figure 5.2: Variable Importance Openness - Random Forest

Gender dependency on Openness - RF

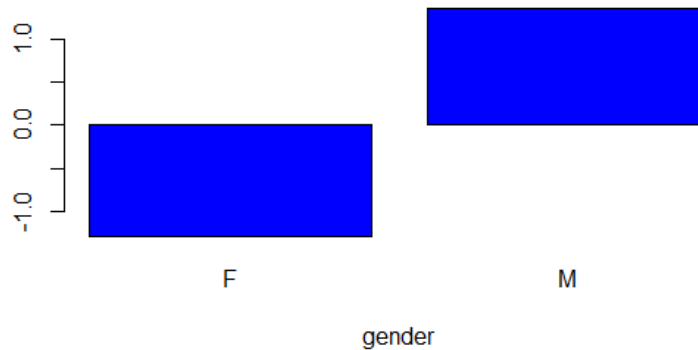


Figure 5.3: Dependency of Gender on Openness

m_{try}	OOB error (in %)
1	6.74
2	6.22
3	6.12
4	5.91
5	6.01
8	5.94
10	5.93

Table 5.4: m_{try} values for Conscientiousness RF model

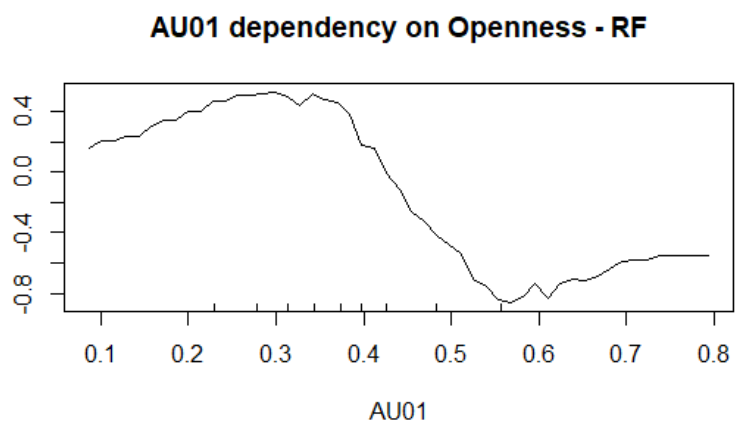


Figure 5.4: Dependency of AU01 on Openness

Variable importance for conscientiousness - RF

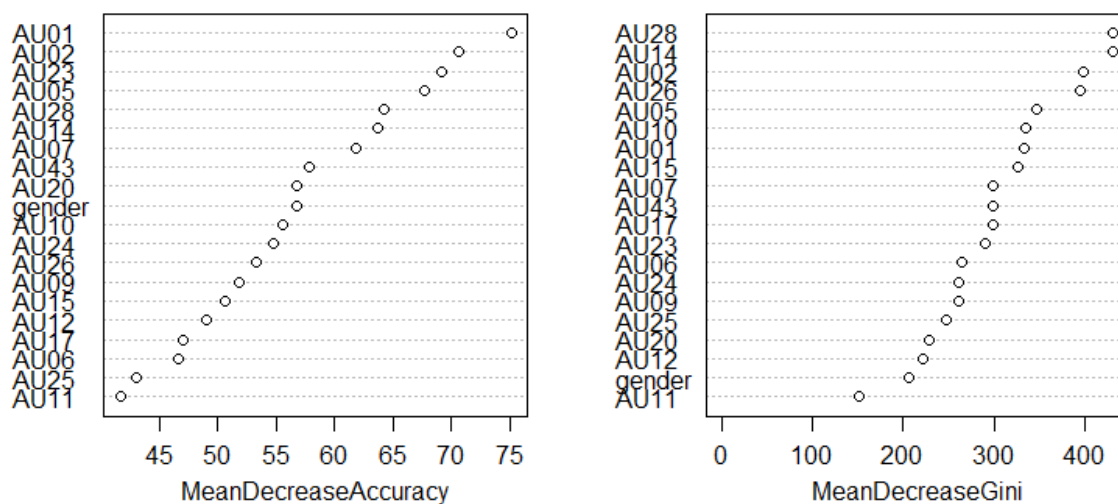


Figure 5.5: Variable Importance Conscientiousness - Random Forest

5.1.3 Extraversion

The code 4.9 was run for Extraversion, and the OOB error obtained was 6.22%. The confusion matrix generated is shared 5.5.

	High	Low	Class error
High	5193	447	0.07925532
Low	311	6236	0.04750267

Table 5.5: Confusion Matrix for Extraversion Random Forest Default Model

The model is plotted to check for the best fit values for the trees and can be seen in Fig 5.6. Error decreases significantly as the number of trees increases, and after 300 trees, the error values become constant and start declining again at around 500 trees. Thus the value selected for $n_{trees} = 500$, which is also the default value.

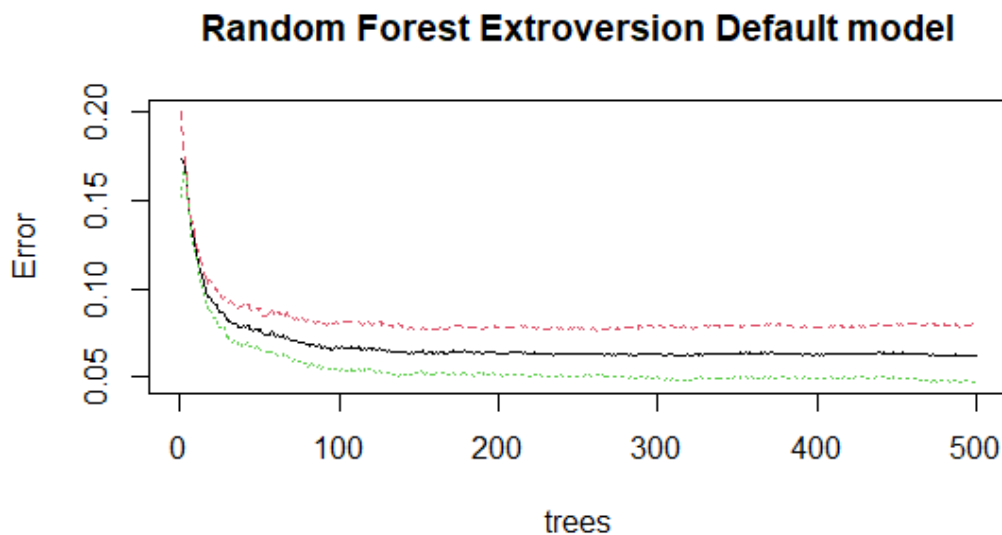


Figure 5.6: error vs trees for Extraversion - Random Forest

After selecting the n_{trees} value, the model is hyper-tuned for m_{try} . The results can be seen in table 5.6. As can be observed, the OOB error value for 5 is the lowest and is considered in the final model.

Running the model with final values $m_{try} = 5$ and $n_{trees} = 500$ results in an OOB error of 6.2%. The accuracy on the test dataset is 86.53% which is less than the default model, but the class error rate reduced, and predictions for the "High" class label improved. The table 5.7 can be seen for reference.

A plot is drawn to see the importance of variables in the model and can be seen in Fig 5.7. The variables with highest dependency selected are - ["AU02", "AU01", "AU15", "AU17", "AU23"].

m_{try}	OOB error (in%)
1	6.74
2	6.2
4	5.9
5	5.89
6	6.06
8	6.02

Table 5.6: m_{try} values for Extraversion RF model

Parameters	RF Default	RF Tuned
ntrees	500	500
m_{try}	4	5
OOB error	6.22	6.2
Accuracy on train	93.78	93.8
Accuracy on test	86.62	86.53
Sensitivity "High"	83.87	84.18

Table 5.7: Extraversion RF model Accuracy

Variable importance for Extroversion - RF

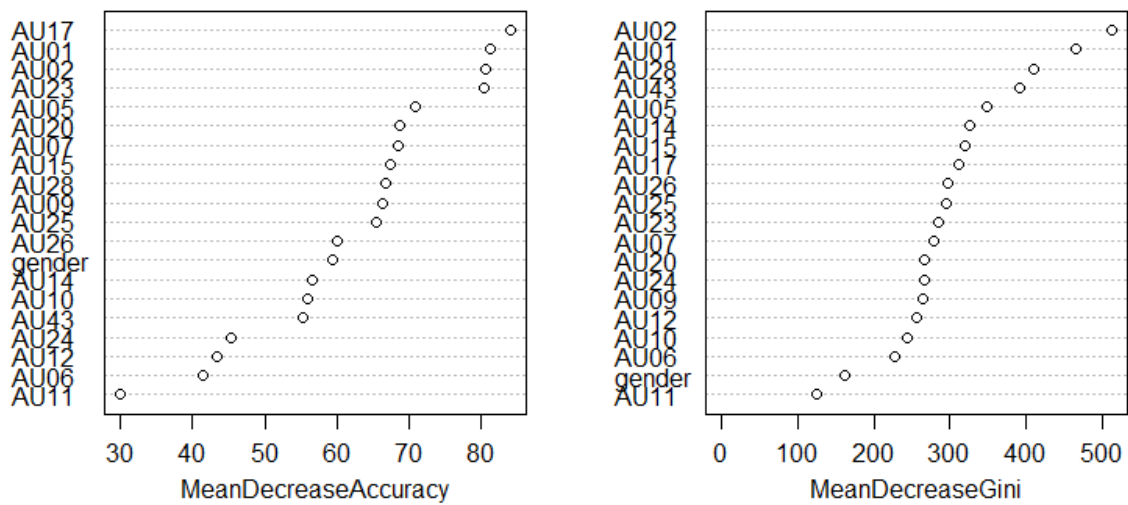


Figure 5.7: Variable Importance Extraversion - Random Forest

5.1.4 Agreeableness

To predict Agreeableness and find important variables, first a default model is run with $m_{try} = 4$ and $ntrees = 500$. The OOB error obtained is 5.32%. The model is hyper tuned and as can be seen in the table 5.8 the least OOB error is for $m_{try} = 5$.

m_{try}	OOB error (in %)
2	6.28
3	6.06
4	6.12
5	5.99
8	6

Table 5.8: m_{try} values for Agreeableness RF model

The new model is run with $m_{try} = 5$ and $ntrees = 500$. The model shows a decrease in the OOB error with new values being 5.58%. The accuracy of the new model on test dataset is 89.46% (refer Table 5.9). The error in predicting the "High" class is decreased, increasing the sensitivity for that class, as seen in the comparative confusion matrix shown in table 5.10. On the contrary, error increased for the "Low" class.

Parameters	RF Default	RF HyperTuned
$ntrees$	500	500
m_{try}	4	5
OOB error	5.32	5.28
Accuracy on train	94.68	94.72
Accuracy on test	89.21	89.46
Sensitivity "High"	89.26	89.32

Table 5.9: Agreeableness RF Model Accuracy

Confusion matrix for default model:			
	High	Low	class.error
High	5595	328	0.055377
Low	320	5944	0.051086
Confusion matrix for hypertuned model:			
High	5611	312	0.052676
Low	332	5932	0.053001

Table 5.10: Confusion matrix for default and hypertuned RF model - Agreeableness

The variable importance is plotted in Fig 5.8 and can be seen that ["AU28", "AU05", "AU01", "AU15", "AU14"] are strong predictors for Agreeableness.

Variable importance for Agreeableness - RF

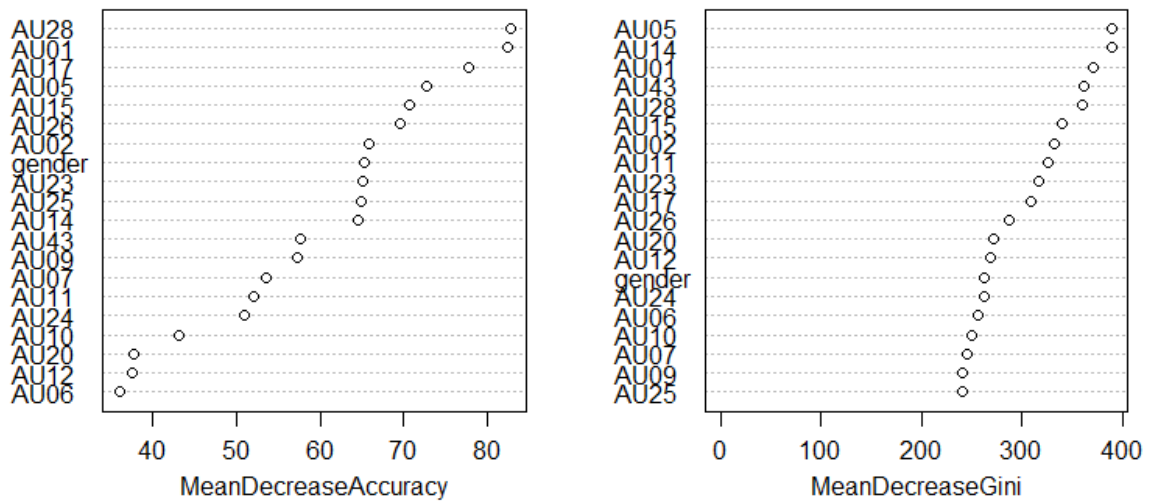


Figure 5.8: Variable Importance Agreeableness - Random Forest

5.1.5 Neuroticism

For Neuroticism, the model is trained on default values and then predicted on the test dataset. The OOB error on the training dataset is 5.46%, and the accuracy on the test dataset is 88.41%. For determining the ntrees, a plot has been drawn against error and no of trees (refer Fig 5.9). The error is lowest at the default value of 500. Thus the final value taken for ntrees is 500. The model is hyper tuned, and as seen in table 5.11, there are two values for the least OOB error, 4 and 5. As explained in segment 3.4.2, the data has enough relevant predictors, so a lower value of m_{try} is preferred, i.e., 4.

m_{try}	OOB error (in %)
1	6.62
2	6.22
3	6.01
4	5.92
5	5.92
8	6.05
10	6.01

Table 5.11: m_{try} values for Neuroticism RF model

For the Neuroticism model, the default values have the best accuracy of 88.41% on the test dataset with a sensitivity for the "High" label - 89.18%. The final values are shown in the table 5.12 and the confusion matrix is shown in table 5.13. Variable importance is checked, and the results in Fig 5.10 show that AU14 and AU28 have the highest dependency. Also,

Random Forest Neuroticism Default model

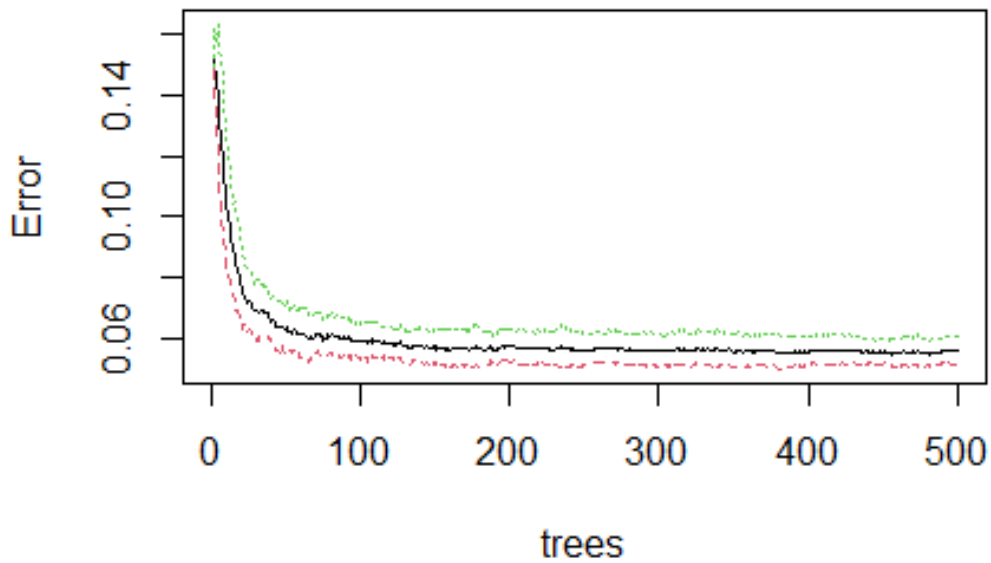


Figure 5.9: Error vs trees for Neuroticism - Random Forest

gender variable plays a role in improving accuracy. The final selected features are - ["AU14", "AU28", "AU02", "AU05", "AU15"]

Parameters	RF Default
ntrees	500
m_{try}	4
OOB error	5.46
Accuracy on train	94.54
Accuracy on test	88.41
Sensitivity "High"	89.18

Table 5.12: Neuroticism RF Model Accuracy

5.2 Decision Trees

The model generated for Decision Trees is run for each personality trait and is tuned for improving accuracy. Finally, the results are recorded and discussed in respective subsections.

	High	Low	class.error
High	6168	312	0.048148
Low	354	5353	0.062029

Table 5.13: Confusion Matrix - Neuroticism RF Model

Variable importance for Neuroticism - RF

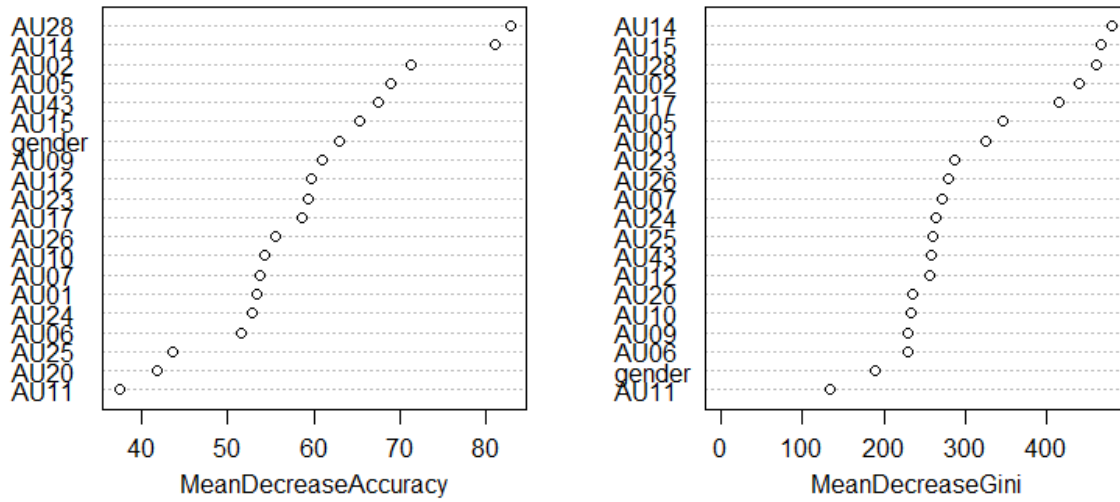


Figure 5.10: Variable Importance Neuroticism - Random Forest

5.2.1 Openness

As discussed in the Implementation part of Decision Trees in section 4.3.1, a default model is first to run, passing the training dataset, and the accuracy score is recorded. Later, the model is hyper-tuned to improve accuracy results. Code 4.8 is run for the dependent variable *openness_score* with independent feature set consisting of all the action units and gender. The accuracy of the unpruned tree is recorded in Table 5.14. A plot is generated for cp vs error to investigate the most optimal value for cp (as increasing cp leads to an increase in tree size). After observing Fig 5.11, the best cp value for the tree is taken as 0.001. After selecting the cp value, different values of minsplit and minbucket were tried. However, as seen in Table 5.14, accuracy does not change even after changing the values of minsplit. The value of maxdepth was kept constant at 5, so the tree does not grow to overfit and ensure the tree is still readable. However, even values of maxdepth 4 and 5 did not cause much difference in the accuracy, so in the final model, it was further reduced to 4. The final values taken are cp = 0.001, maxdepth =4 and minsplit = 150.

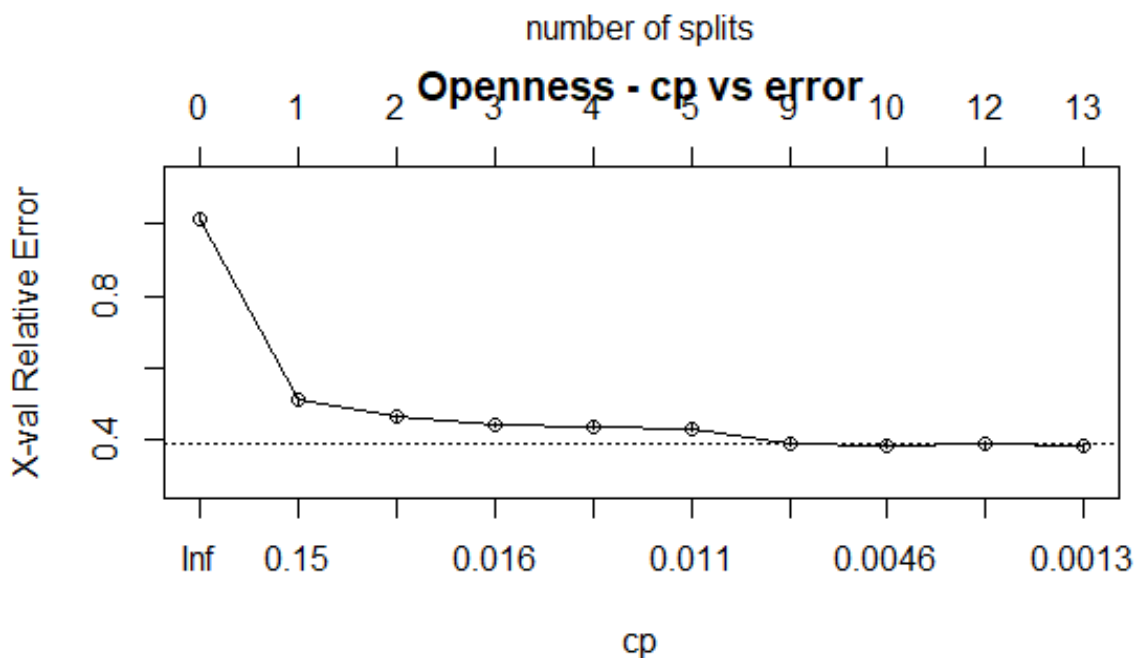


Figure 5.11: Cp vs error - Openness

	Tree Default	Tree Tuned					
cp	0.01	0.01	0.01	0.001	0.001	0.005	0.005
minsplits	20	4	150	180	150	240	180
minbucket	6.67	1.33	50	60	50	80	60
accuracy	79.74	79.74	79.74	80.3	80.3	80.02	80.02

Table 5.14: Openness Decision Tree Accuracy

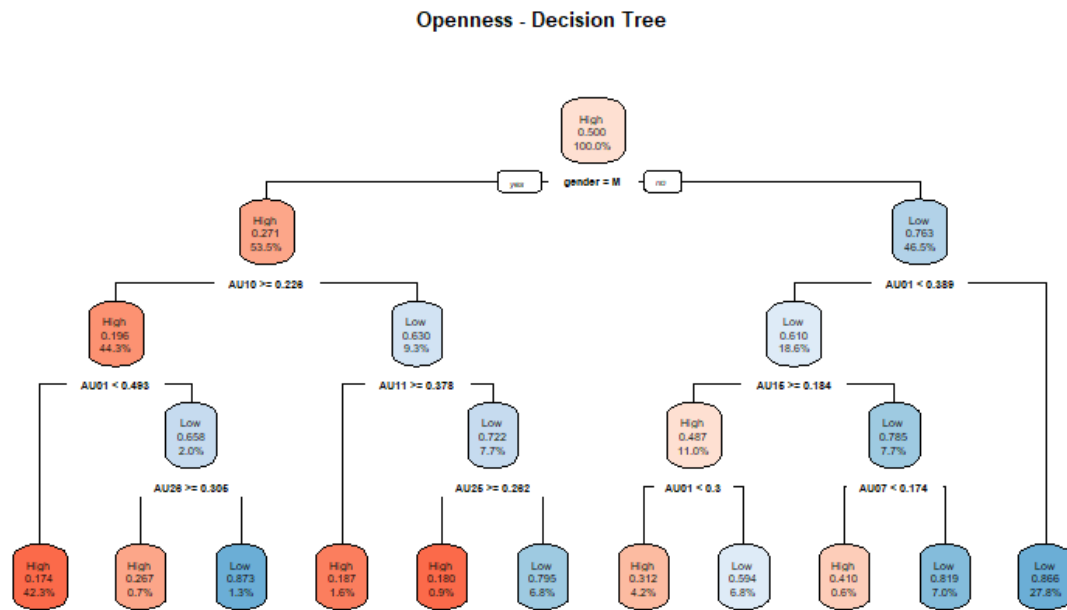


Figure 5.12: Decision Tree - Openness

The final decision tree is present in Fig 5.12. As can be observed, the root node shows that the probability of getting a High percentile score in Openness for this dataset is 50%. The first split is done on the feature *Gender*. If the gender is *Female*, then there is a 76% chance that the Openness personality trait percentile is below 50%, i.e. "Low". However, if the gender is *Male*, then further splits are required to decide based on AU10 and later AU01 and AU11. Thus the following variables are used - ["Gender", "AU10", "AU01", "AU11", "AU28", "AU26", "AU07"].

5.2.2 Conscientiousness

For the conscientiousness personality trait, a tree is trained the same way as above. Different values of *cp* are checked by the `rpart()` using cross-validation. The corresponding error is plotted in the Fig 5.13. As can be observed that *cp* = 0.001 yields the most optimal result. The minimum split taken is 14. The model is trained, and predictions are run on the test dataset. The accuracy increases from 71% to 73% as can be seen in Table 5.15

	Tree Default	Tree Tuned
<i>cp</i>	0.01	0.001
<i>minsplit</i>	20	14
<i>minbucket</i>	6.67	4.67
<i>accuracy</i>	71.49	73.31

Table 5.15: Conscientiousness Decision Tree Accuracy

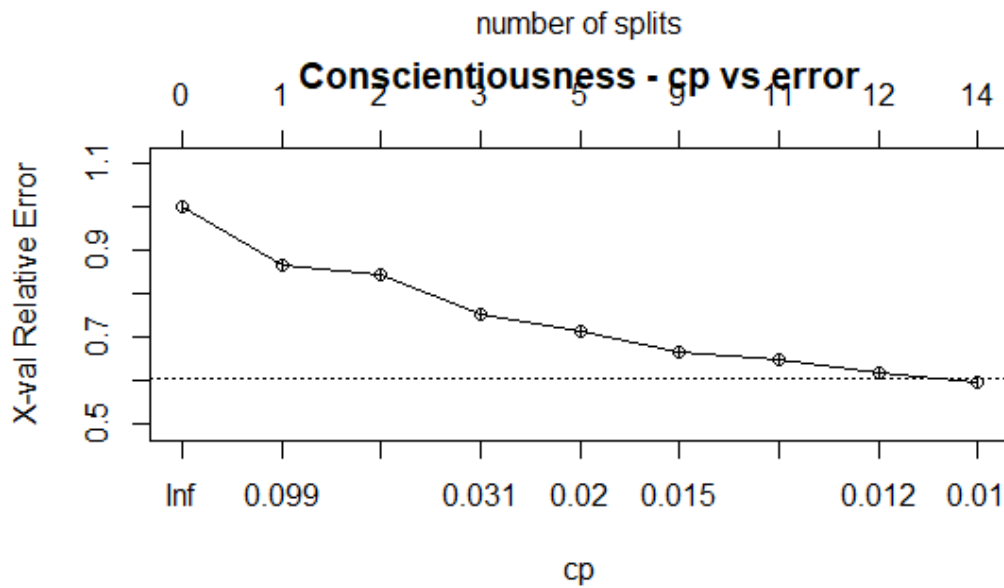


Figure 5.13: Cp vs error - Conscientiousness

	Tree Default	Tree Tuned	
cp	0.01	0.001	0.005
minsplit	120	240	180
maxdepth	5	6	5
minbucket	40	80	60
accuracy	64.12	64.12	64.12

Table 5.16: Extraversion Decision Tree Accuracy

A decision tree is plotted for the tuned model and is shown in Fig 5.14. As can be observed, the first split is done based on AU43. If the value of AU43 is less than 0.11 and AU14 is greater than 0.45, the chances of having a High class is 16%. The same way other nodes can be interpreted as well. The variables used for building the classification tree are - ["AU05", "AU06", "AU09", "AU11", "AU14", "AU43"].

5.2.3 Extraversion

For extraversion, a tree is built with default values of minsplit = 20, and the accuracy obtained is 65.40%. Another decision tree is plotted with maxdepth = 5 and cp=0.001. The new tree generated is plotted in Fig 5.15. The variables used in the construction of the tree are - ["AU15", "AU02", "AU43", "AU05"]. The accuracy obtained after hyper-tuning the parameters does not change even after trying multiple parameter combinations. (refer Table 5.15)

Conscientious - Decision Tree

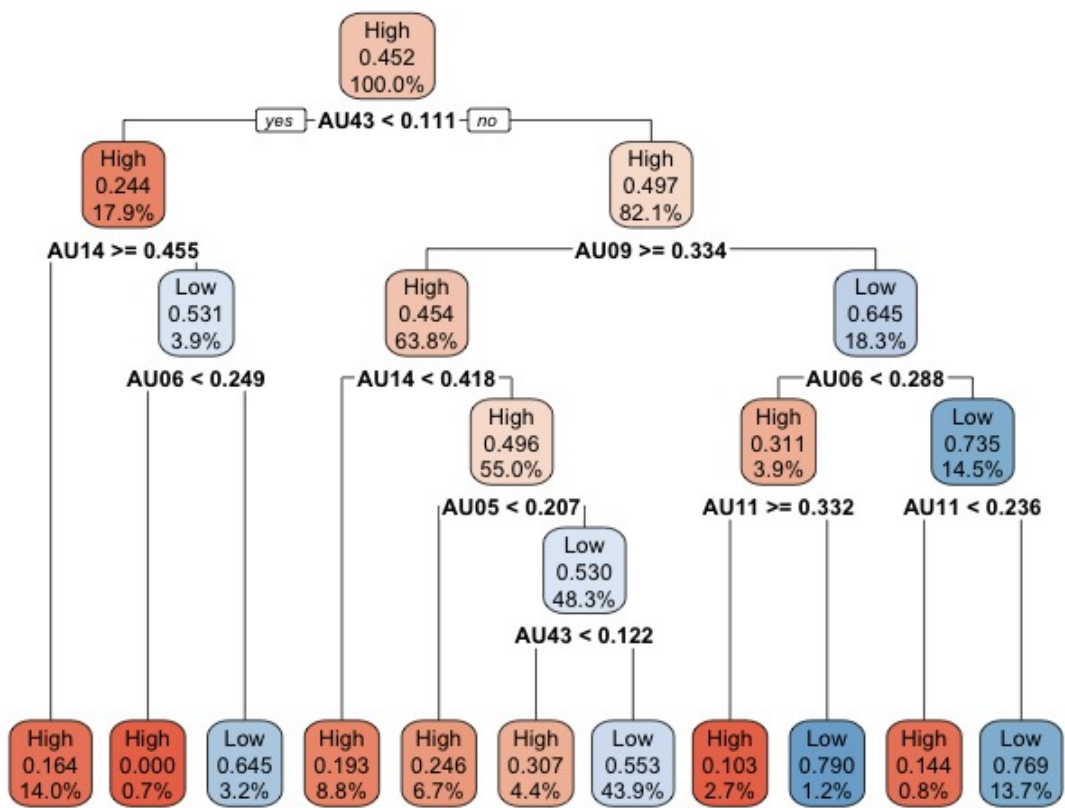


Figure 5.14: Decision Tree - Conscientiousness

Extraversion - Decision Tree

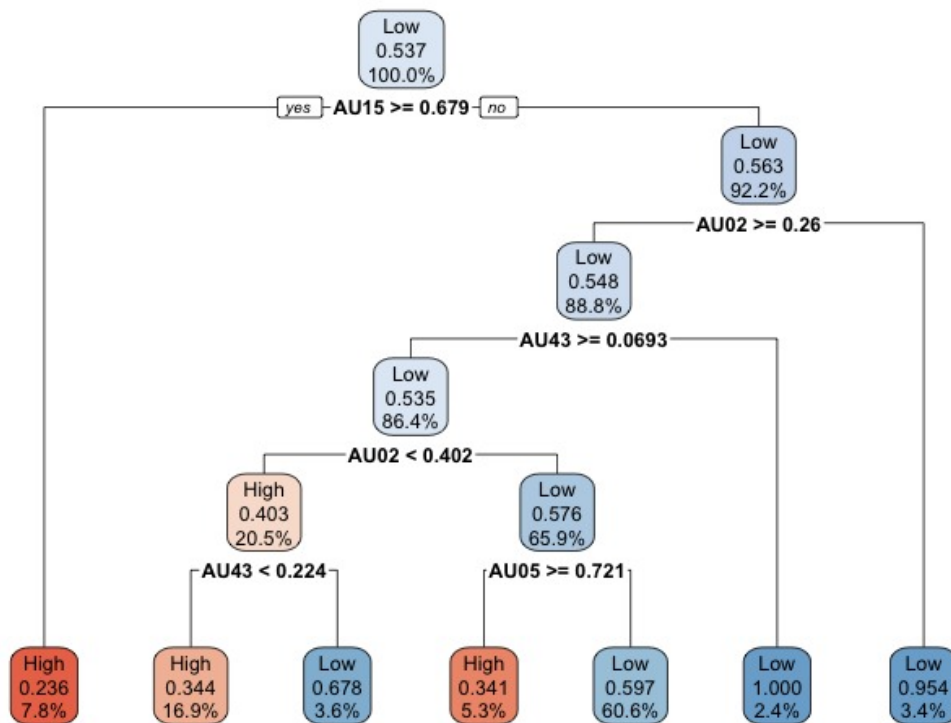


Figure 5.15: Decision Tree - Extraversion

5.2.4 Agreeableness

The same process is replicated for Agreeableness, and the final parameters are considered. The corresponding accuracy is mentioned in Table 5.17 and the final generated tree is shown in Fig 5.17. The feature list considered for generating the tree are - ["AU12", "AU11", "AU10", "AU20", "AU25", "AU24", "AU05"]

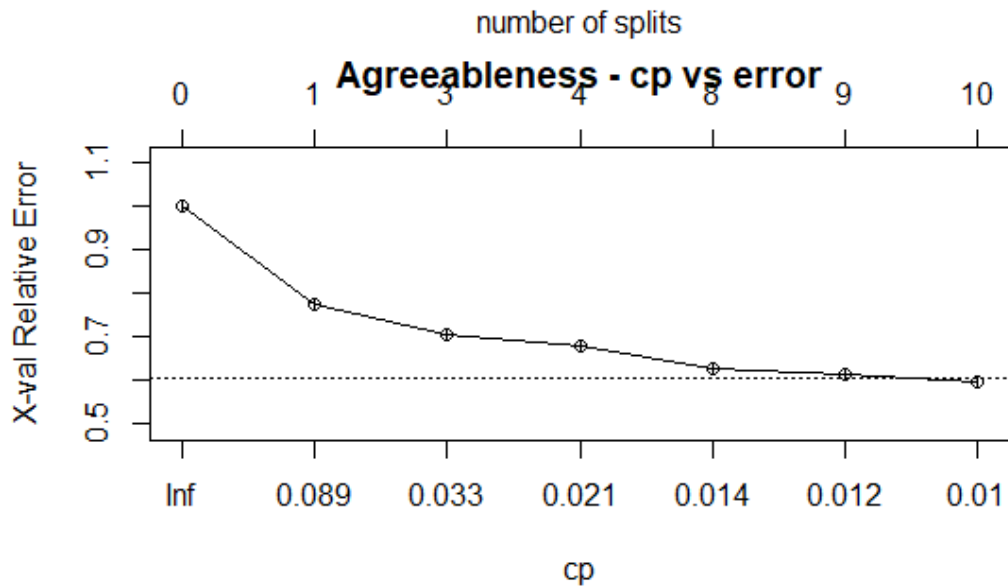


Figure 5.16: Cp vs error - Agreeableness

	Tree Default	Tree Tuned	
cp	0.01	0.001	0.005
minsplit	150	150	150
minbucket	50	50	50
accuracy	72.95	72.99	73.1

Table 5.17: Agreeableness Decision Tree Accuracy

5.2.5 Neuroticism

The same concept is used for Neuroticism, and the final parameters are considered. The corresponding accuracy is mentioned in Table 5.18 and the final generated tree is shown in Fig 5.19. The feature list considered for generating the tree are - ["AU25", "AU20", "AU11", "AU26", "AU06", "AU43"]

Agreeableness - Decision Tree

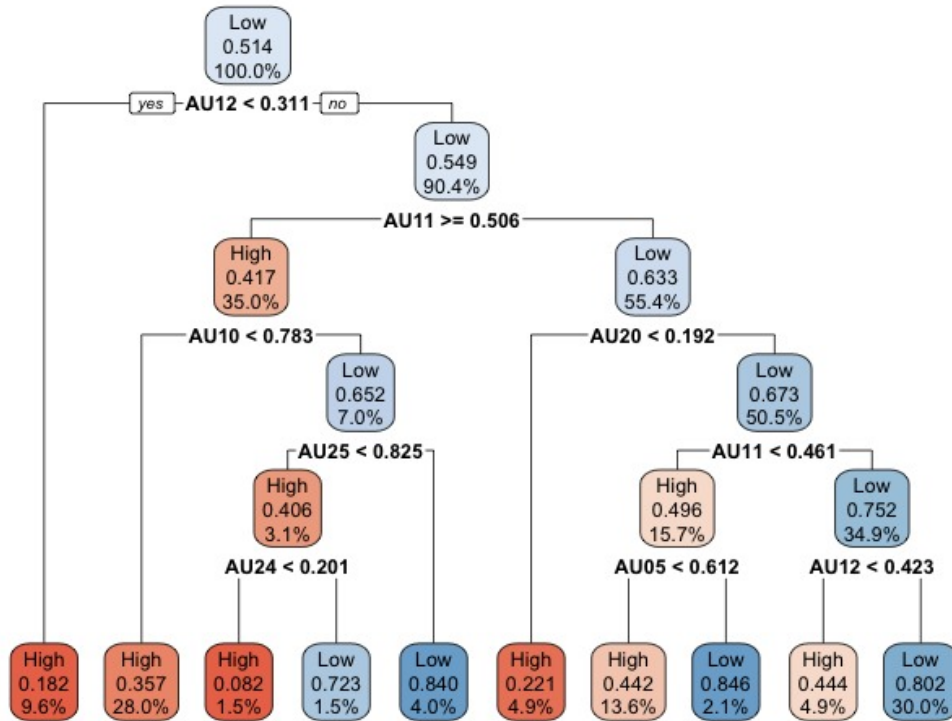


Figure 5.17: Decision Tree - Agreeableness

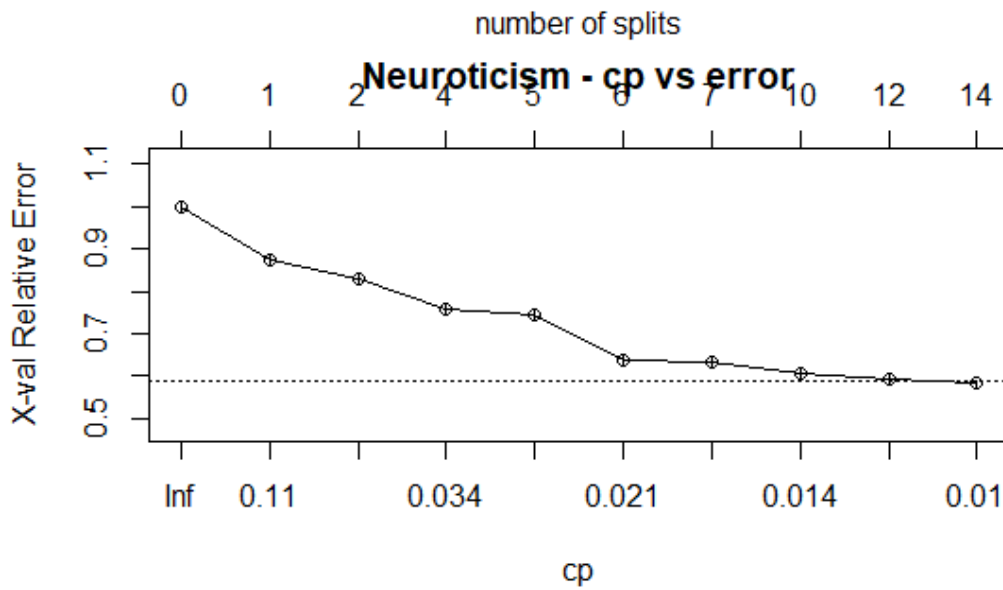


Figure 5.18: Cp vs error - Neuroticism

	Tree Default	Tree Tuned	
cp	0.01	0.001	0.005
minsplit	20	14	50
minbucket	6.67	4.67	16.67
accuracy	77.89	78.05	78.01

Table 5.18: Neuroticism Decision Tree Accuracy

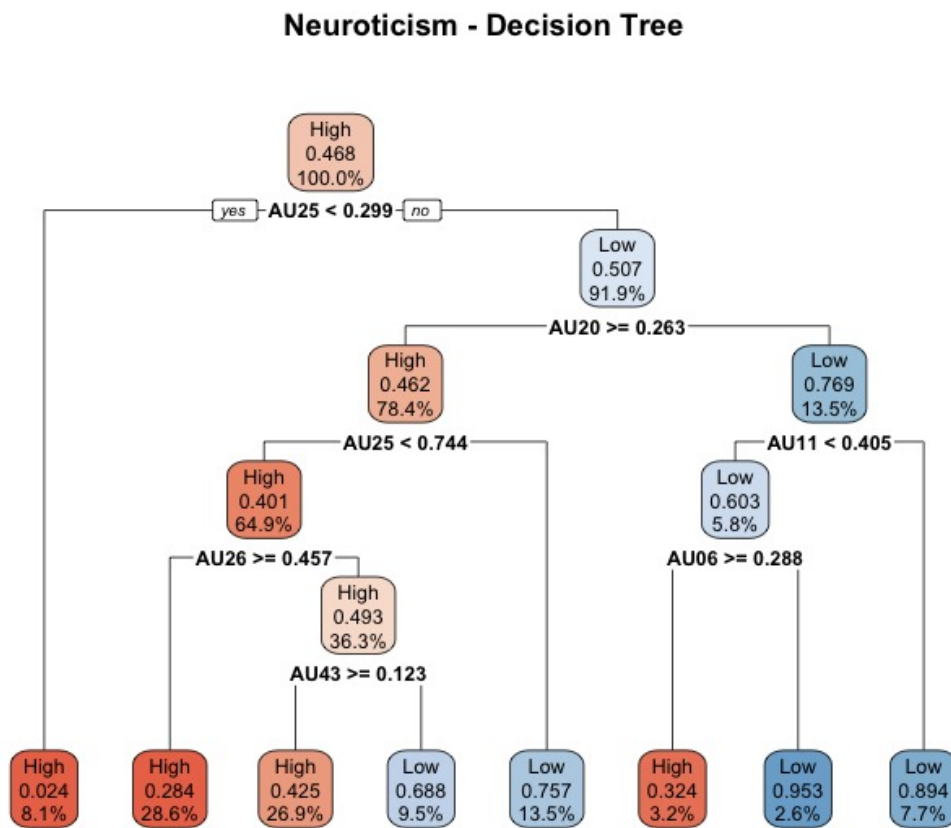


Figure 5.19: Decision Tree - Neuroticism

5.3 Conclusion

This chapter discussed the performance of all the machine learning algorithms used and which one would be the best to use to predict different personality traits. It also sheds light on the relationship between different AUs and Personality Traits and if personality traits can be identified using Facial Action Units. The main final findings of the experiment and any challenges faced or future work required are discussed in the next chapter.

6 Conclusion

In the previous chapter, Evaluation results were generated for the Random Forest and Decision Tree models. This section discusses the final findings of those results and confirms the validity of the results produced. The research objectives posed at the start of the thesis and their solution received from this experiments are discussed. Later, Challenges or roadblocks faced in the course of this research are discussed, ending the chapter with Future Work required that might be required in the field is shared.

6.1 Discussions

After running both the models for each personality trait, the final results generated showed that random forest performed significantly better than Decision Trees. Thus the feature list used to build random forests (variable importance) is verified by applying statistical significance test - Mann Whitney U. The significance value is taken as 0.05. This process is done for each personality trait and the results are shared below.

6.1.1 Openness

For Openness, the feature list generated through RF is checked for the Mann-Whitney U test to verify the results. The outcomes can be seen in Table 6.1. Only gender and AU01 showed variable importance and after running Mann-Whitney on AU01, the null hypothesis can be rejected as $p\text{-value} < 0.05$. Therefore, the median of AU01 differed for High and Low class of Openness, suggesting that AU01 values act as a predictor for the personality trait Openness. Referring Fig 5.12, it can be seen that the decision tree also suggested that a Male having lower values of AU01 was classified as having High Openness scores, thus confirming the results.

Aus	P-value	Null Hypothesis
AU01	$< 2.2e-16$	Rejected

Table 6.1: Action Units vs Openness - Mann Whitney U test

6.1.2 Conscientiousness

For Conscientiousness, the same procedure is repeated as above, and for all the action units, the null hypothesis is rejected. This suggests that different values of AU01, AU02, AU14, AU28, AU05 and AU07 can predict Conscientiousness High Low class. Although, no assumption can be made on how these action unit values interact to predict Conscientiousness.

Aus	P-value	Null Hypothesis
AU01	0.000309	Rejected
AU02	< 2.2e-16	Rejected
AU14	< 2.2e-16	Rejected
AU28	< 2.2e-16	Rejected
AU05	6.84E-10	Rejected
AU07	7.65E-10	Rejected

Table 6.2: Action Units vs Conscientiousness - Mann Whitney U test

6.1.3 Extraversion

For Extraversion - according to Mann-Whitney U Test AU01, AU02, AU15, AU17 and AU23 all reject the null hypothesis, suggesting that different values of these action units can help classify Extraversion scores.

Aus	P-value	Null Hypothesis
AU01	< 2.2e-16	Rejected
AU02	< 2.2e-16	Rejected
AU15	2.84E-14	Rejected
AU17	4.81E-03	Rejected
AU23	9.22E-11	Rejected

Table 6.3: Action Units vs Extraversion - Mann Whitney U test

6.1.4 Agreeableness

When the same test was run for action units - AU01, AU14, AU28, AU15 and AU05. For AU01 and AU05 p-value > 0.05, thus the null hypothesis cannot be rejected in this case. Although it is to be kept in mind that Agreeableness is an emotion based trait and the research involved task based activity.

6.1.5 Neuroticism

For Neuroticism all the action units rejected the null hypothesis for Mann Whitney U test suggesting that their values are useful to classify into High Low percentile scores for

Aus	P-value	Null Hypothesis
AU01	0.5556	Not Rejected
AU14	< 2.2e-16	Rejected
AU28	3.62E-05	Rejected
AU15	< 2.2e-16	Rejected
AU05	0.07285	Not Rejected

Table 6.4: Action Units vs Agreeableness - Mann Whitney U test

Neuroticism.

Aus	P-value	Null Hypothesis
AU14	< 2.2e-16	Rejected
AU02	< 2.2e-16	Rejected
AU15	< 2.2e-16	Rejected
AU28	< 2.2e-16	Rejected
AU05	4.99E-08	Rejected

Table 6.5: Action Units vs Neuroticism - Mann Whitney U test

6.1.6 Research Finding

The objectives of this research were posed in the section 1.2. The findings of those are mentioned below -

- The experiments conducted in this research showed that few action units can be associated with the Five Factor Personality Traits and can be used in designing APR system. The different facial action units, their meaning and associated personality traits are shown in Table 6.6
- Clubbing objective 2 and 3, two different classifiers were designed in this research. Random Forest showed the best results, with an average of accuracy of above 90% for all personality traits. Other classifier algorithms were considered but none of them would have performed better than Random Forest given the size of data, outliers and the non-normalized distribution of data. Decision trees was designed to see how combinations of action units could determine different traits but the accuracy for the test dataset is low to rely on the trees generated.
- Gender had a significant role in determining High Scores of Openness but for other traits, not that much.

Thereby, the null hypothesis of this research can be accepted as there exists an association between personality traits and facial action units. Thus making it possible to determine personality traits of an individual if their facial action unit data is present.

Action Units	FACS name	Personality Trait
AU01	Inner Brow Raiser	Openness, Conscientiousness, Extraversion, Neuroticism
AU02	Outer Brow Raiser	Conscientiousness, Extraversion, Neuroticism
AU05	Upper Lid Raiser	Conscientiousness, Neuroticism
AU07	Lid Tightener	Conscientiousness
AU14	Dimpler	Conscientiousness, Agreeableness, Neuroticism
AU15	Lip Corner Depressor	Extraversion, Agreeableness
AU17	Chin Raiser	Extraversion
AU23	Lip Tightener	Extraversion
AU28	Lip Suck	Conscientiousness, Agreeableness, Neuroticism

Table 6.6: Actions Units and Associated Personality traits

6.2 Challenges

There were a few challenges faced during the tenure of this research. Some were technical bound whereas some required time to conduct topic based research and thus could not be done and can be a part of Future Work.

Due to computational limits, research could not be performed while taking all the video frames into consideration. Every 30th frame was considered. Nevertheless, there is always an issue while doing video analysis. The video frames have no memory of the previous frames and thus sometimes it is hard to figure out if the action unit was triggered because the participant is pronouncing a particular letter or word. This has a major impact on the facial expressions as well since sometimes the person is not surprised rather saying the word "OH". To accommodate somewhat for this factor, neutral facial expressions were deducted from the facial expressions but still it was still a challenge. If all the frames could have been considered then this issue might have been resolved to some account. Bbut the computational speed to process all made it impossible. Also, certain subtle facial action units that hardly last for half second on the face can also not be taken into account.

For this research, participants filled a personality questionnaire test but one can not be too reliable on humans while judging the assessment test. Psychologist Lisa Barrett in her study stated that if humans are not coached to read human emotions then their judgements are close to random [Baron-Cohen et al., 2001]. In another study, it was claimed that humans are not very good in personality traits perception either [Youyou et al., 2015]. Thus, for the time frame available for this research, only Automatic Personality Recognition factor could be considered for facial action units, rather than looking for both Personality Recognition and Perception.

6.3 Future Work

There is a great scope in future for this research and multiple factors can be added to this research to further enhance the results. Some of them are -

- A column can be added to each frame indicating if the individual is speaking or not. Further study can be done to understand how to balance the facial feature values to accommodate for the speaking factor.
- A baseline facial feature value dataset can be generated for each participant rather than deducting median face value. This requires some research on its own.

This research was heavily reliant on feature engineering steps to improve the results as for this dataset, Random Forest proves to be one of the best classifier. Thus enhancing the feature engineering steps might help in making the results even better.

Bibliography

- [1] Nov 2016. URL <https://www.psychologydiscussion.net/notes/psychology-notes/personality-psychology-notes/personality-methods-of-personality-assessment/2601>.
- [2] Corina Sheerin. Destiny or 'choice': Women in investment management-why so few? *Irish Journal of Management*, 32, 01 2012.
- [3] Kwangeun Ko and Kwee-Bo Sim. Study of emotion recognition based on facial image for emotional rehabilitation biofeedback. *Journal of Institute of Control, Robotics and Systems*, 16(10):957–962, 10 2010. doi: 10.5302/J.ICROS.2010.16.10.957.
- [4] Francesca Lazzeri. How to accelerate devops with machine learning lifecycle management | by francesca lazzeri | microsoft azure | medium, Jun 2019. URL <https://medium.com/microsoftazure/how-to-accelerate-devops-with-machine-learning-lifecycle-management-2ca4c86387a0>.
- [5] Bradley Boehmke Brandon Greenwell. Chapter 11 random forests | hands-on machine learning with r, 02 2020. URL <https://bradleyboehmke.github.io/HOML/random-forest.html>.
- [6] Anuganti Suresh. What is a confusion matrix?. everything you should know about... | by anuganti suresh | analytics vidhya | medium, Jun 2021. URL <https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5>.
- [7] James Uleman, S Saribay, and Celia Gonzalez. Spontaneous inferences, implicit impressions, and implicit theories. *Annual review of psychology*, 59(1):329–60, 02 2008. doi: 10.1146/annurev.psych.59.103006.093707. URL <https://doi.org/10.1146/annurev.psych.59.103006.093707>. PMID: 17854284.
- [8] Rosanna E. Guadagno, Bradley M. Okdie, and Cassie A. Eno. Who blogs? personality predictors of blogging. *Computers in Human Behavior*, 24(5):1993–2004, 2008. ISSN 0747-5632. doi: <https://doi.org/10.1016/j.chb.2007.09.001>. URL <https://www.sciencedirect.com/science/article/pii/S074756320700146X>. Including the Special Issue: Internet Empowerment.

- [9] Sarah Butt and James G. Phillips. Personality and self reported mobile phone use. *Computers in Human Behavior*, 24(2):346–360, 2008. ISSN 0747-5632. doi: <https://doi.org/10.1016/j.chb.2007.01.019>. URL <https://www.sciencedirect.com/science/article/pii/S0747563207000295>. Part Special Issue: Cognition and Exploratory Learning in Digital Age.
- [10] T. E. Yeo. Modeling personality influences on youtube usage. *Proceedings of the International AAAI Conference on Web and Social Media*, 4(1):367–370, May 2010. URL <https://ojs.aaai.org/index.php/ICWSM/article/view/14054>.
- [11] Lin Qiu, Han Lin, Jonathan Ramsay, and Fang Yang. You are what you tweet: Personality expression and perception on twitter. *Journal of Research in Personality*, 46(6):710–718, 2012. ISSN 0092-6566. doi: <https://doi.org/10.1016/j.jrp.2012.08.008>. URL <https://www.sciencedirect.com/science/article/pii/S009265661200133X>.
- [12] Andreas M. Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, 53(1):59–68, 02 2010. ISSN 0007-6813. doi: <https://doi.org/10.1016/j.bushor.2009.09.003>. URL <https://www.sciencedirect.com/science/article/pii/S0007681309001232>.
- [13] Lee Rainie and Barry Wellman. *Networked: The New Social Operating System*. The MIT Press, 04 2012. ISBN 9780262526166. doi: 10.4018/jep.2013040106. URL <http://www.jstor.org/stable/j.ctt5vjq62>.
- [14] Mika Raento, Antti Oulasvirta, and Nathan Eagle. Smartphones: An emerging tool for social scientists. *Sociological Methods Research - SOCIOL METHOD RES*, 37(3):426–454, 02 2009. doi: 10.1177/0049124108330005. URL <https://doi.org/10.1177/0049124108330005>.
- [15] Alessandro Vinciarelli, Maja Pantic, Dirk Heylen, Catherine Pelachaud, Isabella Poggi, Francesca D’Errico, and Marc Sch. Bridging the gap between social animal and unsocial machine: A survey of social signal processing. *IEEE TRANSACTIONS ON AFFECTIVE COMPUTING*, 1(1):69–87, 03 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.27. URL <https://ieeexplore.ieee.org/document/5989788>.
- [16] Alessandro Vinciarelli and Gelareh Mohammadi. A survey of personality computing. *IEEE Transactions on Affective Computing*, 2014. doi: 10.1109/TAFFC.2014.2330816.
- [17] Gerald Matthews, Ian J. Deary, and Martha C. Whiteman. *Personality Traits*. Cambridge University Press, 3 edition, 2009. doi: 10.1017/CBO9780511812743.

- [18] David C. Funder. Personality. *Annual Review of Psychology*, 52(1):197–221, 2001. doi: 10.1146/annurev.psych.52.1.197. URL <https://doi.org/10.1146/annurev.psych.52.1.197>. PMID: 11148304.
- [19] Daniel J. Ozer and Verónica Benet-Martínez. Personality and the prediction of consequential outcomes. *Annual Review of Psychology*, 57(1):401–421, 2006. doi: 10.1146/annurev.psych.57.102904.190127. URL <https://doi.org/10.1146/annurev.psych.57.102904.190127>. PMID: 16318601.
- [20] Egon Brunswik. *Perception and the Representative Design of Psychological Experiments*. University of California Press, Berkeley, 2020. ISBN 9780520350519. doi: doi:10.1525/9780520350519. URL <https://doi.org/10.1525/9780520350519>.
- [21] Klaus Rainer Scherer. *Social markers in speech*, chapter Personality markers in speech, pages 147–209. Cambridge University Press, Cambridge, 1979.
- [22] François Mairesse, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *J. Artif. Int. Res.*, 30(1):457–500, nov 2007. ISSN 1076-9757.
- [23] Alexei Ivanov, Giuseppe Riccardi, Adam Sporka, and Jakub Franc. Recognition of personality traits from human spoken conversations. In *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association*, pages 1549–1552, Florence, Italy, 01 2011. ISCA.
- [24] Fabio Pianesi, Nadia Mana, Alessandro Cappelletti, Bruno Lepri, and Massimo Zancanaro. Multimodal recognition of personality traits in social interactions. In *Proceedings of the 10th International Conference on Multimodal Interfaces, ICMI '08*, page 53–60, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581989. doi: 10.1145/1452392.1452404. URL <https://doi.org/10.1145/1452392.1452404>.
- [25] Ligia Batrinca, Bruno Lepri, Nadia Mana, and Fabio Pianesi. Multimodal recognition of personality traits in human-computer collaborative tasks. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12*, page 39–46, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314671. doi: 10.1145/2388676.2388687. URL <https://doi.org/10.1145/2388676.2388687>.
- [26] Gloria Zen, Bruno Lepri, Elisa Ricci, and Oswald Lanz. Space speaks: Towards socially and personality aware visual surveillance. In *Proceedings of the 1st ACM International Workshop on Multimodal Pervasive Video Analysis, MPVA '10*, page 37–42, New York,

- NY, USA, 2010. Association for Computing Machinery. ISBN 9781450301671. doi: 10.1145/1878039.1878048. URL <https://doi.org/10.1145/1878039.1878048>.
- [27] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 01 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.52. URL <https://ieeexplore.ieee.org/document/4468714>.
- [28] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, Nov 2010. ISSN 1432-1882. doi: 10.1007/s00530-010-0182-0. URL <https://doi.org/10.1007/s00530-010-0182-0>.
- [29] Ian J. Deary. *The trait approach to personality*, page 89–109. Cambridge Handbooks in Psychology. Cambridge University Press, 2009.
- [30] Susan Cloninger. *Conceptual issues in personality theory*, chapter 1, page 3–26. Cambridge Handbooks in Psychology. Cambridge University Press, Cambridge, 05 2009. URL <https://www.cambridge.org/core/books/abs/cambridge-handbook-of-personality-psychology/conceptual-issues-in-personality-theory/6661CB57C3C4E8C3FFD3B4767AC3272D>.
- [31] Gregory J. Boyle. Myers-briggs type indicator (mbti): Some psychometric limitations. *Australian Psychologist*, 30(1):71–74, 1995. doi: <https://doi.org/10.1111/j.1742-9544.1995.tb01750.x>. URL <https://aps.onlinelibrary.wiley.com/doi/abs/10.1111/j.1742-9544.1995.tb01750.x>.
- [32] Will Schutz. Beyond firo-b—three new theory-derived measures—element b: Behavior, element f: Feelings, element s: Self. *Psychological Reports*, 70(3):915–937, 1992. doi: 10.2466/pr0.1992.70.3.915. URL <https://doi.org/10.2466/pr0.1992.70.3.915>. PMID: 1620783.
- [33] Aitor Aritzeta, Stephen Swales, and Barbara Senior. Belbin’s team role model: Development, validity and applications for team building*. *Journal of Management Studies*, 44(1):96–118, 2007. doi: <https://doi.org/10.1111/j.1467-6486.2007.00666.x>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6486.2007.00666.x>.
- [34] Ioanna Lykourantzou, Angeliki Antoniou, Yannick Naudet, and Steven P. Dow. Personality matters: Balancing for personality types leads to better outcomes for crowd teams. In *Proceedings of the 19th ACM Conference on Computer-Supported*

- Cooperative Work amp; Social Computing*, CSCW '16, page 260–273, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450335928. doi: 10.1145/2818048.2819979. URL <https://doi.org/10.1145/2818048.2819979>.
- [35] URL <https://blog.motivemetrics.com/Psychological-Traits-vs-Personality-Type-Theory>.
- [36] T. G. Andrews. *Methods of psychology / T. G. Andrews, editor*. John Wiley N.Y, 1948.
- [37] Gregory J. Boyle and Edward Helmes. *Methods of personality assessment*, page 110–126. Cambridge Handbooks in Psychology. Cambridge University Press, 2009.
- [38] Beatrice Rammstedt and Oliver P. John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of Research in Personality*, 41(1):203–212, 2007. ISSN 0092-6566. doi: <https://doi.org/10.1016/j.jrp.2006.02.001>. URL <https://www.sciencedirect.com/science/article/pii/S0092656606000195>.
- [39] Paul T. Costa Jr. and Robert R. McCrae. Domains and facets: Hierarchical personality assessment using the revised neo personality inventory. *Journal of Personality Assessment*, 64(1):21–50, 1995. doi: 10.1207/s15327752jpa6401_2. URL https://doi.org/10.1207/s15327752jpa6401_2. PMID: 16367732.
- [40] O. P. John, E. M. Donahue, and R. L. Kentle. The big-five inventory-version 4a and 54. *Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research*, 1991.
- [41] Robert R. McCrae and Paul T. Costa. A contemplated revision of the neo five-factor inventory. *Personality and Individual Differences*, 36(3):587–596, 2004. ISSN 0191-8869. doi: [https://doi.org/10.1016/S0191-8869\(03\)00118-1](https://doi.org/10.1016/S0191-8869(03)00118-1). URL <https://www.sciencedirect.com/science/article/pii/S0191886903001181>.
- [42] Robert R. McCrae. *The Five-Factor Model of Personality: Consensus and Controversy*, chapter 09, page 129–141. Cambridge Handbooks in Psychology. Cambridge University Press, 2 edition, 09 2020. doi: 10.1017/9781108264822.013. URL <https://www.cambridge.org/core/books/abs/cambridge-handbook-of-personality-psychology/fivefactor-model-of-personality-consensus-and-controversy/B378236A6B16A7CBC1C8CD5CD12D01BF>.
- [43] Brent W. Roberts and Daniel Mroczek. Personality trait change in adulthood. *Current Directions in Psychological Science*, 17(1):31–35, 2008. doi: 10.1111/j.1467-8721.2008.00543.x. URL <https://doi.org/10.1111/j.1467-8721.2008.00543>. PMID: 19756219.

- [44] Meera Komarraju, Steven J. Karau, Ronald R. Schmeck, and Alen Avdic. The big five personality traits, learning styles, and academic achievement. *Personality and Individual Differences*, 51(4):472–477, 2011. ISSN 0191-8869. doi: <https://doi.org/10.1016/j.paid.2011.04.019>. URL <https://www.sciencedirect.com/science/article/pii/S0191886911002194>. Digit Ratio (2D:4D) and Individual Differences Research.
- [45] M. Jokela, C. Hakulinen, A. Singh-Manoux, and M. Kivimäki. Personality change associated with chronic diseases: pooled analysis of four prospective cohort studies. *Psychological Medicine*, 44(12):2629–2640, 09 2014. doi: 10.1017/S0033291714000257. URL <https://www.cambridge.org/core/journals/psychological-medicine/article/abs/personality-change-associated-with-chronic-diseases-pooled-analysis-of-four-prosp/9A390646DEE8FC331104AE889946F0D8>.
- [46] Christopher J. Soto, Anna Kronauer, and Josephine K. Liang. *Five-Factor Model of Personality*, pages 1–5. John Wiley Sons, Ltd, 2015. ISBN 9781118521373. doi: <https://doi.org/10.1002/9781118521373.wbeaa014>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118521373.wbeaa014>.
- [47] John M Digman. *The curious history of the five-factor model*, volume optional, chapter 1, pages 1–20. Guilford Press, optional 1996.
- [48] Hans Jurgen Eysenck. Dimensions of personality: 16, 5 or 3?—criteria for a taxonomic paradigm. *Personality and Individual Differences*, 12(8):773–790, 1991. ISSN 0191-8869. doi: [https://doi.org/10.1016/0191-8869\(91\)90144-Z](https://doi.org/10.1016/0191-8869(91)90144-Z). URL <https://www.sciencedirect.com/science/article/pii/019188699190144Z>.
- [49] Lewis R Goldberg. The structure of phenotypic personality traits. *American psychologist*, 48(1):26, 1993. doi: 10.1037/0003-066X.48.1.26. URL <https://psycnet.apa.org/doiLanding?doi=10.1037%2F0003-066X.48.1.26>.
- [50] Oliver P John, Laura P Naumann, and Christopher J Soto. *Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues*, volume 03, pages 114–158. The Guilford Press, 01 2008.
- [51] K. Anderson and P.W. McOwan. A real-time automated system for the recognition of human facial expressions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1):96–105, 02 2006. ISSN 1941-0492. doi: 10.1109/TSMCB.2005.854502. URL <https://ieeexplore.ieee.org/document/1580621>.

- [52] Maja Pantic and Ioannis Patras. Patras, i.: Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE transactions on systems, man, and cybernetics, part b* 36(2), 433-449. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 36:433–49, 05 2006. doi: 10.1109/TSMCB.2005.859075.
- [53] Peter A Gloor, Andrea Fronzetti Colladon, Erkin Altuntas, Cengiz Cetinkaya, Maximilian F Kaiser, Lukas Ripperger, and Tim Schaefer. Your face mirrors your deepest beliefs—predicting personality and morals through facial emotion recognition. *Future Internet*, 14(1):5, 12 2021. ISSN 1999-5903. doi: 10.3390/fi14010005. URL <https://www.mdpi.com/1999-5903/14/1/5>.
- [54] P. Ekman and W.V. Friesen. *Facial action coding system: A technique for the measurement of facial movement*. Consulting Psychologists Press, Palo Alto, CA, 1978.
- [55] Chien-Cheng Lee, Cheng-Yuan Shih, Wen-Ping Lai, and Po-Chiang Lin. An improved boosting algorithm and its application to facial emotion recognition. *Journal of Ambient Intelligence and Humanized Computing*, 3(1):11–17, Mar 2012. ISSN 1868-5145. doi: 10.1007/s12652-011-0085-8. URL <https://doi.org/10.1007/s12652-011-0085-8>.
- [56] Michael A. Sayette, Jeffrey F. Cohn, Joan M. Wertz, Michael A. Perrott, and Dominic J. Parrott. A psychometric evaluation of the facial action coding system for assessing spontaneous expression. *Journal of Nonverbal Behavior*, 25(3):167–185, Sep 2001. ISSN 1573-3653. doi: 10.1023/A:1010671109788. URL <https://doi.org/10.1023/A:1010671109788>.
- [57] Joan-Isaac Biel, Lucía Teijeiro-Mosquera, and Daniel Gatica-Perez. Facetube: Predicting personality from facial expressions of emotion in online conversational video. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction, ICMI '12*, page 53–56, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314671. doi: 10.1145/2388676.2388689. URL <https://doi.org/10.1145/2388676.2388689>.
- [58] Mihai Gavrilescu. Study on determining the big-five personality traits of an individual based on facial expressions. In *2015 E-Health and Bioengineering Conference (EHB)*, pages 1–6, Iasi, Romania, 09 2015. IEEE. ISBN 978-1-4673-7544-3. doi: 10.1109/EHB.2015.7391604. URL <https://ieeexplore.ieee.org/document/7391604>.
- [59] Mihai Gavrilescu. Proposed architecture of a fully integrated modular neural network-based automatic facial emotion recognition system based on facial action

- coding system. In *2014 10th International Conference on Communications (COMM)*, pages 1–6, Bucharest, Romania, 05 2014. IEEE. ISBN 978-1-4799-2385-4. doi: 10.1109/ICComm.2014.6866754. URL <https://ieeexplore.ieee.org/document/6866754>.
- [60] Marilyn Hill and Kenneth Craig. Detecting deception in pain expressions: The structure of genuine and deceptive facial displays. *Pain*, 98(1):135–144, 02 2002. doi: 10.1016/S0304-3959(02)00037-4. URL https://journals.lww.com/pain/Abstract/2002/07000/Detecting_deception_in_pain_expressions__the.15.aspx.
- [61] Gwen C. Littlewort, Marian Stewart Bartlett, and Kang Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27(12):1797–1803, 11 2009. ISSN 0262-8856. doi: <https://doi.org/10.1016/j.imavis.2008.12.010>. URL <https://www.sciencedirect.com/science/article/pii/S0262885609000055>. Visual and multimodal analysis of human spontaneous behaviour:.
- [62] Maria Koutsombogera and Carl Vogel. Modeling collaborative multimodal behavior in group dialogues: The multisimo corpus. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, H  ne Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, volume Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pages 2945 – 2951, Paris, France, may 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9. URL <https://aclanthology.org/L18-1466>.
- [63] 2022. URL <http://familyfeudfriends.arjdesigns.com/>.
- [64] Jin Hyun Cheong, Tiankang Xie, Sophie Byrne, and Luke J. Chang. Py-feat: Python facial expression analysis toolbox. *CoRR*, abs/2104.03509, 04 2021. doi: 10.48550/ARXIV.2104.03509. URL <https://arxiv.org/abs/2104.03509>.
- [65] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: An open source facial behavior analysis toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10, Lake Placid, NY, USA, 03 2016. IEEE. doi: 10.1109/WACV.2016.7477553. URL <https://ieeexplore.ieee.org/document/7477553>.
- [66] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv*, abs/1905.00641,

2019. doi: 10.48550/ARXIV.1905.00641. URL
<https://arxiv.org/abs/1905.00641>.
- [67] Li Zhang, Guan Gui, Abdul Mateen Khattak, Minjuan Wang, Wanlin Gao, and Jingdun Jia. Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. *IEEE Access*, 7:56028–56038, 02 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2899940. URL
<https://ieeexplore.ieee.org/document/8643367>.
- [68] Shifeng Zhang, Xiaobo Wang, Zhen Lei, and Stan Z. Li. Faceboxes: A cpu real-time and accurate unconstrained face detector. volume 364, pages 297–309, 2019. doi:
<https://doi.org/10.1016/j.neucom.2019.07.064>. URL
<https://www.sciencedirect.com/science/article/pii/S0925231219310719>.
- [69] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533, 06 2016. doi: 10.1109/CVPR.2016.596.
- [70] Zhiwen Shao, Zhilei Liu, Jianfei Cai, and Lizhuang Ma. Jaa-net: Joint facial action unit detection and face alignment via adaptive attention. *International Journal of Computer Vision*, 129(2):321–340, Feb 2021. ISSN 1573-1405. doi:
10.1007/s11263-020-01378-z. URL
<https://doi.org/10.1007/s11263-020-01378-z>.
- [71] Luan Pham, The Huynh Vu, and Tuan Anh Tran. Facial expression recognition using residual masking network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4513–4519, Milan, Italy, 01 2021. IEEE. doi:
10.1109/ICPR48806.2021.9411919. URL
<https://ieeexplore.ieee.org/document/9411919>.
- [72] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL
<https://doi.org/10.5281/zenodo.3509134>.
- [73] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.
- [74] URL <https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/ai-ml-android-neural-processing/data-collection-pre-processing>.
- [75] Emre Rençberoğlu. Fundamental techniques of feature engineering for machine learning | by emre rençberoğlu | towards data science, Apr 2019. URL

[https://towardsdatascience.com/
feature-engineering-for-machine-learning-3a5e293a5114#199b](https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114#199b).

- [76] Nov 2020. URL
<https://www.geeksforgeeks.org/feature-engineering-in-r-programming/>.
- [77] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50 – 60, 03 1947. doi: 10.1214/aoms/1177730491. URL
<https://doi.org/10.1214/aoms/1177730491>.
- [78] J. DeCoster, A.-M. R. Iselin, and M. Gallucci. A conceptual and empirical examination of justifications for dichotomization. *psychological methods*. *Psychological Methods*, 14 (4):349–366, optional 2009. doi: 10.1037/A0016956. URL
<https://doi.org/10.1037/A0016956>.
- [79] Bjarke Mønsted, Anders Mollgaard, and Joachim Mathiesen. Phone-based metric as a predictor for basic personality traits. *Journal of Research in Personality*, 74:16–22, 2018. ISSN 0092-6566. doi: <https://doi.org/10.1016/j.jrp.2017.12.004>. URL
<https://www.sciencedirect.com/science/article/pii/S0092656618300011>.
- [80] Shazia Afzal and Peter Robinson. Natural affect data — collection annotation in a learning context. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–7, Amsterdam, Netherlands, 09 2009. IEEE. ISBN 978-1-4244-4800-5. URL
<https://ieeexplore.ieee.org/document/5349537>.
- [81] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6, Ljubljana, Slovenia, 05 2015. IEEE. ISBN 978-1-4799-6026-2. doi: 10.1109/FG.2015.7284869. URL
<https://ieeexplore.ieee.org/document/7284869>.
- [82] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
- [83] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,

- M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [84] Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. *R News*, 2(3):18–22, 2002. URL <http://CRAN.R-project.org/doc/Rnews/>.
- [85] Simon Baron-Cohen, Sally Wheelwright, Jacqueline Hill, Yogini Raste, and Ian Plumb. The “reading the mind in the eyes” test revised version: A study with normal adults, and adults with asperger syndrome or high-functioning autism. *Journal of Child Psychology and Psychiatry*, 42(2):241–251, 2001. doi: <https://doi.org/10.1111/1469-7610.00715>. URL <https://acamh.onlinelibrary.wiley.com/doi/abs/10.1111/1469-7610.00715>.
- [86] Wu Youyou, Michal Kosinski, and David Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 01 2015. doi: 10.1073/pnas.1418680112. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1418680112>.