PREDICTING THE PREFERENCE OF LIKING USING FNIRS AND MACHINE LEARNING ALGORITHMS

MEHMET YİĞİT KÖKSAL

MEF UNIVERSITY

JANUARY 2023

MEF UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
MASTER'S IN INFORMATION TECHNOLOGIES

M.Sc. THESIS

PREDICTING THE PREFERENCE OF LIKING USING FNIRS AND MACHINE LEARNING ALGORITHMS

Mehmet Yiğit KÖKSAL
Orcid No: 0000-0002-6513-5163

Asst. Prof. Dr. Tuna ÇAKAR

JANUARY 2023

GCIS

ACADEMIC HONESTY PLEDGE

I declare that all the information in this study is collected and presented in accordance with academic rules and ethical principles, and that all information and documents that are not original in the study are referenced in accordance with the citation standards, within the framework required by the rules and principles.

Name and Surname: Mehmet Yiğit KÖKSAL

Signature:

ABSTRACT

PREDICTING THE PREFERENCE OF LIKING USING FNIRS AND MACHINE LEARNING ALGORITHMS

Mehmet Yiğit KÖKSAL

M.Sc/MA in Information Technologies

Thesis Advisor: Asst. Prof. Dr. Tuna ÇAKAR

January 2023, 94 Pages

The fMRI method, which is generally used to detect behavioral patterns, draws attention with its expensive and impractical features. On the other hand, the near-infrared spectroscopy (fNIRS) method is less expensive and portable, but it is as effective as fMRI in creating a good prediction model. With this method, a model has been developed that can predict whether a person likes a visual stimulus or not, using various classical machine learning algorithms including Support Vector Machines (SVM), Random Forests, XGBoost, LightGBM and K-Nearest Neighbors (KNN). With implementing tree-based and booster algorithms in addition to SVM and KNN which have been frequently used algorithms in this fNIRS domain, it was aimed to do a complementary comparison in addition to these acknowledged algorithms. Moreover, various missing value imputation methodologies were used to find the best suitable approach for this kind of classification problem. K-Means clustering, which is an unsupervised learning method, was also utilized to cluster similar fNIRS measurements of participants that may improve classification results by one-hot encoding those groups. Furthermore, certain feature extraction and wrapper methodologies were also applied for an attempt to enhance the performance of liking prediction models as a secondary goal. PCA, Isomap and t-SNE methodologies were implemented as feature extraction approaches, and forward selection wrapper design was utilized as an additional step to further development of the model by comparing their scores with each other. Cross-validation F1-scores of these models were used to find out the best model among them. Leave-one-group-out cross validation was exploited in comparison of the models. This meant that these cross-validation results corresponded to each of participants' data i.e. testing every participants' fNIRS

measurements alone in each fold. This way both every score specific to each participant could be seen and it ensured models' results were statistically reliable. Following
evaluations also included permutation and Wilcoxon Signed-Rank tests to compare each model's performance with each other by testing the statistical significance of those results.

Keywords: machine learning, decision-making, optical brain imaging, fnirs, feature extraction, feature selection.

Numeric Code of the Field: 92404

ÖZET

FNIRS VE MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE BEĞENİ TAHMİNİ

Mehmet Yiğit KÖKSAL

Bilişim Teknolojileri Yüksek Lisans Programı

Tez Danışmanı: Asst. Prof. Dr. Tuna ÇAKAR

Ocak 2023, 94 sayfa

Davranışsal örüntüleri tespit etmede genel olarak kullanılan fMRI yöntemi pahalı ve pratik olmayan özellikleriyle dikkat çekmektedir. Buna karşın yakın kızılötesi spektroskopi (fNIRS) yöntemi daha ucuz ve portatif özelliklere sahip olmak ile birlikte, iyi bir tahmin modeli oluşturmada fMRI kadar etkilidir. Bu yöntem ile çeşitli makine öğrenmesi algoritmaları kullanılarak insanların bir görsel uyaranı beğenip beğenmediğini tahmin edebilecek bir model geliştirilmiştir. Kullanılan klasik makine öğrenmesi metotları Destekleyici Vektör Makineleri (SVM), Rastgele Orman algoritması, XGBoost, LightGBM, k En Yakın Komşu (KNN) algoritmalarıdır. SVM ve KNN gibi fNIRS çalışmalarında sık kullanılan ve başarıları onaylanmış algoritmaların yanında, artırma ve ağaç bazlı algoritmalar da ek olarak kullanılarak tamamlayıcı bir karşılaştırma yapılması amaçlanmıştır. Bunun yanında, verideki eksik değerleri tamamlamak amacıyla çeşitli eksik veri doldurma yöntemleri kullanılmış ve bu tarz bir sınıflama problemi için aralarından en uygun olanı seçilmiştir. Model geliştirilirken ana odak olan öznitelik indirgeme yöntemleri arasında karşılaştırma yapılmıştır. Gözetimsiz bir eğitme yöntemi olan K-means kümeleme yaklaşımı kullanılarak benzer fNIRS ölçümlerine sahip olan katılımcılar kümelendikten sonra bu kümeler One-hot-encoding yöntemi ile kodlanarak sınıflama sonuçlarının daha başarılı çıkacağı düşünülmüştür. Bunun yanında, ikincil görev olarak, çeşitli öznitelik çıkarım ve sarıcı (öznitelik seçme) yöntemleri de uygulanarak beğeni tahmini modelleri performanslarının artırılması adına denemelerde bulunulmuştur. Kullanılan öznitelik çıkarım metotları arasında PCA, Isomap, t-SNE gibi yaklaşımlar yer almakla birlikte, sarıcı yöntem olarak ileri seçim sarıcı dizaynı ek bir adım olarak kullanılarak modellerin daha da geliştirilmesi amaçlanmıştır. Bu yöntemlerin sınıflama skorları kendi aralarında karşılaştırılarak sonuçlar gösterilmiştir. Modellerin çapraz doğrulama

yönteminden gelen F1 skorları kullanılarak en iyi modeller aranmıştır. Tek bir grubu dışarıda bırakan çapraz doğrulama yöntemi kullanılarak modeller arasında karşılaştırma yapılmıştır. Böylece bu çapraz doğrulama yöntemi kullanılarak her katta farklı bir katılımcının verisi tek başına test edilecek şekilde bir kurgu yapılmıştır. Bu şekilde hem her katılımcı özelinde skorlar görülmüş, hem de model performanslarından çıkan sonuçların istatiksel olarak daha güvenilir olması amaçlanmıştır. Son performans değerlendirme ve karşılaştırma yöntemleri olarak permütasyon ve Wilcoxon İşaretli Sıralama teknikleri kullanılarak modellerin skorları istatiksel olarak karşılaştırılmış ve istatiksel anlamları tespit edilmiştir.

Anahtar Kelimeler: Makine öğrenmesi, karar verme, optik beyin görüntüleme, fnirs, öznitelik çıkarma, öznitelik seçme.

Bilim Dalı Sayısal Kodu: 92404

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my thesis professor, Asst. Prof. Tuna Çakar, for their unwavering support and guidance throughout this project. His openness and willingness to assist me anytime, along with their expertise and patience in answering every one of my questions, made it possible for me to successfully complete this thesis. His teaching and motivation played a crucial role in my development as a student, and I am deeply grateful for everything they have done for me.

I would also like to express my sincere appreciation to my fellow colleague, Esin Tuna for their invaluable contribution to the experimental aspect of this project.

I am forever grateful to my father, Bülent Fahri Köksal, who unfortunately passed away during my pre-teens. His hard work and dedication to providing a bright future for me, my mother, and my brother will always be remembered and deeply appreciated.

I would also like to express my sincere appreciation to my cousin, Gözde Sebzeci, who has always been there for me. Her support and understanding have meant the world to me, and I am grateful for her presence in my life.

I am also thankful to my brother, Burak Yağız Köksal, for his eternal companionship and support throughout my life and for the special bond we share. His laughter and presence have been a constant source of joy and strength. I am grateful to have such a loving and supportive brother. Thank you for always being there for me.

I am forever grateful to my mother, Dilek Köksal for her love, support, and guidance throughout my life. She truly deserves everything that the world has to offer, and more. She has always been there for me, even in my failures or when I made mistakes, and her unwavering belief in me has been a constant source of strength. As a single mother, she has had to balance the roles of both father and mother for me and my brother. Her strength and determination in the face of adversity have been an inspiration to me, and I am eternally grateful for everything that she has done for our family.

Finally, I would like to express my heartfelt appreciation to my wife, Cemran Toydemir Köksal, for her unwavering love and support. Her encouragement and belief in me, even in the face of challenges and difficult decisions, have been a constant source of motivation and strength. Even in my moments of failure or when I made mistakes, she was always there for me and her belief in me never wavered, providing me with a constant sense of strength. I am forever grateful for her presence in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AUC : Area Under the ROC Curve

DLPFC : Dorsolateral Prefrontal Cortex

EEG : Electroencephalogram

FAA : Frontal Alpha Asymmetry

FMRI : Functional Magnetic Resonance Imaging

FNIRS : Functional Near-Infrared Spectroscopy

FP : False Positive

FN : False Negative

FPR : False Positive Rate

GLM : Generalized Linear Model

GMM : Gaussian Mixture Models

GSR : Galvanic Skin Response

HRV : Heart Rate Variability

IQR : Interquartile Range

KNN : K-Nearest Neighbors

LASSO : Least Absolute Shrinkage and Selection Operator

LGBM : Light Gradient-Boosting Machine

LOGOCV : Leave-One-Group-Out Cross Validation

MAR : Missing at Random

MCAR : Missing Completely at Random

MDS : Multidimensional Scaling

MEG : Magnetoencephalography

MICE : Multiple Imputation by Chained Equations

MNAR : Missing Not at Random

OFC : Orbitofrontal Cortex

PCA : Principal Component Analysis

PFC : Prefrontal Cortex

ROC : Receiver Operating Characteristic

SST : Steady-State Topography

SVM : Support Vector Machines

TN : True Negative

TP : True Positive

TPR : True Positive Rate

T-SNE : T-Distributed Stochastic Neighbor Embedding

VIF : Variance Inflation Factor

VMPFC : Ventromedial Prefrontal Cortex

XGB : Extreme Gradient-Boosting Algorithm

INTRODUCTION

1. Purpose of Thesis

Existing studies in neuroscience mainly use the fMRI method. However, it is an expensive and impractical method for decision making studies, despite its more advanced spatial resolution. Due to these aspects, this study makes use of the functional near-infrared spectroscopy (fNIRS), which is a low-cost, non-invasive and portable optical brain imaging methodology [1], [2]. The main aim of this study is to come up with a model to predict whether an image is liked/preferred aesthetically or not based on fNIRS measurements computed on various machine learning algorithms and finally compare the performances of them. Therefore, the main goal is to develop a binary classification model that best predicts each class. As a secondary goal, various feature extraction methodologies will be investigated to further improve the predictive power of the final model. Finally, a wrapper approach will be implemented to see if it could make significant improvements to the model.

2. Related Work

There are few studies using fNIRS in neuroscience context, including purchase behavior [1], identification of activation patterns in the prefrontal cortex during product selection [3], and fewer using it an artificial intelligence context, such as investigation of deep learning for fNIRS based brain computer interface [4], neuromarketing study using machine learning for predicting purchase decision [5].

Brain imaging technologies have been used to better understand which state of mind that an individual is in. Ramirez et al. [6] used various methods including these brain imaging techniques such as electroencephalogram (EEG), and functional near-infrared spectroscopy (FNIRS), with the help of other methodologies such as galvanic skin response (GSR) and heart rate variability (HRV) to see the effect of different colors of a product on consumer preferences. They acquired EEG and fNIRS signals and processed them to obtain AW index and HbO, HbR concentrations. AW index is calculated by taking the difference of alpha powers of two electrodes that correspond to the left and right side of the frontal lobe. This calculation is based on frontal alpha asymmetry theory, which states that the frontal lobes of left and right hemispheres

signify positive and negative emotions, respectively. Thus, it can be said that the negative AW index indicates negative emotions whereas a positive AW index is a sign of positive feelings. Later, these EEG and fNIRS data were combined with other GSR and HRV sensor data into the final feature set, to be used in machine learning algorithms. They stated that classification of EEG and FNIRS signals is difficult due to their nonlinear characteristics, but they used Support Vector Machine (SVM) classifier's kernel functions to overcome this problem. They found that the Fine Gaussian kernel obtained the best results, among others. Furthermore, Subspace k-Nearest Neighbors (KNN) classifier is used to compare the results of SVM, and when their average results among 4 subjects were compared, SVM's is higher than KNN's, 88.2% and 79.6% respectively. However, they added that due to the small number of subjects (4) and low number of trials, the results are limited, thus further testing should be made.

Another study also used EEG signals to find out which frequencies and channels could be better indicators of liking preferences of individuals by making them choose which shoe they like [7]. They stated although fMRI technique is used in neuromarketing studies because of its superior spatial resolution, its lack of speed to catch the blood flow to the activated locations and relevant stimulus, and being an expensive method makes it unpreferable. Thus, they suggested using EEG owing to its relatively cheapness, and its ability to capture high temporal resolution with inferior spatial resolution. To achieve their goal, they used logistic regression in their model. In their implementation of that model, they chose a generalized linear model (GLM) as they stated that logistic regression is a special form of it. Their findings suggest that by reducing the number of channels and focusing certain frequency values to predict liking preference, the prediction process could be cheaper and less time consuming. In addition, due to the high number of predictors regression models became ill-posed, and data being imbalanced where there were 65 like and 144 dislike cases, they divided data into low and high frequencies.

Cakir et al. [1] investigated credibility of fNIRS methodology on neuromarketing studies by developing a neurophysiologically-informed model on purchasing decision of individuals. This study claims that it is original in three ways, one of which is using a newly emerged methodology, fNIRS to decode purchasing

decisions. Secondly, rather than measuring hypothetical preferences or choices, subjects were asked to make a decision to purchase products in a realistic environment with a predetermined budget. Lastly, it is suggested that the accuracy of the model increased when subjects' sensitivity to the budget value was included as an additional feature. Products included in the realistic purchasing test consisted of food, cleaning and personal care groups, having distinct 39, 17, 22 numbers of them respectively. They were shown one by one for each product group to the subjects where they were informed to make buy or pass decisions. Participants were forced to decide by telling them if they do not spend more than 40 Turkish liras, they would only receive half of the unspent money. During these decision-making processes, fNIRS continuously measured oxygenation of the frontal lobe and results of these oxygenation signals were averaged for each individual when they were being extracted, acquiring average signals for buy/pass decisions. Discriminant analysis was used to build a classifier for buy or pass decisions from 16 optodes that measures oxygenation signals, this led to investigation of these optodes for their separate role in purchasing decisions. Their findings suggest that fNIRS can provide practical biomarkers by improving classification accuracy of purchasing behaviors up to 85%. Thus, it can be used as a main or complementary method with traditional neuromarketing research methods. In addition, 85% accuracy is obtained by separating participants into two groups depending on their sensitivity to budget. It is stated that this may indicate a difference in goal structures generated by participants that led to this separation.

Most relevant one to our study is Hosseini et al. [8]'s, where they evaluate whether one likes or dislikes a stimulus according to its fNIRS measurements. This evaluation was made using machine learning, with linear Support Vector Machine (SVM) algorithm. Various objects consisting of sceneries, foods, cars, and animals were shown to subjects as stimulus to indicate their liking. Same as Yılmaz et al.[7]'s suggestion for using EEG and Cakir et al. [1]'s suggestion of fNIRS, this study also implies that although fMRI has high spatial resolution, fNIRS method offers relatively good results while having low-cost and portable properties compared to fMRI. After preprocessing of the fNIRS measurements, due to high dimensionality, having 114 attributes, they used Principal Component Analysis (PCA) to map the data to a lower dimensional space while keeping 99% variance of the data. While applying PCA, they utilized wrappers to find the best combination of principal components that best

explains the attractive/neutral/unattractive classes of preference [9]. In addition, they indicate that SVM works best in low sample data compared to other linear classifiers because it tries to maximize the distance between data points that are closest to the boundaries. Since, their data has a small number of samples where five subjects had participated in this study, they chose linear SVM to predict preference of participants. Validation of the classifier was tested by stratified 10-fold cross validation and this process was applied 10 times as a single cross validation might not be enough as a reliable method. Then, the average of them was taken and final accuracy was acquired. Their findings showed that for all five participants, accuracy of classification was 72.9% for attractive stimuli, and 68.3% for unattractive stimuli. In addition, signal change in the channels related to medial orbitofrontal regions responded more to positive stimuli. In other words, decoding attractiveness is more promising than decoding unattractiveness, as their study suggested.

3. Outline/Overview

This paper is organized as follows. In chapter 1, theoretical background of fNIRS, decision-making, liking preference and emotions, brain regions and machine learning are given. Chapter 2 gives information about experimental design, and methodologies used in data preparation, modeling, and evaluation steps in detail. In Chapter 3, results of the methodologies and models are given and discussion of them are made. Finally, conclusion and future work is given.

1. THEORETICAL BACKGROUND

1.1. Individual decision-making

Decision-making has been a vital part of human life since the beginning of its existence. In particular, the primary characteristic of a human being involves decision making, since it enables humans to be active by forcing it to make choices. Nearly in every moment of an individual's life, decision making takes place such as preferring certain food for another or deciding where to go for a vacation. Even the most basic things like moving one's arm to reach a glass of water to quench his thirst involves decision making. Thus, the absence of this neurological phenomenon could affect the human's functioning, and even so to speak one could say that without it there would not be any difference between a human and an inorganic matter. This neurological phenomenon has been serving a great purpose to humans in various aspects throughout history of its existence, from protecting itself against wilderness, choosing one thing over another to have a greater benefit, to developing strategies to adapt to certain conditions or to its surroundings, with the help of the emotions which is inextricably intertwined with the decision-making phenomenon, due to the evolutionary process of the human brain. Thus, with these neurological events, humans perceive the world according to the certain cues surrounding them.

Decision theory tries to explain decision-making of individuals by concerning the underlying reasons of a person's choices. Daily or mundane decisions generally are made with heuristics, which is a biased state of mind that is formed from an individual's beliefs, values, or desires [10], [11]. There are two interrelated facets of decision theory i.e., normative, descriptive theories. Normative theory concerns finding prescriptions as what might an individual do to satisfy its rational needs. Generally, those prescriptions conform to the decision maker's beliefs and values [12]. Expected utility theory which is an orthodox normative theory states that people tend to prefer the greatest desirable outcome in the situation of uncertainty [10]. On the other hand, descriptive theory seeks how an individual makes the decision, hence describing beliefs and values of the decision maker that ensures that decision to be made.

1.2. Liking Preference and Emotions

The preference on liking or appraisal of a stimulus has been a fundamental subject for psychological studies of human life from classical theories through the transition to experimental psychology and to even present day, eliciting the question how we conclude that something is likeable or not. Besides, there is an ambiguity how this preference of liking is processed as an information and then becomes a meaning to apprehend [13]. Most human decisions are never discrete according to both classic and contemporary knowledge. It can be seen in human actions or decisions that follow one another by affecting the next one. There are two main approaches which explains the assessment of thoughts: (1) aesthetic evaluation of an object's appearance or goodness [14], [15], (2) and cognitive evaluation of understanding, meaning or informational subject [13], [16]. These modes are argued to be core instruments for our interaction with the environment [17], and it is still unknown if these are functionally or behaviorally distinct [13].

Two main arguments have been presented to explain the relationship between liking and understanding, one of which is a classical approach as these are discrete phenomena, raised in context of aesthetics or taste by the likes of Plato, Aristotle and Kant [13], [17], [18]. This approach originates from the thought that evaluation of understanding and liking depends on different stimulus aspects. In brief, the former one pertains to one's socio-cultural position and subjective process of information that is viewed as distinct from hedonic(liking) aspect, the latter one is thought to be more non-cognitive assessment of an object's appeal. These two are thought to be interchangeable when making an assessment [13], [19].

Second argument pertains to easiness of perception for humans to better understand and perceive its surroundings, with the content of perceptual fluency, a subjective experience which facilitates the processing of information. While this heuristic eases the recognition of physical identity of stimulus, it is accompanied by various factors such as repetition, perceptual priming, clarification, display time, or figure-to-ground contrast [20], [21]. In addition to perceptual fluency, a second factor complements how the human brain handles information, conceptual fluency, which avails facilitating mental operations regarding stimulus meaning and its relation to semantic knowledge structures. These two forms the term processing fluency, which

can hedonically create a positive experience. It could be because if the fluency amplifies, the recognition of the object gets easier, thus errors involving the identification of the stimulus reduces. It may also have a beneficial effect on cognitive systems because it cues the stimulus as a familiar object, meaning it cannot cause harm [22]. Thus, a high processing fluency may make a stimulus more likeable and beautiful [23].

Apart from the aesthetical aspect of decision-making, another interdisciplinary field has emerged to better understand how neurobiological factors influence economic decisions [24]. It elicits what variables, which are computed by the brain, affect human decisions in different cases, and how those computations are involved in neurobiological processes to build feasible models for decision-making. In this field, various studies have been made covering hedonic food consumption such as choosing one food over another and purchasing behavior according to the price and content of the product [1]. Because of this field and its studies related to not just cognitive, but also emotional processes, involving reward evaluation, value comparison, ambiguity/risk management, which are all means to make a decision, they could be linked to develop a tendency of liking, or aesthetical appreciation of a stimulus.

1.3. Brain Regions

Some parts of the human brain are related to decision-making, primarily prefrontal cortex (PFC), which facilitates evaluation of perceptual content and reward value of a task [25], [26]. Especially, medial prefrontal cortex and amygdala are predominantly connected with each other to supervise representation of emotions [27]. Ventromedial prefrontal cortex (vmPFC) conducts selection of actions regarding to their reward values while it is regulated by dorsolateral prefrontal cortex (dlPFC) to control emotional values. [28], [29]. DlPFC also is an integral part of regulating volitional actions [30], and perceptual decision-making [31], critically involved in emotional processes, but the most important roles of it are the evaluation of an object's visual aesthetic and suppressing negative emotional states. In addition, aesthetically satisfying images cause a rise in activity in the left dlPFC, mPFC and OFC regions [32]. In addition, Frontal Alpha Asymmetry (FAA) , which explains the difference between the right and left alpha oscillations over the prefrontal channels of the brain,

shows that the left frontal cortex activates more during positive emotional situations [33]. This might make the FAA a strong indicator for liking preference of individuals.

1.4. Functional near infrared spectroscopy (fNIRS)

In the neuromarketing field, there are various methods that gather data from the brain. Some of them collect data directly from the brain that are called neurophysiological methods while biometric methods measure brain activity indirectly through other organs activity. Examples for neurophysiological methods could be given as functional magnetic resonance imaging (fMRI), electroencephalography (EEG), functional near-infrared spectroscopy (fNIRS), magnetoencephalography (MEG), and steady state topography (SST) whereas for biometric methods; galvanic skin response (GSR), eye-tracker, heart rate, and respiratory rate could be given. Some studies use a couple of these methods together to complement each other to come up with a better analysis [33]–[35].

fNIRS is a low-cost and practical method compared to other brain imaging techniques, especially compared to fMRI due to being a high-cost technology and having a fixed setup rather than a portable one such as fNIRS has. EEG is also mobile, but it is more sensitive to movements than fNIRS. In fact, EEG and fNIRS are mostly used together to complement each other in brain studies. fNIRS measures brain activity by assessing hemodynamic activity using near infrared lights. Neural activity is measured with delay due to fNIRS being extremely limited to taking signals below cortical surface. Its spatial resolution is in millimeters and comparable to fMRI's resolution, making it a finer choice in terms of cost effectiveness and portability and lesser sensitivity to movements [36].

1.5. Data Mining

Data mining involves processes of extracting and gathering data, data preparing and preprocessing to come up with a feasible solution for a problem using a large data set. It contains statistics and machine learning in a way, in creating models that find discoverable patterns. Several steps of the data mining process will be given in the next sections.

1.5.1. Data preparation and data preprocessing

This step involves the preparation and processing of the data to get it to be implemented on machine learning models. It can contain several sub-steps depending on the problem at hand.

1.5.1.1. Collecting dataset

The very first step of a machine learning problem is to gather relevant dataset. While gathering it, it is essential to collect the most informative features. It can be done by using an expert's knowledge in that domain. If an expert is not available, then the only option is to use brute-force, using every feature available. The downside of this method is that it comes with noise and missing values, which requires notable data cleaning and preprocessing [37].

In most cases, the dataset at hand is full of noises and errors. Real-world data does not come perfect, so it needs to be corrected. A hierarchy of problems has been proposed to be dealt with to make the dataset ready to be used in algorithms. First thing to look at is the presence of impossible values inputted in features [37]. For instance, if the relevant feature is expected to have binary values, but one instance of it has a discrete value, then we identify it as an impossible value. They can be solved ideally while inputting phase of the data, so that they can be corrected. However, if it is impossible to enter the correct values, they can be simply treated as a missing value category to be removed from the dataset [37].

Next problem to be looked at is that no values have been inputted in an instance of a feature [37]. There are several methods that solve this issue which most of them will be mentioned in upcoming paragraphs. Finally, some features that are irrelevant are present in the dataset [37]. They are simply ignored and left out of the dataset.

1.5.1.2. Data Cleaning

Data cleaning is one of the most important steps in data mining. Detecting and repairing dirty data is the crucial part of data preprocessing. If dirty data left unrepaired, it would lead to inaccurate analytics and unreliable models. Data cleaning usually consists of two phases: error detection and error repairing. For detection of the

errors, quantitative and qualitative approaches exist in defining those errors. While quantitative techniques employ statistical methods to be used in outlier detection, qualitative techniques implement rules, constraints, and patterns to handle errors in the data [38].

1.5.1.3. Outlier Detection

The next thing to look at is the values that seem not likely to be true [37]. Outlier values could be put into this category, where it deviates significantly from other values of the sample in which it occurs [39]. Another way to describe an outlier is that it is an observation which appears to be inconsistent with the remainder of that dataset [39]. One way to look at it is by variable-by-variable data cleansing. These outliers can be spotted by their suspicious presence in the relevant probability distribution that they belong to. In other words, in a normal distribution they are further away than one standard deviation from the mean, or more depending on the domain and also the distribution at hand. Most of the time, these values are ignored and left out of the dataset due to the fact that they could be caused by mechanical faults, changes in system behavior, fraudulent behavior, human error, instrument error or natural deviations in populations [39]. However, for the last cause stated, they could be likely values and not wrong which lie at the tails of a distribution where it is more dispersed than it was thought before, thus having a possible variability [37]. In other cases, they could be veridical data that belongs to a cluster/label but situated inside another cluster's area [39]. Nevertheless, in most cases, they also left out to be able to come up with an accurate model that appeals to the general population.

Outlier detection is essential in certain fields. One of them concerns safety critical environments, where an outlier causes abnormal running conditions such as an aircraft engine rotation defect or a defect caused in a nuclear power plant which may have detrimental effects on the environment. Another one may detect an intruder in a system which should be addressed quickly. An outlier may be spotted in a factory production line by continuously comparing properties of a normal product with the newly produced ones to detect faults, thus reducing error costs. Spotting an outlier in a fraudulent act might be monitoring the usage of a customer's credit card in order to detect a swift change of usage pattern which may indicate that the card is stolen. This

outlier detection is done by analyzing and comparing time series of usage statistics [39].

Outlier detection is also utilized in loan application processing, where fraudulent applications or potentially troublesome customers can be detected. This way, a bank can spot a problematic customer early, and act accordingly to prevent further loans to be given or cancel the customer's current limit to prohibit further usage of the credit [39].

Networks can also have outlier detection systems, which can observe performance of them to catch any network bottleneck. This ensures the system to run properly by taking some preventive actions, averting critical failure of the network, and also avoiding user dissatisfaction [39].

Detection of novelties in images for surveillance systems may also be considered outlier detection, where maleficent acts can be caught before it becomes critical [39].

Maybe one of the most important uses is in the medical field [39]. Recently, the usage of smart watches has increased a lot, where it can monitor the heart rate of its user. Thus, an outlier detection system may detect an anomaly in heart rate that may warn its user to take it easy or see a doctor to check its condition. This way that system may save many lives.

Last but not least, mislabeled data may be detected in a training dataset, which may improve performance of a model [39].

Handling of these outliers differs depending on the application areas that have been mentioned. If an outlier is caused by an instrument reading error, it could simply be erased. If a survey on a certain population’s demographics is made, it could be seen that it may have outliers of very tall people. Thus, it is a natural occurrence and may not be excluded depending on the study that is going to be made. In fraudulent systems, like surveillance cameras, outliers are used to give an alarm. If that alarm has been correctly raised, that outlier data can be stored elsewhere to enhance the detection system’s performance [39].

When detecting an outlier, there are three fundamental approaches that try to come up with a solution to that detection problem. First type is to detect outliers without a knowledge of the data present. This type is especially related to unsupervised clustering where the data at hand is unlabeled, so there is no prior knowledge of the data. There is a requirement of availability of all of the data before processing, and also data has to be static. Data is processed as a static distribution, spotting the most isolated values, and finally marking them as outliers. This method assumes that the errors are located outside of the cluster, meaning they should appear as an outlier. An example is given in figure 1.1, where points V, W, X, Y and Z are located outside of the centered cluster that they could be potential outliers. The approach is generally backward-looking and mostly related to a batch processing system, where the input data is prepared before processing. To compare new values with the existing ones, it requires to accumulate sufficiently large data with good coverage. This approach can be divided into two techniques, diagnosis, and accommodation. Diagnosis methodology underlines potential outliers, so that the system may remove them from future distribution. This can be an iterative approach where outliers are removed continuously by fitting the model to the remaining data until no outliers are left. In contrast, the accommodation approach integrates outliers into the distribution to come up with a robust classification model. This robustness can resist any outlier in the data where even with them it represents a normal behavior. However, this approach comes with a price, it is computationally more costly than non robust methods. If the data is thought to have few outliers, then it may not be feasible to use this costly method, instead a non robust method is more adequate [39].

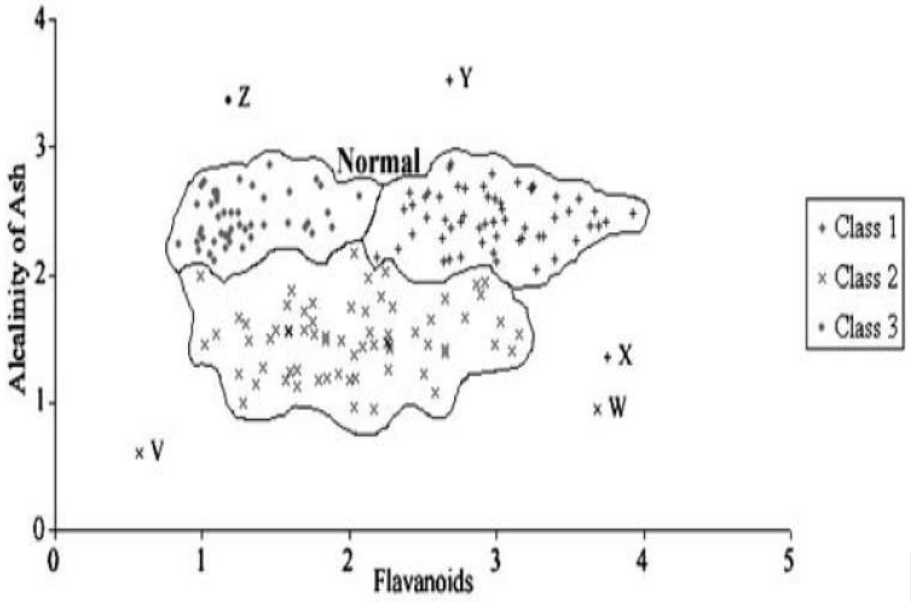


Figure 1.1: Data distribution classified by type 1 outlier classifying. Data comes from Wine dataset [40].

Second type of approach to outlier detection is to model both normality and abnormality that can be compared to supervised classification. This method requires pre-labelled data that is marked as normal or abnormal. In figure 1.2, three classes are specified by pre-labeling with outliers outside of the normal data that is in the center. Depending on the problem, either centered data points could be classified as a single classification as representing normality and the outlying points as abnormal values, or centered ones split into three classes as normal, and outliers are abnormal. Classification methods best fits into stationary data which has a static distribution as the classification needs to be refitted according to the new distribution. However, it should be mentioned that in evolutionary neural networks, the model does not need to be rebuilt as it is an incremental classifier. Second type can be used in online classification, where a model is formed from existing data and then upcoming data is classified as normal or abnormal (outlier) according to the learned model. It goes without saying that the data at hand should cover all variability and distribution of the true population to classify these outliers correctly [39].

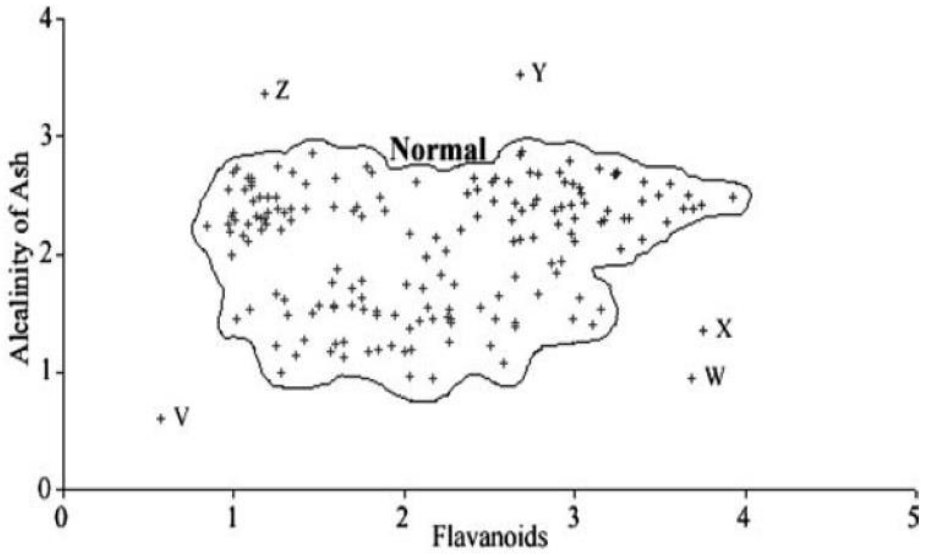


Figure 1.2: Data distribution classified by type 1 outlier classifying. Data comes from Wine dataset [40].

Lastly, the third type focuses only on modeling normality. However, on a few occasions, abnormality could be modeled. It is related to semi-supervised detection or recognition where the normal class is taught whereas the model learns to recognize abnormality. In other words, this approach only requires normal and pre-labelled data with the aim of inducing a boundary of normality. It can learn the model incrementally with each new data, thus tuning the model accordingly. Therefore, to reach this aim, a full spectrum of normal data should be available to come up with a generalization. Despite that, abnormal data is not needed for training unlike type 2. This brings some benefits in certain cases, such as fault detection areas with an example of outlier detection in an aircraft engine where it would be too costly to acquire abnormal data by simply damaging the engine to train the model. Another case would be fraud detection systems in which a new type of fraud could be encountered, making the model not handling it correctly. However, by only modeling normality, the model can detect that new fraud unless it lies in normality [39].

1.5.1.3.1. Methods of Outlier Removal

Before mentioning the methods that manage outliers, the simplest method is to visualize a feature to see the bad values that occur in a regular pattern, which is also a powerful and effective tool when handling outliers. There are two fundamental considerations when selecting a suitable method for outlier detection. First one is to

find an algorithm that can accurately model the distribution of the data and pinpoint outlier regions for a clustering, classification, or recognition type of model. Second one is choosing an appropriate neighborhood of interest. Whether user-defined or autonomously induced, selection of neighborhoods should be applicable for all density of distribution that could be encountered [39].

Most of the techniques used for detection of outliers come from the same fundamentals but have different names. Some of these names are, outlier detection, novelty detection, anomaly detection, noise detection, deviation detection or exception mining [39].

It is observed that outlier detection methods come from three fields which are statistics, neural networks, and machine learning. Some algorithms may have chosen more than one of these fields to come up with a better model or only one [39].

1.5.1.3.2. Statistical Methods

Statistical methods were the earliest approaches to outlier detection problems. Earliest ones are only applicable to single dimensional data sets, some techniques being univariate at best. One example for a single dimensional technique is Grubbs' method. It calculates a Z value by taking the difference between the mean value of attribute and the query value and dividing it to the standard deviation of all these values. This method requires no user parameters as all parameters are derived from the data. In addition, it relies on the number of values as higher numbers makes the model more statistically representative [39].

Statistical models have limited applicability as they are more suitable to quantitative real-valued and quantitative ordinal data sets. This suitability also increases processing time if complex data transformations are necessary [39].

Informal box plots can be applied to both univariate and multivariate data sets to detect outliers. This method allows its user to pinpoint outliers by just looking at the box plot. It can handle real-valued, ordinal and categorical attributes [39]. Box plots display five-number summaries which are the lower extreme, lower quartile, median, upper quartile and upper extreme points [41]. They are suitable to both symmetric and skewed distributions, but they can also identify infrequent values in categorical data

sets. By just visual detection, the points beyond the lower and upper extremes can be determined as outliers. But a better technique would be to pick upper and lower thresholds as being 1.5 x interquartile range (IQR) away from the upper and lower quartiles, beyond where the lower and upper outliers lie. In univariate outlier removal, outlier identification is made for each variable. If there are too many univariate outliers that correspond to a large portion of the data, they can be ranked by the frequency of these univariate outliers and trim the most frequent ones [42]. In Laurikkale et al.'s [42] study, they removed 10% of the worst examples within each class.

In univariate outlier detection identification, it is based on ordering of the data, mostly in ascending order and the five-number summary can be found according to that ordering. However, in multivariate data, there is no unambiguous total ordering [42]. To solve this problem, a reduced sub-ordering method is suggested [43]. It is done by transforming each multivariate observation into a set of scalars by using a distance metric. Mahalanobis distance is probably the most suitable candidate to be utilized in multivariate cases because it incorporates the dependencies between attributes which accomplishes the goal of multivariate outlier identification being detection of unusual value combinations. It is stated that many distance metrics, including Euclidean distance, are not suitable for this kind of data sets as they only include location information [42].

1.5.1.3.3. Machine learning outlier removal methods

Machine learning approaches can detect outliers in categorical data compared to most statistical approaches that do not have a mechanism to detect that data type. Some studies use the C4.5 decision tree to detect outliers for categorical data [44], [45]. For decision trees, prior knowledge is not required to detect outliers which could make the detection a bit faster than statistical methods that need information about the distribution and parameters of the data. Moreover, rule-based systems are also exploited for outlier detection, which are more flexible and incremental than decision tree techniques as rules can be changed or replaced in detection of outliers [39].

1.5.1.4. Methods to handle incomplete data

In most real-world data sets, incomplete or missing values are confronted which need to be dealt with. These missing values might exist due to several factors such as they are missing due to being forgotten or lost, it is not applicable to give a value for that instance, and lastly the designer of the data set might not care to give a value for that observation in the data set [37].

For the not applicable and lost values, randomness of the lost values should be looked at to come up with a proper technique in dealing with those missing values. When looking at randomness, three types come forward, missing completely at random (MCAR), missing not at random (MNAR), and missing at random (MAR). MCAR usually happens by accident such as accidentally losing the measurements taken from a subject. Hence it means that probability of missingness is not related to any other feature that belongs to the subject. This type of missing values can be handled properly with various methods contrary to MNAR data where missing values are related to unobserved information that belong to other characteristics of the subject. For instance, errors in the setup of an experiment could cause missing values to appear in a data set that cannot be resolved with a universal method [46]. MAR happens when the missing values are conditionally dependent on the outcome or other predictor variables. For example, if one feature has missing values in case of a device malfunction while not in properly functioning instances, it is an example for MAR.

When dealing with missing values, there are various approaches that can resolve incomplete data problems. For the MNAR cases, a solution could be to resampling the feature or repeating the experiment with proper ways. On the other hand, for MCAR and MAR situations, features having missing values can be imputed depending on other features or within itself from its own characteristics. Usually single and multiple imputation methods come into play to deal with these types of missing values. The main reason for imputing data rather than deleting them is to reduce the biases of the data set. It goes without saying that deleting the incomplete cases could be the most proper way in some situations, especially for MNAR data [47].

There are numerous single imputation methods some of which are, mean imputation, imputation with distributions, regression imputation and k-nearest

neighbor (KNN) imputation. Mean imputation involves imputing missing values with either mean, median or mode depending on the distribution of the data. This method has some drawbacks, one of which is that if there are many missing values, it can change the distribution of the data and make it biased. For imputation with distributions, missing value are imputed according to that features' distribution without changing its shape [47].

Regression techniques apply imputation with exploiting other variables in the data set, making it a more sophisticated technique compared to previous methods. Linear relationship should be present between the imputed feature and other features to have an unbiased imputation, especially in the presence of MAR and MNAR data [47], [48].

Lastly for single imputation, KNN technique employs an evaluation of the distance from k number of neighbors to impute similar values. Higher k value causes this method to be computationally expensive as more computations will be made.

Multiple imputation is the process of averaging the results across multiple imputed data sets. This resolves the possible biases that may be caused by single imputation methods in the presence of uncertainty of imputed values. Therefore, it tries to eliminate that uncertainty by implementing numerous feasible imputations, and in a way reducing those single imputation error's by averaging them. Three steps exist in this approach which are, (1) imputing data set n times resulting in n data sets at hand, (2) analyzing those data sets, (3) consolidation of all of these n data sets [47]. Most notable technique for multiple imputation is multiple imputation by chained equations (MICE), which exploits the correlation neighborhood between imputed features and others to come up with a proper handling of missing values.

1.5.2. Applications Of Machine Learning

There are many applications of machine learning, most important of which is predictive data mining [37].

A dataset is represented by a set of features which is used by machine learning algorithms to make predictions. These features may be of type continuous, categorical,

or binary depending on the given problem and data. There are two approaches as learning methods in general, supervised, and unsupervised learning.

1.5.2.1. Supervised learning

If labels of any given instance of a dataset is present, then we call it supervised learning. In supervised learning, the goal is to label unseen data that has just been encountered. This is done by using labeled data by putting them into training to find out the description of classes, which in turn is used to label newly encountered data [37].

Supervised learning can be divided into two subcategories, which are classification and regression problems. Classification involves training a prelabeled set and trying to predict those labels in the unseen data. Popular algorithms for classification problems are KNN, decision trees, random forests, and support vector machines. On the other hand, regression is used to understand the relationship between dependent and independent variables. Linear regression, logistic regression and polynomial regression are the most basic regression algorithms.

There are various evaluation metrics to compare the performances of supervised learning approaches, some of which are accuracy, f1-score, Receiver Operating Characteristic (ROC), and confusion matrix.

The accuracy of a model is defined in equation below,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1.1)$$

where positive being 1 (liking) and negative being 0 (disliking),

TP is True Positive e.g., where the target value is 1 and it is predicted as 1,

FP is False Positive e.g., where the target value is 0 and it is predicted as 1,

TN is True Negative e.g., where the target value is 0 and it is predicted as 0,

FN is False Negative e.g., where the target value is 1 and it is predicted as 0.

In addition to the average F1-score, Receiver Operating Characteristic (ROC) curve and area under that curve (AUC) were also plotted for all models. In figure 1.3, a representation of ROC curve is given,

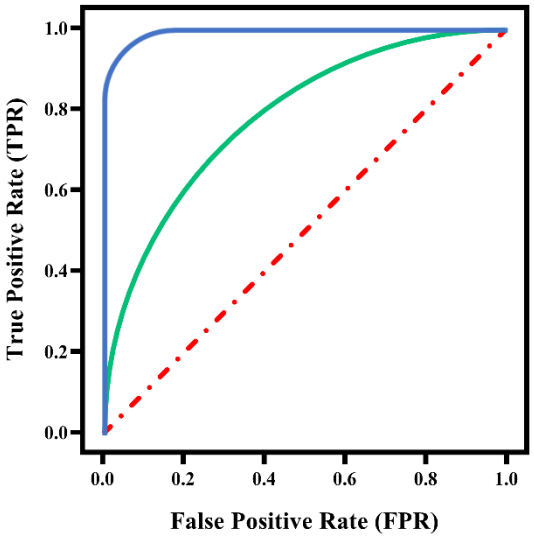


Figure 1.3: ROC curve plot where blue line indicates a nearly perfect classification, green line a good classification and dashed red line signifies random guessing.

where,

TPR is True Positive Rate (Recall) i.e. liked stimuli that were predicted correctly,

$$TPR = \frac{TP}{TP + FN} \quad (1.2)$$

FPR is False Positive Rate i.e. liked stimuli that were predicted as disliked,

$$FPR = \frac{FP}{FP + TN} \quad (1.3)$$

ROC curve is based on these two metrics that evaluates the model's predictive power on a binary classification problem. Bigger AUC represents a better classifier, and the reference (the red line) in the figure 1.3 represents random guessing.

Finally, the confusion matrix that shows the values of TP, FP, TN, FN gives an idea about the prediction vs actual values comparison in table 1.1.

Table 1.1 : Confusion Matrix

		Predicted Condition	
	Population = P + N	Positive (PP)	Negative (PN)
Actual Condition	Positive (P)	True Positive (TP)	False Negative (FN)
	Negative (N)	False Positive (FP)	True Negative (TN)

1.5.2.2. Unsupervised learning

In unsupervised learning, predictions are made in the absence of labels. Most common unsupervised learning tasks are clustering, anomaly detection, and density estimation [49]. Unsupervised learning such as clustering, uses unlabeled data to group patterns at hand into meaningful clusters. Thus, it is done by a data driven manner, in which labels are obtained through data [50]. Examples for clustering applications can be given as recommendation engines, search engines, image segmentation and dimensionality reduction. Most common algorithms exploited for these kinds of problems are K-means, DBSCAN, agglomerative clustering and affinity propagation [49].

Unsupervised learning algorithms such as K-means and DBSCAN can also be used as an anomaly detection method. In addition to those algorithms, gaussian mixture models (GMM), minimum covariance determinant (fast-MCD), isolation forest, local outlier factor (LOF), and One-class SVM are another approach that detects outlier in a particular problem [49].

Lastly unsupervised learning can be used in density estimation. It is generally used in data visualization and analysis, by estimating the probability density function of the random process that formed the dataset. GMM and DBSCAN can be used as a density estimation method [49].

1.5.2.3. Ensemble learning

Every algorithm has strengths and weaknesses according to certain types of problems. These properties should be understood clearly to come up with a better algorithm which uses multiple algorithms by using one's strength to complement the other's weaknesses. In certain cases, it may be impossible to find a single algorithm that achieves the best accuracy or any other chosen score metric, thus combining two or more classifiers might be a good idea in these situations, and we call this method ensemble learning [37].

Most popular ensemble methods are voting, bootstrap aggregating (bagging), pasting, boosting, stacking approaches. Voting method includes more than one classifier for classification problems which can dominate individual classifiers that try to make a prediction alone. Diversity of the classifiers in this ensemble method is important as independence between them leads to different types of error that benefits the accuracy of the ensemble model. On the other hand, bagging technique uses only one type of algorithm for every predictor but training them on different subsets taken from the original dataset. There is one condition that should be satisfied for it to be called bagging which is making sampling with replacement. However, when sampling is done without replacement, it is called pasting [49], [51]. In boosting ensemble learning methods, the idea is to train the predictors sequentially where each predictor tries to correct its predecessor. Most popular examples are Adaptive Boosting, Gradient Boosting, Extreme Gradient Boosting (XGBoost), LightGBM, and CatBoost [49]. Lastly, stacking algorithms, which is short for stacked generalization, trains a model from an ensemble of algorithms instead of aggregating those classifiers' predictions. Final model is called a meta learner or blender, which takes the predictions of those classifiers as input or training data to make the final prediction [49].

2. EXPERIMENT DESIGN AND METHODOLOGY

In this section, design and methodology of the experiments will be elaborated to answer the thesis question. Due to the nature of the problem, an iterative approach has been followed to build the best model possible. When doing that, mainly, generic CRISP-DM reference models will be abided by to implement a good structure for this section. Thus, this part will consist of five phases i.e., business understanding, data understanding, data preparation, modeling, and evaluation. The only differences compared to the CRISP-DM model will be the exclusion of the development phase and modifications of those five phases [52]. The main framework of the CRISP-DM model can be found in figure 2.1.

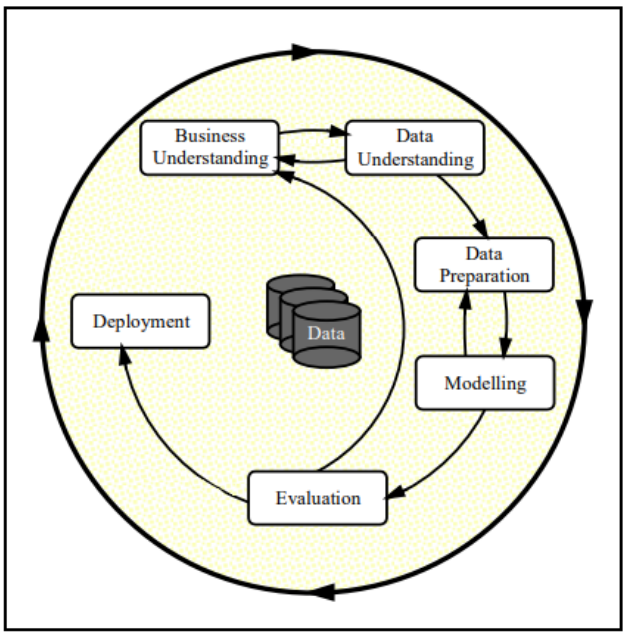


Figure 2.1: Phases of the CRISP-DM Process Model for Data Mining [52].

The main goal of this research is to build a model that predicts the liking preference of individuals based on fNIRS measurements. Data mining processes will be performed in Python language and following sections will provide detailed information on these processes.

2.1. Business Understanding

The main goal of this research is the liking prediction of individuals using five machine learning algorithms. To achieve that, first, various images of objects were

shown to individuals and asked them to indicate their liking preference while measuring their prefrontal hemodynamic activity with fNIRS. Then, these measurements were trained using five supervised learning algorithms to achieve the primary goal of implementing the best possible liking prediction model.

As a secondary goal, various feature extraction and methodologies were implemented to further improve the best algorithm's performance that was selected in the first goal.

Finally, the last side goal was to apply the wrapper method on the best model selected from the feature extraction step to come up with a better approach.

2.2. Experimental Setup

In this section, a brief explanation of the experimental setup will be given.

29 participants (18 female) in the age range 22 – 42 have participated in this experiment. They were chosen randomly from a consumer database, being all right-handed, which is tested by the Edinburgh handedness survey [53], for the purpose of preventing any variation in functional response due to lateralization biases. The data of two subjects has been removed due to not making any decisions of choosing his/her preference of liking. This study was approved by MEF University human subjects research ethics committee with the approval number E-47749665-050.01.04-1113 and written informed consent has been obtained before the experiment.

The task included 60 trials where subjects were asked to give a response on whether they like the image or not. One trial can be seen as a block, where participants had 5 seconds to view the image, 3 seconds to decide according to their choice of liking or disliking, followed by 8 seconds of fixation. Thus, each block lasted a total of 16 seconds, making the duration of the experiment to 16 minutes. There were two keys to be pressed upon to indicate their preference of liking or disliking, which were randomly switched in each block to avoid lateralization of brain function. The images consisted of real-life objects. The participants were informed to press either of the buttons to indicate their preference. One of the 60 stimuli is given in the figure 2.2 below.

Figure 2.2: One of the 60 Visual Stimuli.

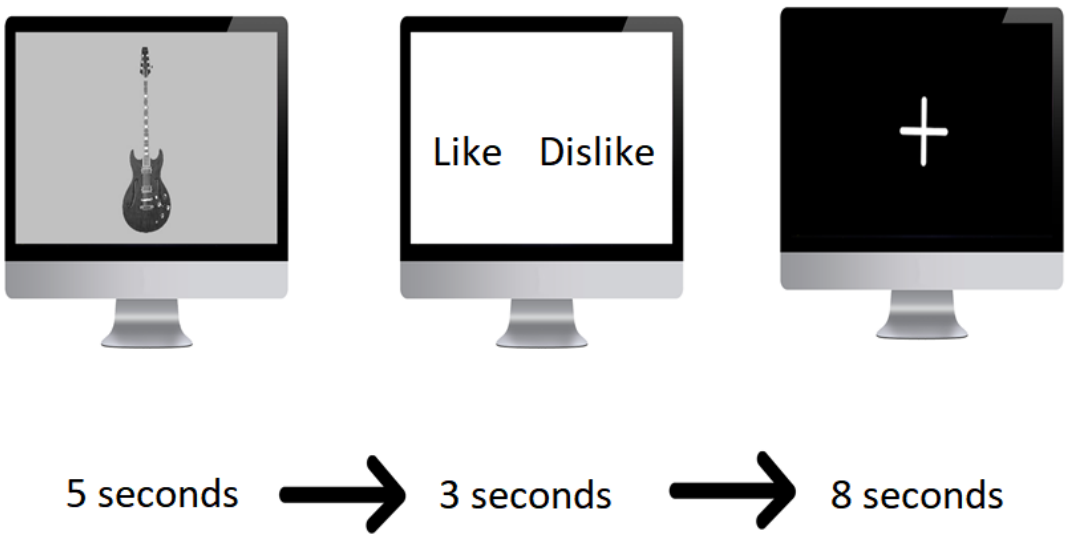


Figure 2.3: Experimental Process Steps.

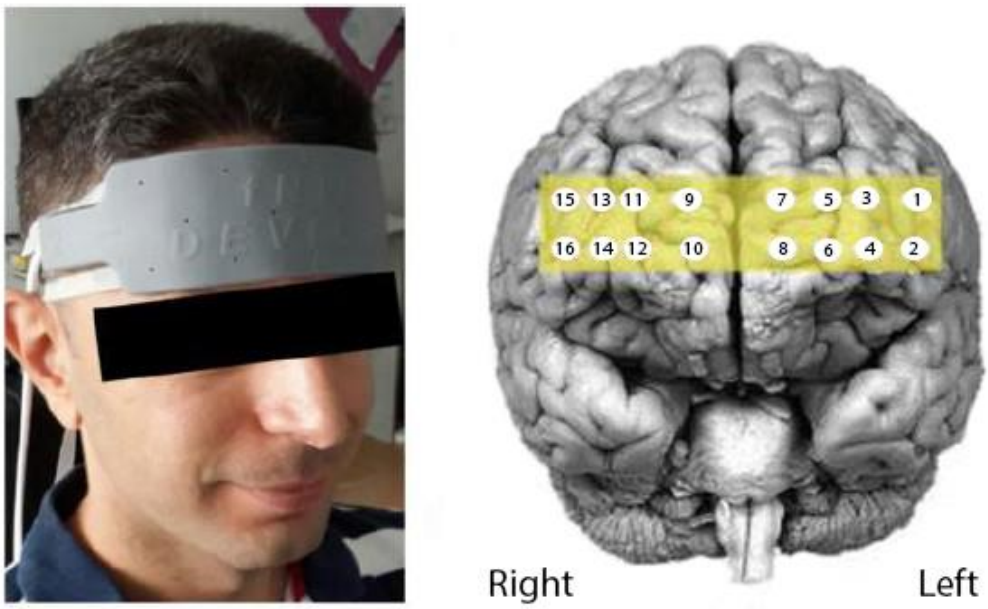


Figure 2.4: fNIR sensor pad placement on the forehead (left), projection of optodes on the prefrontal lobe (right).

Prefrontal cortex of each subject was continuously recorded using a fNIRS device made of a sensor pad that holds 4 light sources, 10 photodetectors to acquire oxygenation measures and 16 optodes placed on the forehead of the subject (figure 2.4). The sensor is capable of measuring data from a depth of 1.5cm and receiving frequency of 2 Hz from 16 optodes. Optode locations correspond to specific prefrontal cortex regions such as optode 1-4 takes data from dlPFC, optodes 5-8 from medial anterior PFC [54], optode 5-6 from left dorsomedial PFC, and lastly optode 7-8 from frontopolar PFC regions. For the right lobe of the brain, optodes 9-16 are responsible for the regions as defined for the left part, symmetrically. FNIRS takes these continuous measurements with respect to the changes in oxygenation of these regions to determine the neural activity, as hemodynamics pertains to functional brain activity with the aid of a mechanism noted as neurovascular coupling [55]. Then, the acquired fNIRS signals are preprocessed by COBI Studio Software.

For each participant, like/dislike decisions, response time and raw fNIRS measurements were acquired for all 60 images. This raw data was put into a filtering of high frequency noise due to respiration and cardiac pulsation with a finite impulse response, linear phase filter with order 20 and cut-off frequency of 0.1 Hz [1], [54]. Then, motion artifacts were detected and removed following a sliding windows motion artifact filter [56].

The average of HbO, HbR, Hbt and Oxy concentrations were calculated with respect to each block which consists of image viewing and decision-making phases, and since there were 16 optodes, total of 128 features were obtained by multiplying these numbers and steps (4 x 2 x 16). Then, all data from 60 trials are put together into a dataset for each participant as a Comma Separated Values (CSV) format. Then, this csv file was imported into an IDE of python to further processing of the data to put into machine learning algorithms.

The next chapter will further explain and analyze this CSV data at hand.

2.3. Data Understanding

As mentioned in the previous section, the dataset contains information of fNIRS average oxygenation measurements which were taken from 28 participants. There are a total of 128 of these oxygenation biomarkers, and additional features i.e., response time, sex, and liking preferences were also present. Other features such as demographic ones (e.g., age, education) were also available. As 60 images had been shown to participants to indicate their liking preference, with the participant count of 28, a total of 1.680 rows of data was at hand at the beginning of the data cleaning process.

The liking preference was the dependent variable which could take either 0 or 1 binary values that represent ‘dislike’ and ‘like’ respectively. HbO, HbR, Hbt, and Oxy features were predictors for liking preference which were continuous variables. Each of these four feature categories have two instances of measurements where first one measured when participant sees the stimuli and second one when participant indicates his liking preference, which were denoted with G and K abbreviations at the start of the feature’s naming, respectively. Below in table 2.1, a representation of all features has been presented with their type and description. Following data preparation and analysis steps will include:

1. Statistical analysis of the data: Basic and descriptive statistics i.e., average, min, max values, distribution of all categorical and numerical features, standard deviation, percentiles will be all presented.

2. Missing values in the data: Analysis of missing percentage and count of each feature will be given.

3. Outlier Detection and Analysis: Outlier detection will be made on each feature to decide whether to remove some of them or not depending on their value contribution to the dataset.

4. Correlation Analysis: Correlation heatmap matrix will be implemented to see the relation between dependent variable liking preference with predictor variables of fNIRS measurements. Besides, due to the size of the predictor features, top highly correlated predictors (> 0.5 for positively correlated or < -0.5 negatively correlated) will also be given.

Table 2.1 : Description and Types of Features

Feature	Type	Description
Age	N	Between 22 - 42
Sex	C	Female, Male
Education	C	Through High School to PhD Degree
Participant No	N	Enumeration of participants through 1 to 29
Stim_ID	N	Enumeration of images that were shown to participants through 1 to 60
RespTime	N	Response Time of Participants indicating their liking preference
Response	C	Left, Right
Like_OR_Nolike	C	True, False
Goxy1-2-3…16	N	Value of difference between HbO and HbR concentrations when viewing the stimulus through optodes 1 to 16
GHbr1-2-3…16	N	Concentration changes in deoxy-hemoglobin when viewing the stimulus through optodes 1 to 16
GHbo1-2-3…16	N	Concentration changes in oxy-hemoglobin when viewing the stimulus through optodes 1 to 16
GHbt1-2-3…16	N	Concentration changes in total hemoglobin when viewing the stimulus through optodes 1 to 16
Koxy1-2-3…16	N	Value of difference between HbO and HbR concentrations after deciding liking preference of the stimulus through optodes 1 to 16
KHbr1-2-3…16	N	Concentration changes in deoxy-hemoglobin after deciding liking preference of the stimulus through optodes 1 to 16
KHbo1-2-3…16	N	Concentration changes in oxy-hemoglobin after deciding liking preference of the stimulus through optodes 1 to 16
KHbt1-2-3…16	N	Concentration changes in total hemoglobin after deciding liking preference of the stimulus through optodes 1 to 16

Following section will provide a detailed explanation of how the data preparation process has been handled.

2.4. Data Preparation

This section will include all phases (i.e., data cleaning, outlier detection and removal, imputation for missing values, data clustering, frontal alpha asymmetry index, one-hot encoding, and standardization) that will make the data ready to be put into prediction models.

2.4.1. Data cleaning

It was seen that data was erroneous that should be dealt with. Two participant's data had all fNIRS measurements values, but they had not indicated their liking preference for any stimuli. For this reason, their data had been removed. In addition, one response corresponding to a stimulus of a participant was invalid due to pressing the wrong button, thus it had also been removed from the data.

2.4.2. Outlier detection and removal

Outlier analysis is essential while dealing with brain imaging measurements. Furthermore, some machine learning methodologies are also sensitive to outliers, meaning their predictive power reduces in presence of such values. Since fNIRS measurements were used with several supervised learning methods to predict the liking preference of participants, it was mandatory to exclude these outlying points from the data set. There are various methods to deal with outliers which were mentioned in the theoretical part.

In this study, as an initial method, boxplots had been used to spot outlying values and the IQR method had been used to remove those outliers. Thus, the points lay below lower boundary, and above upper boundary were masked as Not-Applicable (NA) values according to the given formula of IQR and of those boundaries in the equations below to be later imputed. They were not completely removed as looking at all trials or records, at least one outlier was there for nearly half of the available records. That meant removing half of the data, thus it was inconvenient in doing so.

$$IQR = Q3 - Q1 \qquad (2.1)$$

$$Lower\ Boundary = Q1\ - 1.5\ \times IQR \quad (2.2)$$

$$Upper\ Boundary = Q3 + 1.5\ \times IQR \quad (2.3)$$

Later, as an attempt to improve the performance of the models, a different approach was used as an application of outlier removal. Same as the first method, outliers were spotted with the IQR method for each feature as explained above. However, instead of masking those outliers, they were capped to the lower and upper boundaries of IQR. This capping was thought to be more accurate as those outlier values may have critical information by having those drastic measurements in prediction of liking preference.

2.4.3. Imputation

Incomplete data is unavoidable in most cases, e.g., missing value was forgotten or lost, value being not applicable meaning it does not exist for a given case. Because certain algorithms require complete data and do not work in the presence of missing values e.g., k-NN, neural networks [37]. To deal with this problem, one could remove every row that has at least one missing value. However, this approach would result in greatly reducing the data size which might be valuable even though it has missing values, thus it is an unfavorable approach. Another method is data imputation which is the process of statistically inputting missing values [57].

In this study, several imputation methods i.e., mean, median, multivariate imputation by chained equations (MICE), iterative imputation and k-Nearest Neighbors (KNN), neighbor imputation have been utilized in order to complete the data. MICE was applied using “impyute” library [58], iterative, KNN and median imputation was implemented using scikit-learn library [59], on the other hand mean imputation was basically employed with Pandas library’s dataframe functions. Neighbor imputation was a special one that was created only for this study that utilizes the neighborhood of optodes to come up with an accurate imputation.

All of these imputations were compared with original incomplete data with respect to their probability distribution functions by plotting them. Results will be shown in the implementation and results section.

2.4.4. Data clustering based on fNIRS features

Data clustering is an unsupervised method to find similar patterns between data points to group them together [50]. In this study, in order to improve the models' performances, K-means clustering was applied to group the hemodynamic responses of participants by using scikit-learn library's methods [60]. To find the optimal group number, silhouette scores were calculated and visualized in a line graph with group numbers ranging from 2 to 19.

2.4.5. Frontal-alpha-asymmetry Index

Frontal Alpha Asymmetry (FAA) is the difference in alpha power activity in right and left prefrontal regions according to the effect of a stimulus, corresponding to the negative and positive emotions respectively. Therefore, when an individual is exposed to a positive stimulus, there is a tendency to respond more intensely that is caused by activation of left regions of the brain, whereas being affected by a negative stimulus increases the tendency to react more intensely with the right frontal activity increasing.

The possibility of FAA effect has been considered by calculating a FAA index from measurements of hemodynamic responses of the prefrontal regions. Ramirez et al.[6] investigated this FAA effect by creating an index from EEG signals by taking the difference of the left and right hemisphere's alpha powers. Therefore, that study has taken as the basis for creating a similar index but from fNIRS measurements, taking the difference of left and right hemisphere's relative change in oxygenation signals i.e., oxy values (ΔHbO_2 - ΔHbR).

2.4.6. One-hot encoding

There was one categorical variable used in the prediction models, which was cluster numbers based on fNIRS measurements. Since certain machine learning algorithms only work on numeric variables such as SVM in this study, these categorical variables needed to be converted into numerical form. One-hot encoding comes in handy in this situation in conversion of these categorical variables due to there not any ordinal relationship existing within these variables. If there was an

ordinal relationship between them, in that case integer encoding would be more adequate to use.

2.4.7. Standardization of the numeric data

Most of the machine learning algorithms struggle to perform well when the numerical features have varied scales [49]. Therefore, it is mandatory to put those values into the same scale before inputting them into the model. There are two ways to achieve that: normalization or with its alternative name min-max scaling, and standardization. In this study, standardization has been preferred due to being less affected by outlier values compared to normalization and not dealing with any neural network algorithms since some of them expect input values ranging from 0 to 1 [49].

Let denote the 128-dimensional feature set in this study as $Y = \{X_1, X_2, \dots, X_{128}\}$, thus it has a data matrix such as given below,

$$X_1, X_2, \dots, X_d = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix} \quad (2.4)$$

where, n is the number of trials that were shown to participants as stimuli which will later be given in next chapter, and d = 128 which are all hemodynamic response values taken from fNIRS measurements.

Therefore, to standardize these given 128-dimensional input features, standard score or also known as Z-score of each value had to be calculated with the given formula below,

$$Z(x_{ij}) = \frac{x_{ij} - \underline{x}_j}{\sigma_j} \quad (2.5)$$

where,

$\underline{x}_j$ is the mean of the jth sample,

σ_j is the standard deviation of the jth sample.

This scaling of the dataset had been done with *StandardScaler* of scikit-learn library by first fitting the scalar to the train set, and then using the fitted scaler to transform both train and test sets [59].

2.5. Modeling

The major goal of this study was to investigate the use of various machine learning algorithms which were Random Forest, SVM, KNN, XGBoost (XGB) and lastly Light Gradient Boosting Machine (LGBM), in classification models for determining the liking prediction of individuals based on fNIRS measurements. These are all supervised machine learning algorithms. Therefore, there were a total of five models that had been developed with these algorithms and were compared with each other by their evaluation of performance. The comparison details will be given in the evaluation section.

Another goal was to apply three feature extraction methods that were PCA, Isomap and t-SNE, to an attempt to further improve the models' score and compare their results within each other and also with the result of models without feature extraction. Thus, all five previously formed models were put into three feature extraction, resulting in 15 models.

The third goal was to investigate the power of an ensemble learning model, which was developed with algorithms that achieved the best results according to their evaluation. The details of this ensemble learning will be mentioned in a future chapter.

In the next section, with these feature extraction methods and ML algorithms, how the modeling framework was designed will be briefly given.

2.5.1. Modeling framework

In this section, an explanation of how the modeling framework was created will be explained. Since there were five algorithms and three feature extraction methodologies were used and additionally a wrapper method as feature selection, it would have been a little bit complicated and unoriented to compare all their results with each other, unless a modeling framework was designed. Therefore, there were six main frameworks defined i.e., main model without feature extraction, model with PCA, model with t-SNE and model with Isomap, model with frontal alpha asymmetry transformation and model with feature selection. In the main model, all ML algorithms were used and then the best algorithm was selected to be used in feature extraction

models, and the best model among feature extraction and main models was chosen to apply feature selection to further improvement of score.

In the next section, ML algorithms that were used in the main model will be mentioned indicating their properties and their hyperparameter tuning.

2.5.2. Machine learning algorithms

Random Forest (RF) is an ensemble learning algorithm consisting of multiple collections of decision trees. It is based on the concept of bootstrap aggregation, also called in short bagging, which is the resampling of the instances of data with replacement to reduce variance [51], [61]. Thus, with bagging shown in figure 2.5 below, instances of the data are randomly sampled multiple times for each predictor, and then put into training for each predictor.

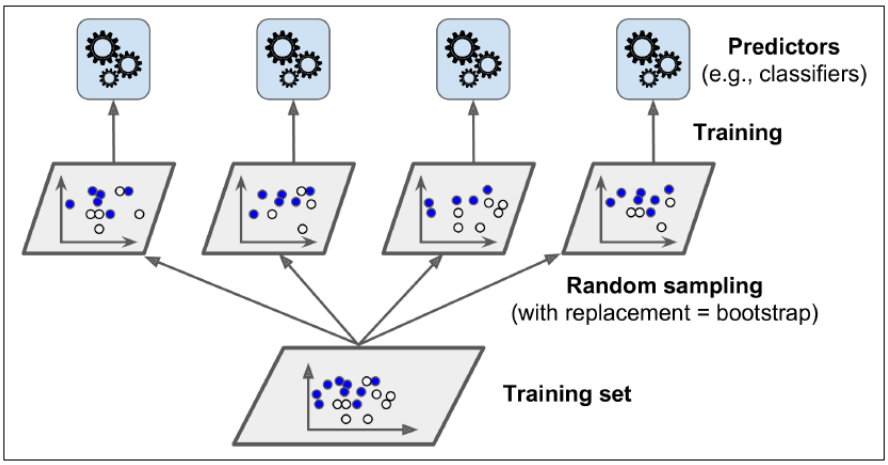


Figure 2.5: Bagging sampling and Random Forest [49].

In random forest, multiple trees are used in averaging predictions coming from them for regression problems, or in classification problems, taking the majority of the votes coming from each decision tree predictors. Thus, the final prediction will be that majority's choice. One critical thing is that all decision trees should be de-correlated with each other to produce a better prediction model, due to minimizing errors of those uncorrelated decision trees. In other words, the accuracy of the model depends on the strength of each tree classifier in a random forest, and also the dependency between all these trees [62]. In this study, *RandomForestClassifier* of scikit-learn was used with tuning different values of 'n_estimators' which represents the number of trees and

tuning ‘max_depth’ of those trees with using two different loss function for measuring the quality of the split of trees [59]. Tuned hyperparameters were shown below in table 2.2.

Table 2.2 : Random Forest Hyperparameter Tuning

max_depth	**3, 4, 5, 6, 7, 8, 9**
n_estimators	100, 300, 500
Criterion (loss function)	‘gini’, ‘entropy’

A Support Vector Machine (SVM) is an extremely strong and flexible machine learning model that can do regression, outlier identification, and linear or nonlinear classification [49]. It is based on statistical learning theory nominated by Vapnik [63] and also developed by him and his colleagues. In a N-dimensional space, this algorithm constructs a hyperplane or multiple hyperplanes that act as a boundary between two or more classes which separates them, in a classification problem. In order to achieve the best separation by minimizing the generalization error, the hyperplane with the largest functional margin, i.e. the distance between the nearest points from each class which are called support vectors, is selected as the boundary between those classes. Below in figure 2.6, there is an example for a hyperplane constructed by SVM algorithm for two classes that are linearly separable, with 3 support vectors sitting on each side of the hyperplane with all having equal distance to it. Therefore, those support vectors are critical in forming a hyperplane, and removal of them will lead to the alteration of the hyperplane [64].

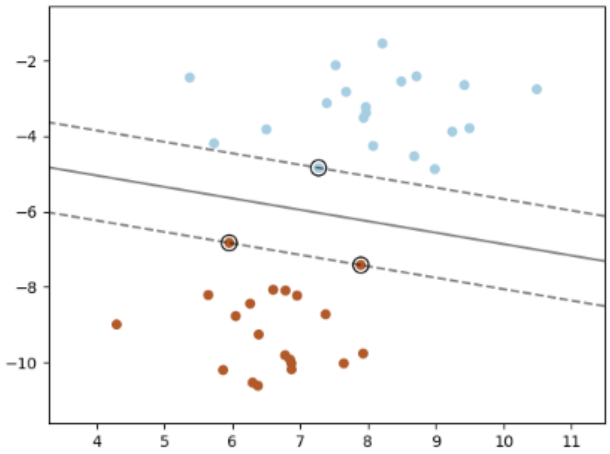


Figure 2.6: Support Vector Machine Hyperplane and Support Vectors [59].

Since 128-dimensional data was present at hand which cannot be linearly separable like the example above in figure 2.6, the so-called kernel trick had to be applied allowing nonlinear data to be separable by hyperplanes in a higher dimensional space. SVM with linear kernel had been chosen for this study as another similar study also used it and had satisfactory results with it [8]. Hyperparameters 'C' and 'γ' of SVM were tuned with grid search to find the optimal parameters to create a good model (table 2.3).

Table 2.3 : SVM Hyperparameter Tuning

C	**0.1, 1, 10, 100**
gamma	0.1, 1, 5, 10

K-Nearest Neighbor (KNN) may be called the simplest algorithm across all supervised learning methods, especially compared to other algorithms used in this study. Despite its simplicity, it has achieved satisfactory results in many classification and regression problems. It computes the distance of a point to other points around it to classify based on the majority vote looking at the nearest neighbors' classes while specifying the neighbor number. Euclidean distance was utilized as distance metric and K-neighbors had been obtained through grid searching K from 2 to 20 (table 2.4).

Table 2.4 : KNN Hyperparameter Tuning

n_neighbors	**List(range(2,20))**

XGB [65] and LGBM are optimized boosting techniques for gradient-boosted algorithms. Like RF, they also utilize more than one decision tree, thus being also an ensemble learning technique. The difference between RF and these gradient-boosted tree algorithms is how the trees are built. In RF, all trees are built independently while in gradient boosting, trees are built one at a time. In Gradient Boosting, an ensemble of weak learners is utilized to enhance the model performance by filtering out the instances that cannot predict well, and then form new weak learners based on that filtering.

XGB abbreviation represents 'Extreme Gradient Boosting' where gradient boosting is based on the paper of Friedman [66]. Hyperparameters were tuned with grid search as shown in table 2.5.

Table 2.5 : XGBoost Hyperparameter Tuning

booster	**'dart'**
learning_rate	0.03, 0.05, 0.1
max_depth	3, 4, 5, 6, 7, 8, 9
gamma	1, 5, 10
n_estimators (number of trees)	100, 300, 500
min_child_weight	3, 4, 5

LGBM uses leaf-wise tree growth contrary to level-wise tree growth of XGB. This enables LGBM to achieve higher loss reduction, thus higher accuracy compared to XGB, but this may cause higher overfitting on the training set. This may lead to tuning parameters more carefully in LGBM, which makes XGB a more robust algorithm than LGBM. Hyperparameter tuning for LGBM is given in the table 2.6 below. Num_leaves parameter was given 2^(max_depth) for each max_depth value to control the complexity of the tree. Leaf-wise trees are much deeper than level-wise trees so to not cause overfitting, this formulation was taken as stated in LGBM's documentation [67].

Table 2.6 : LGBM Hyperparameter Tuning

boosting_type	**'dart'**
learning_rate	0.03, 0.05, 0.1
max_depth	3, 4, 5, 6, 7, 8, 9
max_bin	10, 50, 100
num_iterations (number of trees)	100, 300, 500
path_smooth	3, 4, 5
min_child_samples	List(range(20, 40))

In the next section, dimensionality reduction techniques that were used in feature extraction models will be explained.

2.5.3. Dimensionality reduction models (feature extraction)

High dimensionality in the feature set of the data causes problems for classification algorithms which are increased computational cost and memory usage. Apart from hardware issues, high dimensional data also consists of inessential, unnecessary, or deceptive features which makes it difficult to process data to come up with a good learning outcome. In other words, it can include highly correlated features that should be dealt with. Therefore, performance of the model could be diminished in presence of noise-containing or highly correlated features. There are some techniques to reduce the dimensionality of the dataset to provide some solution to these issues,

one of them is feature extraction. This method reduces the dimensionality of the data by transforming the original features into features that are more representative and useful to be implemented in a model. The information of the former feature space is retained mostly in this process. Feature extraction can reduce complexity of the data and by linearly combining the original variables for each feature set, it can represent each of those variables belonging to its feature space [68]. In addition, by representing the data in a lower dimension, it could reduce the runtime of an algorithm which is also crucial [69].

Maybe the most prominent feature extraction method is Principal Component Analysis (PCA) which is also the most used one. It is a non-parametric technique to reduce the dimensionality of the relevant dataset by extracting the most useful and relevant information from it, eliminating the noisy and misleading features. It reduces the redundancy, which is measured through covariance, and maximizes the information obtained which is measured through the variance [68].

Thus, in this study, as a means to reduce the 128-dimensional data to improve the performance of the models by removing irrelevant and noise-making data, numerical features that consist of fNIRS measurements were transformed using PCA. This was done by keeping %90 variance of the original feature set, to keep the maximum statistical information from our extracted feature space. In the end, reduced feature space consisted of few principal components that explains 90% variance of the original data in which the first one represents nearly 40% of the variance and reducing with each principal component added, totaling to 90%. These principal components are actually eigenvectors of the data's covariance matrix. 90% of variance is also kept the same for other two feature extraction methodologies that will be mentioned next, to properly compare their performances on the dataset.

Another feature extraction method used is Isomap whose main aim is to capture the intrinsic degrees of freedom of input variables. In addition, this method was developed to come up with a solution to nonlinear feature sets which are invisible to both PCA and Multidimensional Scaling (MDS). It was built on top of classical MDS but with maintaining the intrinsic geometry of a data manifold by calculating the geodesic distance between all points and finally making an estimate with finding shortest paths in a graph edge linking each data point's neighbors on the manifold. If

there is enough data present, this method is guaranteed to obtain the true geometric structure and dimensionality from high dimensional nonlinear manifolds [70].

Lastly, t-Stochastic Neighbor Embedding (t-SNE) was used as a dimensionality reduction method. The main usage or development of t-SNE is to visualize high dimensional data in two- or three-dimensional space. However, in this study, it was used as a feature extraction technique with a secondary goal to compare its performance against PCA and Isomap. It is a variation of Stochastic Neighbor Embedding that enables easier optimization and better visualizations. It can acquire most of the local structure of the high dimensional data with a great effort. Same as Isomap, it is a nonlinear dimensionality reduction technique that models the probability distribution of similar objects that are assigned high probability and dissimilar objects that are assigned low probability, and then does the same distribution with the points in lower dimensional space with using Student's t-distribution and minimizing these two distributions with Kullback-Leibler divergence (KL divergence) [71]. When calculating those conditional probabilities, the variance is set to 90% the same as PCA and Isomap in this study.

In the next section, the application of the feature selection model will be defined.

2.5.4. Wrapper model

As an attempt to improve predictive power, certain fNIRS features might be more important than others in predicting the liking of a stimulus. Therefore, a wrapper approach has been implemented as a model, which is a feature subset selection method that uses an induction algorithm as a black box [9]. Black box signifies that no knowledge of the algorithm is needed when searching for the best feature subset. Greedy approach has been considered as the search algorithm for this model, and feature subset size was taken as half of the number of hemodynamic features. Forward selection was used as an initial state, meaning at the beginning of the search process, the feature subset was empty. 2-fold cross validation was selected as an evaluation method instead of higher fold approaches due to limited computational power. In each iteration, the feature which gives the best results among others was added into the

feature subset and the process continued until subset size reached half of the number of original feature size.

In the next section, how the model performances were evaluated will be explained.

2.6. Evaluation

In the evaluation of classification models, two methodologies were used in this study. First one was using leave-one-group-out cross validation for tuning hyperparameters for all models, and the second one being a permutation test to evaluate if the final prediction on the test set is statistically better than by chance.

2.6.1. Leave-one-group-out cross validation

Cross-validation is an evaluation technique that analyzes a model if it is generalizable on an independent data set. It iterates over the data to divide it into train and test sets multiple times with a given random state, to be tested by a model. This way, several evaluations are made by the stated number of times, making the model more reliable by eliminating selection biases and also reducing the chance of overfitting. All results of these validations are combined into the final score by taking average, giving an estimate how a model could perform in a nearly real-life environment. It is especially useful if the data set is too small, compared to making a simple train/test split or so-called hold out validation.

In leave-one-group-out cross validation (LOGOCV), folding is made in such a way that each group is selected alone as a test set for prediction. Therefore, with given grouping information for all records, it is guaranteed that each group appears exactly once in the validation set across all folds. Groups were set as participant ids to ensure that the model generalizes well by avoiding a participant's records to appear in both sets. Taking that into consideration, data leakage was aimed to be kept at minimum.

In LOGOCV, the number of folds was set as $k = 27$, meaning one of the k subsets was taken as validation set in each iteration over 27 folds. Remaining 26 subsets were taken as training set. In each fold, each pair of parameters had been

searched and obtained scores were recorded. For each of the pairs, average results in all folds were taken and the highest one was selected as the best pair of parameters.

When k value gets close to the total record number, or in this version of cross-validation being close to the number of groups, the statistical power of the evaluation gets stronger. However, as k increases, the computational power also drastically increases. This is one of the drawbacks of LOGOCV. In figure 2.7, there is a representation of 5-fold cross validation, which might give an idea about LOGOCV.

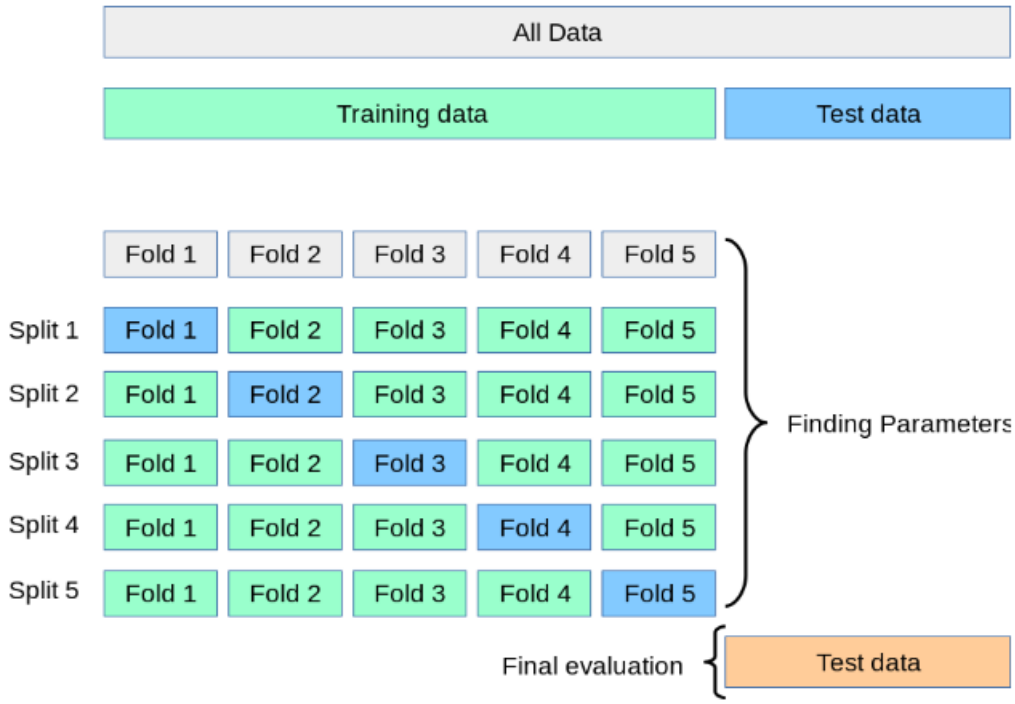


Figure 2.7: 5-fold Cross Validation for Hyperparameter Tuning [59].

Following cross validation, 27 scores were gathered for each model. F1-score metric was taken as an evaluation metric to compare the performance of the models. All these scores were averaged and then the comparisons were made based on these averages. In addition to F1-score, precision and recall scores were also reported.

The F1-score of a model is defined in equation 3.3, it can either be defined as the harmonic mean of the precision and recall or in terms of the second equation that its components are explained below.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (2.6)$$

where positive represents liking and negative being disliking,

TP is True Positive e.g., where the target value is like and it is predicted as like,
FP is False Positive e.g., where the target value is dislike and it is predicted as like,
FN is False Negative e.g., where the target value is like and it is predicted as dislike.

Precision can be defined as the fraction of true positive among true positive and false positive, i.e. accuracy of correctly predicting the positive class. On the other hand, recall, also known as sensitivity, can be defined as the division of true positives to the total number of positive records, meaning the accuracy of only predicting the positive class.

2.6.2. Permutation test

Permutation test was applied on all F1-scores to show the significance of these classification scores. Null hypothesis (H_0) assumes that the classifier cannot find a real connection between the data and class labels i.e., they are independent from each other. Therefore, standard permutation test [72] was applied by permuting 'like' and 'dislike' labels 100 times randomly, showing if the classifier can reject H_0 with low p-value that tells the actual test score obtained is better than by chance alone. Probability distribution of permuted classification scores and the actual test score will be plotted and given such as in the figure 2.8 below, which states that the score obtained by the original model is better than by chance (*p=0.001)*.

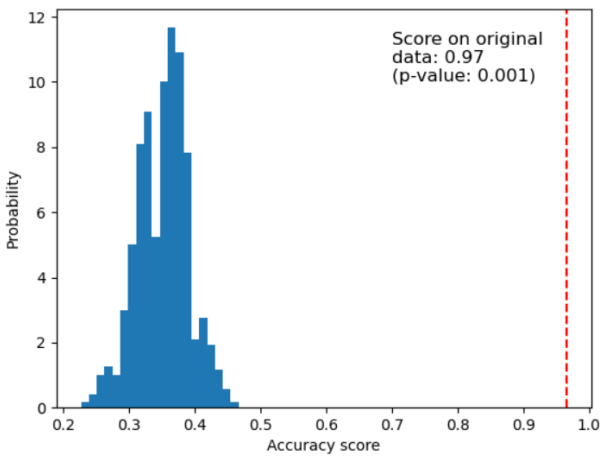


Figure 2.8: Histogram of the Null Distribution (Permutation Scores) and the actual test score showed as a dashed red line [59].

2.6.3. Wilcoxon signed-rank test

Final evaluation will be to use Wilcoxon Signed-Rank Test i.e., a non-parametric test that assesses the null hypothesis that the two related samples come from the same distribution. Therefore, statistical significance of each pair of models will be tested by comparing the distributions of their F1-scores.

3. RESULTS AND DISCUSSION

This chapter will give results of each part written in experiment design and methodology by the same order with their analysis and discussion.

3.1. Business Understanding

To investigate the main goal and side goals stated in the previous chapter, following steps and their results will be analyzed and discussed.

1. Exploratory Data Analysis including data cleaning, descriptive statistics, missing value, outlier, and correlation analysis
2. Outlier removal and imputing missing values
3. Data clustering with K-Means
4. One-hot encoding of categorical values
5. Standardization of train and test sets
6. Training with five algorithms on train and test sets
7. Comparison of those algorithms and answering primary research goal
8. Implementing three feature extraction methods and comparison of their performance to answer secondary goal
9. Feature selection with wrapper approach and its performance evaluation on the best algorithm with comparison to the original data set to answer secondary goal.

3.2. Data Understanding

Before proceeding to the descriptive statistics, data was inspected if there were invalid values. First observation was the minimum response time being zero. Minimum response time for image stimuli can be seen as zero. When further investigation was made by looking at the ‘Response’ categorical value, zero values of ‘Response Time’ belonged to NA values of ‘Response’ feature which means that certain trials were ended without indication of any preference of liking of a participant or pushing the wrong button when indicating their preference for the related stimuli. Hence, these trials totaling 115 records were removed from the data. 60 of these records belong to all trials of participant number 18, so with the initial exclusion of a participant that were mentioned in the previous chapter, participant number reduced

from 29 to 27 with this additional one. Final participant number can be seen in the figure 3.1.

```
array([ 1,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 19,
       20, 21, 22, 23, 24, 25, 26, 27, 28, 29])
```

Figure 3.1: Array of Remaining Participant Numbers

Descriptive statistics of the numeric values are given in table 3.1. These statistics include number of records, mean, standard deviation, minimum, maximum and quartile values for each feature. There are a total of 1.565 records in the data set which was reduced from 1.680 due to invalid responses of participants in the experiment. Age of participants ranges from 22 to 39 with a mean value of 29.

Table 3.1 : Descriptive Statistics of Numeric Features

Feature	Count	Mean	Std. Deviation	Min.	25%	50%	75%	Max.
Age	1.565	29,0	5,1	22	24	29	32	39
RespTime	1.565	1.115	430	413	799	1.016	1.315	2.994
Goxy1	1.451	-0,03	0,34	-3,09	-0,19	-0,03	0,13	2,77
Goxy2	1.469	-0,02	0,33	-3,64	-0,17	-0,03	0,11	4,27
Goxy3	1.434	-0,05	0,35	-1,76	-0,24	-0,04	0,16	1,52
Goxy4	1.495	-0,02	0,31	-3,40	-0,17	-0,02	0,12	2,15
Goxy5	1.563	0,00	0,29	-1,64	-0,17	-0,01	0,18	1,10
Goxy6	1.557	-0,01	0,32	-3,27	-0,16	-0,02	0,13	2,69
Goxy7	1.561	0,00	0,37	-2,29	-0,21	0,00	0,22	1,52
Goxy8	1.495	0,01	0,28	-3,96	-0,11	0,00	0,12	2,48
Goxy9	1.559	0,01	0,34	-1,98	-0,19	0,01	0,21	1,49
Goxy10	1.484	0,01	0,29	-1,58	-0,11	0,00	0,14	3,79
Goxy11	1.561	0,00	0,42	-2,30	-0,23	0,00	0,23	1,73
Goxy12	1.533	-0,02	0,30	-4,91	-0,16	-0,02	0,12	2,94
Goxy13	1.538	-0,03	0,53	-3,57	-0,30	-0,04	0,24	3,43
Goxy14	1.458	-0,03	0,31	-3,84	-0,17	-0,03	0,11	2,44
Goxy15	1.117	-0,01	0,42	-6,48	-0,20	-0,01	0,18	4,44
Goxy16	1.064	-0,01	0,29	-1,98	-0,12	-0,01	0,10	2,12
Ghbr1	1.445	0,01	0,22	-4,16	-0,06	0,01	0,08	2,23
Ghbr2	1.469	0,02	0,31	-4,81	-0,08	0,02	0,11	4,36
Ghbr3	1.434	0,02	0,15	-0,75	-0,06	0,02	0,10	1,22
Ghbr4	1.495	0,01	0,25	-2,58	-0,07	0,01	0,09	3,93
Ghbr5	1.563	0,01	0,15	-0,91	-0,07	0,01	0,08	0,99
Ghbr6	1.557	0,01	0,27	-3,01	-0,08	0,01	0,10	4,49
Ghbr7	1.561	0,01	0,23	-1,62	-0,09	0,00	0,10	2,52
Ghbr8	1.495	0,00	0,33	-2,84	-0,10	0,00	0,10	4,78

Ghbr9	1.559	0,01	0,17	-1,01	-0,09	0,00	0,10	0,97
Ghbr10	1.484	0,01	0,33	-4,19	-0,10	0,01	0,12	2,31
Ghbr11	1.561	0,01	0,22	-1,18	-0,10	0,01	0,12	1,59
Ghbr12	1.533	0,02	0,30	-2,85	-0,08	0,02	0,13	5,38
Ghbr13	1.538	0,02	0,27	-1,99	-0,10	0,01	0,14	2,59
Ghbr14	1.458	0,03	0,30	-2,43	-0,08	0,02	0,14	4,13
Ghbr15	1.117	0,01	0,30	-3,58	-0,09	0,00	0,10	5,62
Ghbr16	1.064	0,03	0,40	-2,79	-0,10	0,02	0,16	3,08
Ghbo1	1.445	-0,02	0,26	-1,48	-0,15	-0,02	0,11	1,93
Ghbo2	1.469	0,00	0,28	-1,24	-0,14	-0,01	0,12	2,30
Ghbo3	1.434	-0,03	0,28	-1,63	-0,16	-0,02	0,13	1,27
Ghbo4	1.495	-0,01	0,29	-2,63	-0,14	-0,01	0,12	1,79
Ghbo5	1.563	0,01	0,26	-2,55	-0,13	0,01	0,15	1,07
Ghbo6	1.557	0,01	0,29	-1,41	-0,14	-0,01	0,15	3,20
Ghbo7	1.561	0,01	0,31	-3,53	-0,15	0,00	0,18	1,21
Ghbo8	1.495	0,01	0,26	-1,31	-0,12	0,00	0,14	2,04
Ghbo9	1.559	0,02	0,29	-1,76	-0,15	0,02	0,18	1,42
Ghbo10	1.484	0,02	0,28	-1,83	-0,12	0,01	0,16	2,07
Ghbo11	1.561	0,01	0,33	-2,28	-0,16	0,00	0,18	1,82
Ghbo12	1.533	0,01	0,28	-3,12	-0,13	0,01	0,13	2,71
Ghbo13	1.538	-0,02	0,36	-2,15	-0,20	-0,02	0,15	1,78
Ghbo14	1.458	0,00	0,28	-3,19	-0,12	-0,01	0,12	3,48
Ghbo15	1.117	0,00	0,28	-1,96	-0,15	0,00	0,14	1,58
Ghbo16	1.064	0,01	0,27	-1,77	-0,11	0,00	0,14	1,65
Ghbt1	1.445	-0,01	0,34	-5,55	-0,14	-0,02	0,11	3,11
Ghbt2	1.469	0,01	0,49	-5,35	-0,18	0,00	0,19	5,20
Ghbt3	1.434	-0,01	0,28	-2,10	-0,12	0,00	0,12	1,62
Ghbt4	1.495	0,00	0,44	-4,91	-0,17	0,02	0,17	4,47
Ghbt5	1.563	0,01	0,30	-3,47	-0,10	0,02	0,15	1,67
Ghbt6	1.557	0,02	0,46	-3,32	-0,18	0,01	0,20	5,71
Ghbt7	1.561	0,01	0,40	-4,77	-0,14	0,00	0,17	3,62
Ghbt8	1.495	0,01	0,52	-3,20	-0,20	0,00	0,22	5,59
Ghbt9	1.559	0,02	0,34	-2,77	-0,14	0,01	0,17	1,76
Ghbt10	1.484	0,03	0,55	-4,59	-0,21	0,01	0,25	4,11
Ghbt11	1.561	0,02	0,36	-3,27	-0,11	0,01	0,14	2,53
Ghbt12	1.533	0,03	0,49	-4,36	-0,17	0,01	0,24	5,85
Ghbt13	1.538	0,00	0,34	-2,74	-0,13	-0,01	0,12	2,81
Ghbt14	1.458	0,03	0,48	-4,14	-0,15	0,02	0,22	5,00
Ghbt15	1.117	0,00	0,38	-2,71	-0,15	0,01	0,16	4,77
Ghbt16	1.064	0,04	0,62	-4,37	-0,19	0,03	0,29	4,18
Koxy1	1.442	0,00	0,46	-4,83	-0,22	0,00	0,22	4,28
Koxy2	1.465	-0,01	0,43	-3,22	-0,20	-0,01	0,18	3,41
Koxy3	1.434	-0,03	0,47	-2,36	-0,28	-0,02	0,25	2,09
Koxy4	1.499	-0,01	0,41	-3,33	-0,20	-0,02	0,17	2,49
Koxy5	1.562	-0,02	0,39	-2,00	-0,25	0,00	0,22	1,43
Koxy6	1.558	0,00	0,42	-3,50	-0,21	-0,02	0,19	2,87
Koxy7	1.564	0,00	0,49	-3,13	-0,29	0,00	0,29	2,01

Koxy8	1.494	0,00	0,39	-4,07	-0,17	-0,01	0,18	2,74
Koxy9	1.557	0,00	0,45	-2,52	-0,27	-0,01	0,27	2,00
Koxy10	1.480	0,01	0,42	-3,11	-0,18	-0,01	0,18	3,27
Koxy11	1.559	-0,01	0,53	-3,19	-0,31	-0,01	0,28	2,71
Koxy12	1.534	-0,01	0,42	-5,07	-0,21	-0,02	0,17	3,96
Koxy13	1.536	-0,03	0,68	-5,86	-0,39	-0,02	0,33	3,93
Koxy14	1.459	-0,03	0,43	-3,86	-0,22	-0,03	0,16	2,50
Koxy15	1.111	0,00	0,50	-5,95	-0,25	0,01	0,25	2,49
Koxy16	1.062	0,00	0,46	-3,90	-0,17	0,00	0,15	4,91
Khbr1	1.442	0,00	0,30	-4,86	-0,08	0,00	0,10	3,37
Khbr2	1.465	0,01	0,41	-2,29	-0,14	0,00	0,15	4,17
Khbr3	1.434	0,02	0,21	-1,17	-0,09	0,02	0,12	1,34
Khbr4	1.499	0,01	0,36	-2,82	-0,10	0,01	0,13	3,99
Khbr5	1.562	0,03	0,21	-1,23	-0,09	0,02	0,13	1,44
Khbr6	1.558	0,02	0,37	-3,14	-0,11	0,01	0,14	4,58
Khbr7	1.564	0,01	0,31	-3,10	-0,12	0,02	0,14	2,53
Khbr8	1.494	0,01	0,46	-3,51	-0,14	0,01	0,17	4,92
Khbr9	1.557	0,02	0,23	-1,02	-0,10	0,02	0,14	1,52
Khbr10	1.480	0,02	0,49	-4,46	-0,13	0,02	0,18	4,27
Khbr11	1.559	0,02	0,28	-1,46	-0,11	0,02	0,16	1,97
Khbr12	1.534	0,02	0,47	-4,32	-0,13	0,01	0,15	5,53
Khbr13	1.536	0,02	0,34	-2,07	-0,15	0,01	0,17	2,83
Khbr14	1.459	0,02	0,46	-4,71	-0,13	0,02	0,17	4,30
Khbr15	1.111	0,01	0,33	-1,67	-0,13	0,00	0,13	5,19
Khbr16	1.062	0,02	0,72	-8,83	-0,18	0,00	0,21	6,98
Khbo1	1.442	0,00	0,35	-1,83	-0,17	0,00	0,17	2,42
Khbo2	1.465	0,00	0,40	-3,22	-0,19	-0,02	0,19	2,70
Khbo3	1.434	-0,01	0,36	-1,94	-0,20	0,00	0,19	1,51
Khbo4	1.499	0,00	0,39	-3,09	-0,18	-0,01	0,16	2,76
Khbo5	1.562	0,01	0,34	-2,73	-0,17	0,02	0,19	1,59
Khbo6	1.558	0,01	0,39	-1,71	-0,19	0,00	0,21	3,35
Khbo7	1.564	0,01	0,42	-3,61	-0,21	0,01	0,23	1,84
Khbo8	1.494	0,02	0,37	-1,98	-0,18	0,01	0,20	2,80
Khbo9	1.557	0,02	0,40	-2,06	-0,20	0,02	0,23	1,91
Khbo10	1.480	0,03	0,42	-1,95	-0,17	0,02	0,21	3,80
Khbo11	1.559	0,01	0,43	-2,39	-0,21	0,01	0,23	2,25
Khbo12	1.534	0,01	0,39	-3,93	-0,17	0,00	0,18	2,73
Khbo13	1.536	-0,01	0,46	-3,03	-0,25	-0,01	0,23	2,37
Khbo14	1.459	-0,01	0,39	-5,79	-0,18	-0,02	0,16	2,58
Khbo15	1.111	0,01	0,37	-1,87	-0,20	0,00	0,19	1,62
Khbo16	1.062	0,02	0,43	-3,92	-0,17	0,00	0,18	4,55
Khbt1	1.442	0,00	0,45	-6,45	-0,18	0,01	0,17	2,74
Khbt2	1.465	0,01	0,68	-5,51	-0,29	-0,01	0,28	5,63
Khbt3	1.434	0,01	0,37	-2,22	-0,15	0,01	0,18	1,81
Khbt4	1.499	0,00	0,63	-5,14	-0,24	0,00	0,26	5,04
Khbt5	1.562	0,04	0,40	-3,46	-0,14	0,03	0,21	2,69
Khbt6	1.558	0,03	0,63	-3,42	-0,25	0,02	0,29	5,66

Khbt7	1.564	0,03	0,54	-5,05	-0,18	0,03	0,24	3,70
Khbt8	1.494	0,03	0,74	-5,50	-0,28	0,01	0,33	5,77
Khbt9	1.557	0,04	0,47	-2,75	-0,17	0,03	0,23	2,97
Khbt10	1.480	0,05	0,81	-6,27	-0,25	0,04	0,36	5,44
Khbt11	1.559	0,04	0,48	-3,01	-0,14	0,03	0,20	2,63
Khbt12	1.534	0,02	0,75	-7,07	-0,26	0,02	0,31	5,99
Khbt13	1.536	0,01	0,45	-3,54	-0,16	0,01	0,18	3,14
Khbt14	1.459	0,01	0,75	-9,22	-0,26	0,01	0,29	5,26
Khbt15	1.111	0,02	0,48	-2,87	-0,19	0,01	0,23	4,44
Khbt16	1.062	0,04	1,09	-12,75	-0,31	0,00	0,38	11,53

For categorical features, bar plots are presented in figure 3.2 below, showing frequencies of them for each of their distinct values against the target values of 'Like' and 'NoLike'. There is not too much imbalance between two target values, corresponding ~53% for 'Like' and ~47% for 'NoLike' of total records. Therefore, it was thought that there was not any need for using balancing algorithms e.g., SMOTE. Female participants are higher than male participants with ~67% and ~33% of total data respectively. It is not expected that education and target values have any relationship, but its plot is also given as an additional information. Since in every trial buttons were switched to avoid lateralization of brain functions, there isn't any clear distinction between 'left' and 'right' values of 'Response' feature with respect to target values as expected.

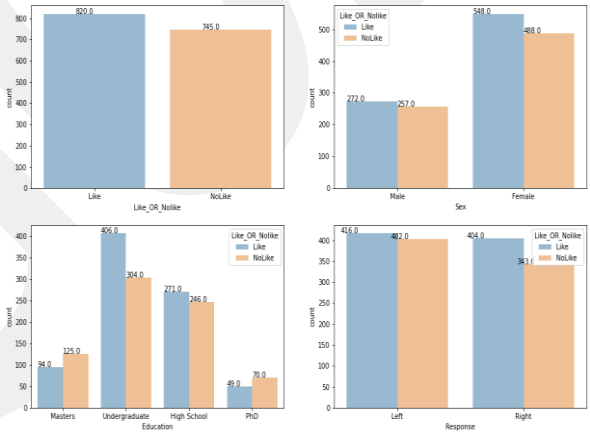
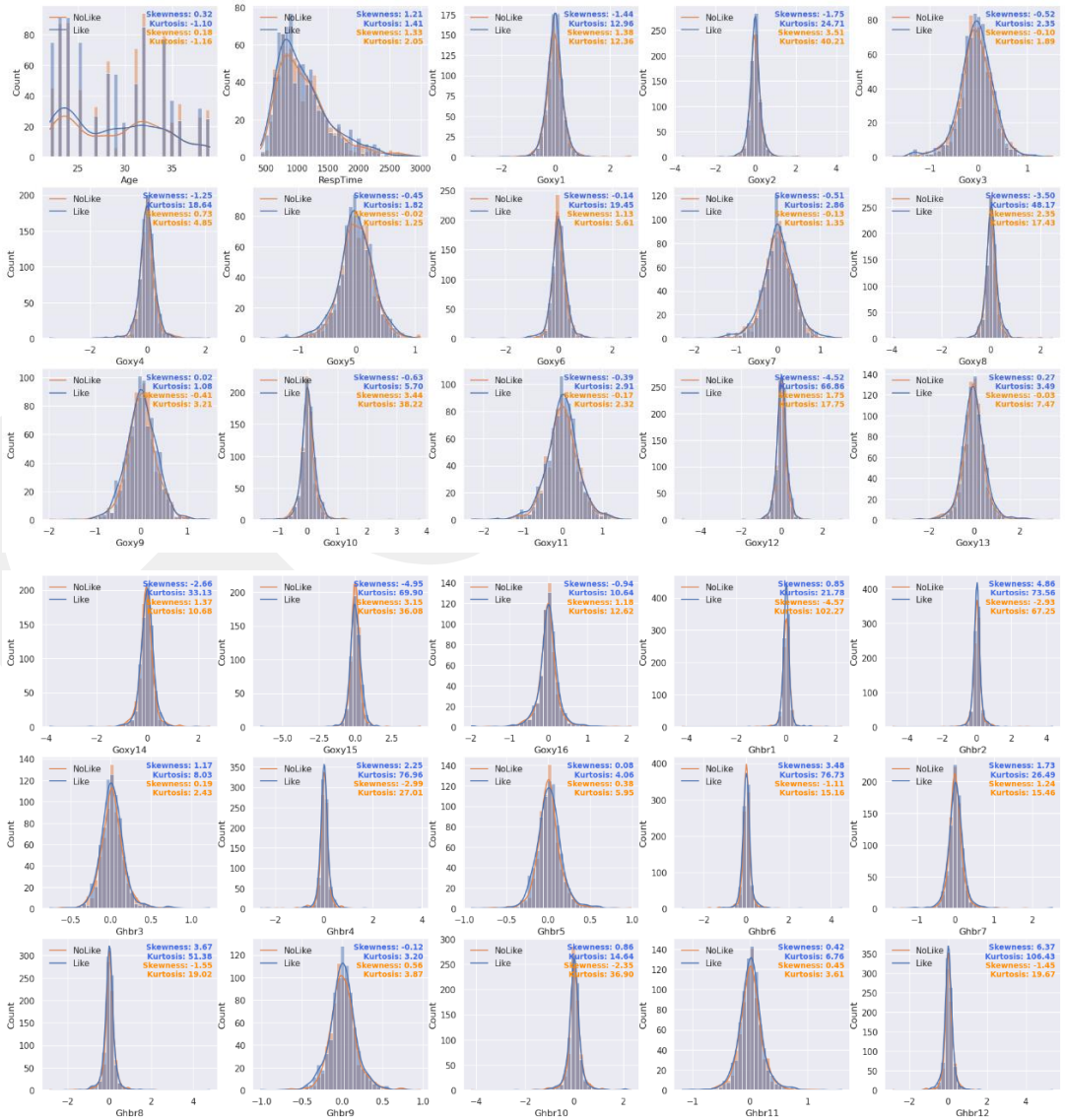
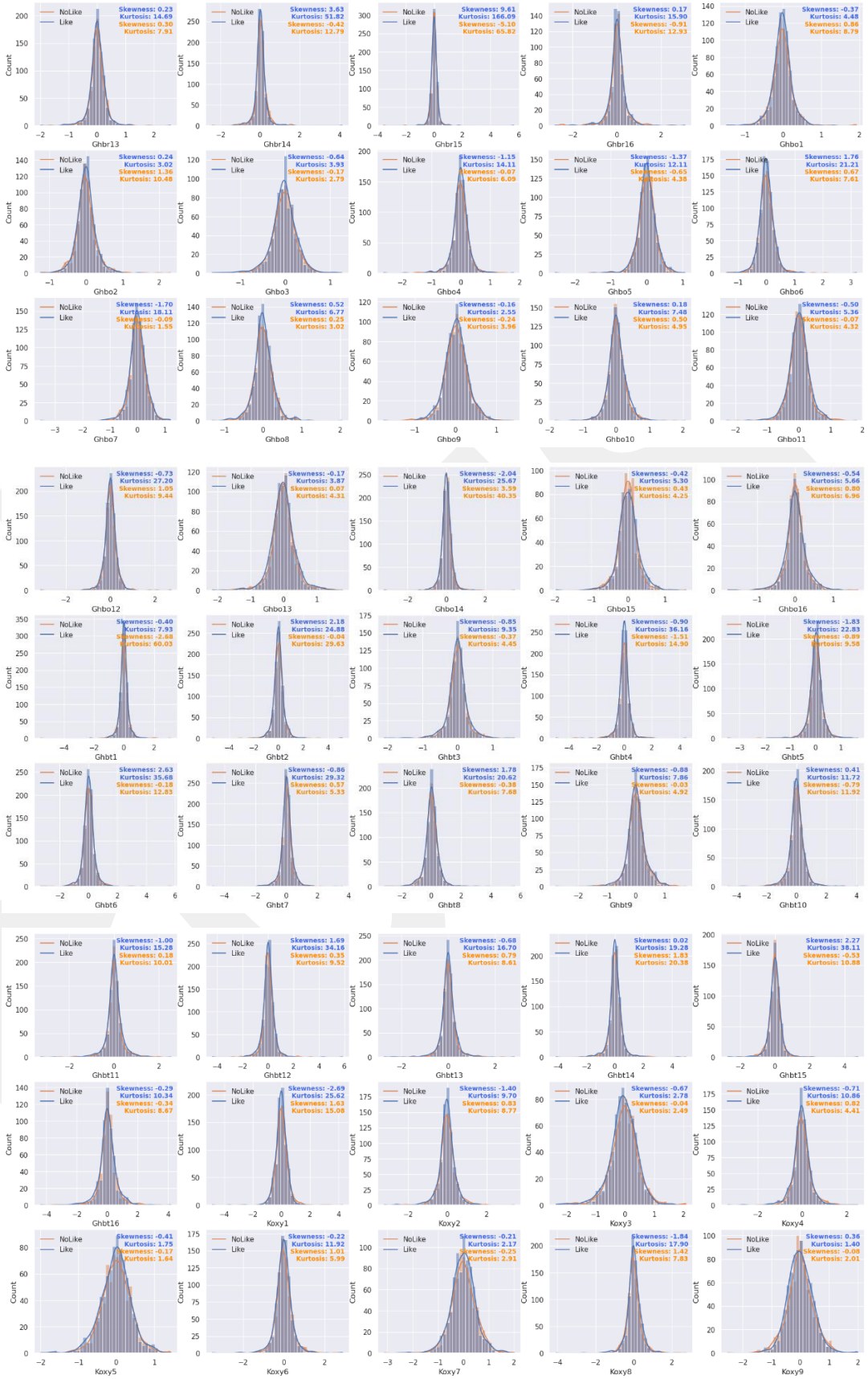


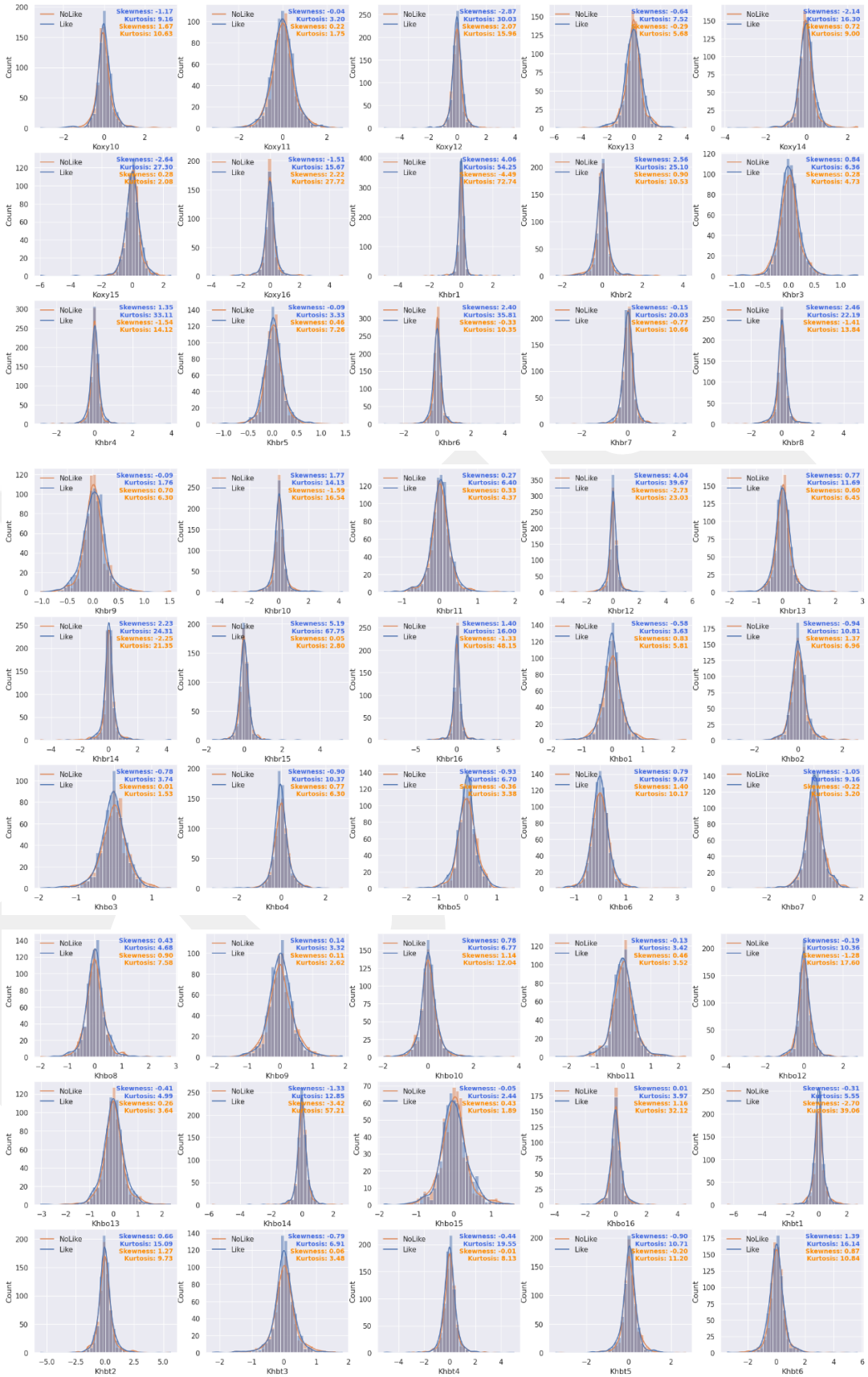
Figure 3.2: Categorical Feature Plots Against Target Values.

Numerical variables distributions are given with respect to the target values in figure 3.3. Skewness and kurtosis values also written on top right of all plots for like and dislike classes separately. First chart shows the distribution of trials across ages with respect to liking preferences, slightly different from other charts as the

participant's ages were predetermined compared to other features. There is a slight difference in distributions of like and dislike targets around the age of thirties where disliking tendency seems to dominate more, since there is a steep curve rising due to increased liking samples. Response times (RespTime) of stimuli have highly skewed distribution for both classes with around 1,3 skewness level. Most of the oxygenation measurements seem to have normal distribution and when looking at table 3.1, mean and median values are approximately equal within all these features. However, some of them have too high kurtosis (much higher than 3) meaning they have double exponential distribution rather than normal e.g., Goxy1, Goxy2, Goxy4. Furthermore, some of them are too much skewed, negatively, or positively such as Ghbr7, Ghbr4. Since ML algorithms work better with normal distributions, a transformation may be needed to develop a better model.




NoLike
Like
Count
Ghbr13
Ghbr14
Ghbr15
Ghbr16
Ghbo1
Ghbo2
Ghbo3
Ghbo4
Ghbo5
Ghbo6
Ghbo7
Ghbo8
Ghbo9
Ghbo10
Ghbo11
Ghbo12
Ghbo13
Ghbo14
Ghbo15
Ghbo16
Ghbt1
Ghbt2
Ghbt3
Ghbt4
Ghbt5
Ghbt6
Ghbt7
Ghbt8
Ghbt9
Ghbt10
Ghbt11
Ghbt12
Ghbt13
Ghbt14
Ghbt15
Ghbt16
Koxy1
Koxy2
Koxy3
Koxy4
Koxy5
Koxy6
Koxy7
Koxy8
Koxy9


NoLike
Like
Count
Koxy10 Skewness: -1.17 Kurtosis: 9.16 Skewness: 1.67 Kurtosis: 10.63
Koxy11 Skewness: -0.04 Kurtosis: 3.20 Skewness: 0.22 Kurtosis: 1.75
Koxy12 Skewness: -2.87 Kurtosis: 30.03 Skewness: 2.07 Kurtosis: 15.96
Koxy13 Skewness: -0.64 Kurtosis: 7.52 Skewness: -0.29 Kurtosis: 5.68
Koxy14 Skewness: -2.14 Kurtosis: 16.30 Skewness: 0.72 Kurtosis: 9.00
Koxy15 Skewness: -2.64 Kurtosis: 27.30 Skewness: 0.28 Kurtosis: 2.08
Koxy16 Skewness: -1.51 Kurtosis: 15.67 Skewness: 2.22 Kurtosis: 27.72
Khbr1 Skewness: 4.06 Kurtosis: 54.25 Skewness: -4.49 Kurtosis: 72.74
Khbr2 Skewness: 2.56 Kurtosis: 25.10 Skewness: 0.90 Kurtosis: 10.53
Khbr3 Skewness: 0.84 Kurtosis: 6.36 Skewness: 0.28 Kurtosis: 4.73
Khbr4 Skewness: 1.35 Kurtosis: 33.11 Skewness: -1.54 Kurtosis: 14.12
Khbr5 Skewness: -0.09 Kurtosis: 3.33 Skewness: 0.46 Kurtosis: 7.26
Khbr6 Skewness: 2.40 Kurtosis: 35.81 Skewness: -0.33 Kurtosis: 10.35
Khbr7 Skewness: -0.15 Kurtosis: 20.03 Skewness: -0.77 Kurtosis: 10.66
Khbr8 Skewness: 2.46 Kurtosis: 22.19 Skewness: -1.41 Kurtosis: 13.84
Khbr9 Skewness: -0.09 Kurtosis: 1.76 Skewness: 0.70 Kurtosis: 6.30
Khbr10 Skewness: 1.77 Kurtosis: 14.13 Skewness: -1.59 Kurtosis: 16.54
Khbr11 Skewness: 0.27 Kurtosis: 6.40 Skewness: 0.33 Kurtosis: 4.37
Khbr12 Skewness: 4.04 Kurtosis: 39.67 Skewness: -2.73 Kurtosis: 23.03
Khbr13 Skewness: 0.77 Kurtosis: 11.69 Skewness: 0.60 Kurtosis: 6.45
Khbr14 Skewness: 2.23 Kurtosis: 24.31 Skewness: -2.25 Kurtosis: 21.35
Khbr15 Skewness: 5.19 Kurtosis: 67.75 Skewness: 0.05 Kurtosis: 2.80
Khbr16 Skewness: 1.40 Kurtosis: 16.00 Skewness: -1.33 Kurtosis: 48.15
Khbo1 Skewness: -0.58 Kurtosis: 3.63 Skewness: 0.83 Kurtosis: 5.81
Khbo2 Skewness: -0.94 Kurtosis: 10.81 Skewness: 1.37 Kurtosis: 6.96
Khbo3 Skewness: -0.78 Kurtosis: 3.74 Skewness: 0.01 Kurtosis: 1.53
Khbo4 Skewness: -0.90 Kurtosis: 10.37 Skewness: 0.77 Kurtosis: 6.30
Khbo5 Skewness: -0.93 Kurtosis: 6.70 Skewness: -0.36 Kurtosis: 3.38
Khbo6 Skewness: 0.79 Kurtosis: 9.67 Skewness: 1.40 Kurtosis: 10.17
Khbo7 Skewness: -1.05 Kurtosis: 9.16 Skewness: -0.22 Kurtosis: 3.20
Khbo8 Skewness: 0.43 Kurtosis: 4.68 Skewness: 0.90 Kurtosis: 7.58
Khbo9 Skewness: 0.14 Kurtosis: 3.32 Skewness: 0.11 Kurtosis: 2.62
Khbo10 Skewness: 0.78 Kurtosis: 6.77 Skewness: 1.14 Kurtosis: 12.04
Khbo11 Skewness: -0.13 Kurtosis: 3.42 Skewness: 0.46 Kurtosis: 3.52
Khbo12 Skewness: -0.19 Kurtosis: 10.36 Skewness: -1.28 Kurtosis: 17.60
Khbo13 Skewness: -0.41 Kurtosis: 4.99 Skewness: 0.26 Kurtosis: 3.64
Khbo14 Skewness: -1.33 Kurtosis: 12.85 Skewness: -3.42 Kurtosis: 57.21
Khbo15 Skewness: -0.05 Kurtosis: 2.44 Skewness: 0.43 Kurtosis: 1.89
Khbo16 Skewness: 0.01 Kurtosis: 3.97 Skewness: 1.16 Kurtosis: 32.12
Khbt1 Skewness: -0.31 Kurtosis: 5.55 Skewness: -2.70 Kurtosis: 39.06
Khbt2 Skewness: 0.66 Kurtosis: 15.09 Skewness: 1.27 Kurtosis: 9.73
Khbt3 Skewness: -0.79 Kurtosis: 6.91 Skewness: 0.06 Kurtosis: 3.48
Khbt4 Skewness: -0.44 Kurtosis: 19.55 Skewness: -0.01 Kurtosis: 8.13
Khbt5 Skewness: -0.90 Kurtosis: 10.71 Skewness: -0.20 Kurtosis: 11.20
Khbt6 Skewness: 1.39 Kurtosis: 16.14 Skewness: 0.87 Kurtosis: 10.84

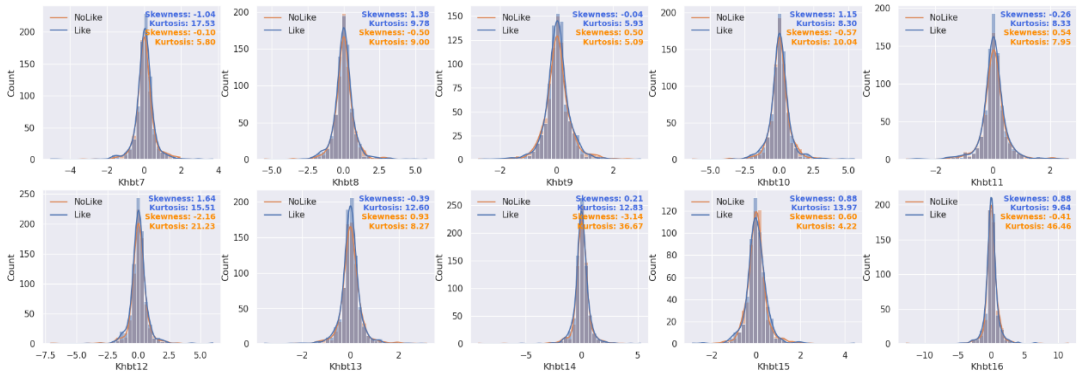


Figure 3.3: Frequency Plot of Age and Distributions of Numeric Features with target.

As there might be differences between male and female subject's hemodynamic responses for like and dislike decisions, a statistical test was performed to test the null hypothesis i.e., male and female average hemodynamic values are the same. Therefore, for each of hemodynamic features, male and female differences were tested by looking at like and dislike targets separately. For that purpose, Welch's t-test was used, and significance level was taken as 0.05 to test the null hypothesis. The choice of this type of test was made because there would be unequal variances between male and female subjects. Some resources and textbooks suggest using conditional testing i.e Student's t-test and Welch t-test depending on the power of rejection of equal variances that is done via Levene's test or others. Therefore, if there are equal variances, Student's t-test would be applied, on the other hand Welch's test would be preferred. However, that kind of conditional test does not perform better than Welch's test and even performs worse than Welch in terms of type I error if there is unequal variance between classes in a population. Furthermore, sample sizes of female and male classes are nearly 2:1 ratio, making Welch test a better option for this kind of testing [73]. Looking at 'Dislike' decision, 19 features' null hypothesis were rejected meaning they have different mean values for male and female participants. In other features, there wasn't such a rejection, emphasizing that the majority of the features have the same mean values within sex classes. On the other hand, for the 'Like' decision, 7 features' null hypothesis were rejected, less than the number for the 'Dislike' decision. In all of those features, only the 'Ghbr' feature is considered to have a different average in female and male participants for both 'Like' and 'Dislike' decisions. Since most of the features are considered to have equal means according to

these statistical tests, it might not be a good idea to include sex features into the training process of models. All statistically significant tests were painted gray on the table 3.2.

Table 3.2 : Statistical Test on Mean Difference of Sex Feature Across All Hemodynamic Measurements for Like and Dislike decisions with $\alpha = 0,05$

Feature	Dislike				Like			
	Tstats	**p-value**	**Male Sample Size**	**Female Sample Size**	**Tstats**	**p-value**	**Male Sample Size**	**Female Sample Size**
Goxy1	1,56	0,13	9(mean = 0.0,sd = 0.07)	18(mean = -0.05,sd = 0.1)	0,42	0,68	9(mean = -0.03,sd = 0.13)	17(mean = -0.05,sd = 0.08)
Goxy2	0,54	0,6	8(mean = 0.0,sd = 0.11)	18(mean = -0.02,sd = 0.07)	0,64	0,54	8(mean = -0.02,sd = 0.12)	18(mean = -0.05,sd = 0.08)
Goxy3	0,85	0,41	9(mean = -0.04,sd = 0.09)	16(mean = -0.07,sd = 0.08)	-0,56	0,59	9(mean = -0.07,sd = 0.14)	16(mean = -0.04,sd = 0.08)
Goxy4	0,48	0,64	9(mean = -0.01,sd = 0.09)	17(mean = -0.02,sd = 0.06)	0,56	0,58	9(mean = -0.01,sd = 0.09)	17(mean = -0.03,sd = 0.06)
Goxy5	2	0,06	9(mean = 0.02,sd = 0.04)	18(mean = -0.02,sd = 0.08)	0,31	0,76	9(mean = -0.0,sd = 0.1)	18(mean = -0.01,sd = 0.06)
Goxy6	2,36	0,03	9(mean = 0.02,sd = 0.05)	18(mean = -0.02,sd = 0.06)	1,32	0,21	9(mean = 0.01,sd = 0.08)	18(mean = -0.02,sd = 0.05)
Goxy7	2,33	0,03	9(mean = 0.04,sd = 0.07)	18(mean = -0.03,sd = 0.08)	0,81	0,44	9(mean = 0.02,sd = 0.13)	18(mean = -0.01,sd = 0.07)
Goxy8	1,34	0,2	9(mean = 0.05,sd = 0.09)	18(mean = 0.0,sd = 0.07)	2,2	0,05	9(mean = 0.04,sd = 0.06)	18(mean = -0.01,sd = 0.05)
Goxy9	1,61	0,12	9(mean = 0.03,sd = 0.07)	18(mean = -0.02,sd = 0.1)	1,05	0,31	9(mean = 0.04,sd = 0.1)	18(mean = -0.0,sd = 0.06)
Goxy10	0,8	0,43	9(mean = 0.03,sd = 0.06)	18(mean = 0.01,sd = 0.07)	2,05	0,06	9(mean = 0.05,sd = 0.06)	18(mean = 0.0,sd = 0.06)
Goxy11	1,76	0,09	9(mean = 0.03,sd = 0.09)	18(mean = -0.04,sd = 0.13)	0,92	0,37	9(mean = 0.02,sd = 0.12)	18(mean = -0.02,sd = 0.09)
Goxy12	1,52	0,14	9(mean = 0.0,sd = 0.05)	18(mean = -0.03,sd = 0.07)	1,81	0,09	9(mean = 0.01,sd = 0.07)	18(mean = -0.04,sd = 0.08)
Goxy13	0,52	0,61	9(mean = -0.03,sd = 0.11)	18(mean = -0.06,sd = 0.1)	0,5	0,62	9(mean = -0.01,sd = 0.15)	18(mean = -0.04,sd = 0.12)
Goxy14	0,29	0,77	9(mean = -0.03,sd = 0.05)	17(mean = -0.04,sd = 0.06)	1,6	0,13	9(mean = -0.0,sd = 0.07)	17(mean = -0.05,sd = 0.07)
Goxy15	-0,26	0,8	9(mean = -0.03,sd = 0.04)	11(mean = -0.02,sd = 0.11)	0,69	0,5	9(mean = 0.01,sd = 0.12)	11(mean = -0.03,sd = 0.12)

Goxy16	0,44	0,67	9(mean = -0.02,sd = 0.04)	12(mean = -0.03,sd = 0.08)	1,65	0,12	9(mean = 0.01,sd = 0.04)	12(mean = -0.05,sd = 0.13)
Ghbr1	-0,97	0,34	9(mean = -0.01,sd = 0.03)	16(mean = 0.01,sd = 0.04)	-0,26	0,8	9(mean = 0.01,sd = 0.05)	16(mean = 0.02,sd = 0.04)
Ghbr2	0,66	0,53	8(mean = 0.02,sd = 0.08)	18(mean = -0.0,sd = 0.04)	-0,06	0,95	8(mean = 0.03,sd = 0.05)	18(mean = 0.03,sd = 0.09)
Ghbr3	-0,42	0,69	9(mean = 0.02,sd = 0.05)	16(mean = 0.03,sd = 0.03)	0,38	0,71	9(mean = 0.03,sd = 0.06)	16(mean = 0.02,sd = 0.04)
Ghbr4	-1,08	0,3	9(mean = -0.01,sd = 0.04)	17(mean = 0.01,sd = 0.04)	-1,81	0,09	9(mean = -0.01,sd = 0.04)	17(mean = 0.02,sd = 0.04)
Ghbr5	0,24	0,81	9(mean = 0.01,sd = 0.05)	18(mean = 0.01,sd = 0.03)	0,4	0,7	9(mean = 0.01,sd = 0.06)	18(mean = 0.01,sd = 0.02)
Ghbr6	-2,6	0,02	9(mean = -0.02,sd = 0.03)	18(mean = 0.02,sd = 0.06)	-2,52	0,02	9(mean = -0.01,sd = 0.03)	18(mean = 0.03,sd = 0.05)
Ghbr7	0,51	0,62	9(mean = 0.01,sd = 0.05)	18(mean = 0.01,sd = 0.03)	-0,12	0,9	9(mean = 0.01,sd = 0.07)	18(mean = 0.01,sd = 0.02)
Ghbr8	-0,53	0,61	9(mean = -0.03,sd = 0.09)	18(mean = -0.01,sd = 0.08)	-2,02	0,06	9(mean = -0.05,sd = 0.08)	18(mean = 0.02,sd = 0.07)
Ghbr9	0,22	0,83	9(mean = 0.02,sd = 0.05)	18(mean = 0.01,sd = 0.04)	-0,99	0,34	9(mean = -0.0,sd = 0.04)	18(mean = 0.01,sd = 0.02)
Ghbr10	0,68	0,5	9(mean = 0.01,sd = 0.07)	18(mean = -0.01,sd = 0.1)	-1,18	0,25	9(mean = -0.02,sd = 0.06)	18(mean = 0.01,sd = 0.09)
Ghbr11	0,02	0,98	9(mean = 0.02,sd = 0.06)	18(mean = 0.02,sd = 0.06)	0,01	1	9(mean = 0.01,sd = 0.04)	18(mean = 0.01,sd = 0.03)
Ghbr12	-0,42	0,68	9(mean = 0.02,sd = 0.05)	18(mean = 0.03,sd = 0.07)	-2,71	0,01	9(mean = -0.02,sd = 0.03)	18(mean = 0.05,sd = 0.1)
Ghbr13	-0,11	0,92	9(mean = 0.02,sd = 0.07)	18(mean = 0.03,sd = 0.05)	-0,36	0,72	9(mean = 0.01,sd = 0.06)	18(mean = 0.02,sd = 0.04)
Ghbr14	-0,57	0,58	9(mean = 0.03,sd = 0.04)	17(mean = 0.04,sd = 0.09)	-2,81	0,01	9(mean = -0.02,sd = 0.05)	17(mean = 0.05,sd = 0.07)
Ghbr15	1,29	0,21	9(mean = 0.03,sd = 0.05)	11(mean = -0.01,sd = 0.07)	-1,13	0,28	9(mean = -0.0,sd = 0.05)	11(mean = 0.04,sd = 0.12)
Ghbr16	-0,24	0,82	9(mean = 0.04,sd = 0.04)	12(mean = 0.05,sd = 0.1)	-1,02	0,33	9(mean = 0.0,sd = 0.06)	12(mean = 0.05,sd = 0.15)
Ghbo1	1,53	0,14	9(mean = -0.0,sd = 0.05)	16(mean = -0.04,sd = 0.07)	0,13	0,9	9(mean = -0.02,sd = 0.1)	16(mean = -0.03,sd = 0.05)
Ghbo2	1,47	0,17	8(mean = 0.02,sd = 0.07)	18(mean = -0.02,sd = 0.06)	0,79	0,45	8(mean = 0.02,sd = 0.1)	18(mean = -0.01,sd = 0.05)
Ghbo3	0,93	0,37	9(mean = -0.02,sd = 0.06)	16(mean = -0.04,sd = 0.06)	-0,54	0,6	9(mean = -0.04,sd = 0.11)	16(mean = -0.02,sd = 0.05)

Ghbo4	-0,15	0,88	9(mean = -0.02,sd = 0.07)	17(mean = -0.01,sd = 0.06)	-0,35	0,73	9(mean = -0.02,sd = 0.08)	17(mean = -0.01,sd = 0.05)
Ghbo5	2,35	0,03	9(mean = 0.04,sd = 0.05)	18(mean = -0.01,sd = 0.06)	0,44	0,67	9(mean = 0.01,sd = 0.13)	18(mean = -0.01,sd = 0.05)
Ghbo6	0,24	0,81	9(mean = 0.0,sd = 0.04)	18(mean = -0.0,sd = 0.07)	-0,17	0,87	9(mean = 0.0,sd = 0.07)	18(mean = 0.01,sd = 0.05)
Ghbo7	3,19	0,01	9(mean = 0.05,sd = 0.06)	18(mean = -0.03,sd = 0.06)	0,87	0,4	9(mean = 0.03,sd = 0.11)	18(mean = -0.0,sd = 0.06)
Ghbo8	1,49	0,15	9(mean = 0.02,sd = 0.04)	18(mean = -0.0,sd = 0.06)	-0,41	0,69	9(mean = -0.0,sd = 0.11)	18(mean = 0.01,sd = 0.06)
Ghbo9	1,52	0,15	9(mean = 0.05,sd = 0.1)	18(mean = -0.01,sd = 0.08)	0,65	0,53	9(mean = 0.03,sd = 0.1)	18(mean = 0.01,sd = 0.07)
Ghbo10	1,18	0,26	9(mean = 0.04,sd = 0.09)	18(mean = -0.0,sd = 0.08)	0,47	0,64	9(mean = 0.03,sd = 0.09)	18(mean = 0.02,sd = 0.07)
Ghbo11	1,87	0,08	9(mean = 0.05,sd = 0.1)	18(mean = -0.02,sd = 0.08)	0,93	0,38	9(mean = 0.03,sd = 0.13)	18(mean = -0.01,sd = 0.07)
Ghbo12	0,9	0,38	9(mean = 0.02,sd = 0.08)	18(mean = -0.0,sd = 0.06)	-0,64	0,53	9(mean = -0.01,sd = 0.07)	18(mean = 0.01,sd = 0.06)
Ghbo13	0,61	0,55	9(mean = -0.01,sd = 0.08)	18(mean = -0.03,sd = 0.07)	0,42	0,68	9(mean = -0.01,sd = 0.14)	18(mean = -0.03,sd = 0.08)
Ghbo14	-0,37	0,72	9(mean = -0.01,sd = 0.05)	17(mean = 0.0,sd = 0.06)	-0,78	0,45	9(mean = -0.02,sd = 0.07)	17(mean = 0.0,sd = 0.05)
Ghbo15	1,02	0,32	9(mean = -0.0,sd = 0.06)	11(mean = -0.03,sd = 0.06)	-0,16	0,88	9(mean = 0.0,sd = 0.12)	11(mean = 0.01,sd = 0.03)
Ghbo16	0,2	0,85	9(mean = 0.02,sd = 0.05)	12(mean = 0.02,sd = 0.03)	0,5	0,62	9(mean = 0.01,sd = 0.08)	12(mean = -0.01,sd = 0.06)
Ghbt1	0,98	0,34	9(mean = -0.01,sd = 0.05)	16(mean = -0.03,sd = 0.06)	0	1	9(mean = -0.01,sd = 0.1)	16(mean = -0.01,sd = 0.05)
Ghbt2	1,52	0,16	8(mean = 0.04,sd = 0.11)	18(mean = -0.03,sd = 0.07)	0,61	0,55	8(mean = 0.05,sd = 0.09)	18(mean = 0.02,sd = 0.12)
Ghbt3	0,64	0,53	9(mean = 0.0,sd = 0.06)	16(mean = -0.01,sd = 0.06)	-0,35	0,73	9(mean = -0.01,sd = 0.1)	16(mean = -0.0,sd = 0.05)
Ghbt4	-0,71	0,49	9(mean = -0.03,sd = 0.08)	17(mean = -0.0,sd = 0.09)	-1,21	0,25	9(mean = -0.02,sd = 0.08)	17(mean = 0.01,sd = 0.07)
Ghbt5	1,62	0,13	9(mean = 0.05,sd = 0.09)	18(mean = -0.0,sd = 0.06)	0,47	0,65	9(mean = 0.03,sd = 0.17)	18(mean = -0.0,sd = 0.05)
Ghbt6	-1,16	0,26	9(mean = -0.02,sd = 0.05)	18(mean = 0.02,sd = 0.12)	-1,34	0,2	9(mean = -0.01,sd = 0.08)	18(mean = 0.04,sd = 0.09)
Ghbt7	3,02	0,01	9(mean = 0.07,sd = 0.08)	18(mean = -0.02,sd = 0.06)	0,67	0,52	9(mean = 0.04,sd = 0.13)	18(mean = 0.01,sd = 0.06)

Ghbt8	0,18	0,86	9(mean = -0.0,sd = 0.11)	18(mean = -0.01,sd = 0.12)	-1,19	0,26	9(mean = -0.05,sd = 0.19)	18(mean = 0.04,sd = 0.13)
Ghbt9	1,25	0,24	9(mean = 0.07,sd = 0.14)	18(mean = 0.0,sd = 0.08)	0,24	0,81	9(mean = 0.03,sd = 0.12)	18(mean = 0.02,sd = 0.08)
Ghbt10	1,03	0,32	9(mean = 0.06,sd = 0.15)	18(mean = -0.01,sd = 0.16)	-0,31	0,76	9(mean = 0.01,sd = 0.14)	18(mean = 0.03,sd = 0.15)
Ghbt11	1,44	0,18	9(mean = 0.07,sd = 0.15)	18(mean = -0.01,sd = 0.06)	0,83	0,43	9(mean = 0.04,sd = 0.15)	18(mean = -0.0,sd = 0.06)
Ghbt12	0,34	0,74	9(mean = 0.04,sd = 0.12)	18(mean = 0.03,sd = 0.11)	-1,96	0,06	9(mean = -0.03,sd = 0.09)	18(mean = 0.06,sd = 0.15)
Ghbt13	0,4	0,7	9(mean = 0.01,sd = 0.12)	18(mean = -0.0,sd = 0.05)	0,25	0,81	9(mean = -0.0,sd = 0.15)	18(mean = -0.01,sd = 0.07)
Ghbt14	-0,53	0,6	9(mean = 0.02,sd = 0.08)	17(mean = 0.04,sd = 0.14)	-2,05	0,06	9(mean = -0.04,sd = 0.1)	17(mean = 0.05,sd = 0.1)
Ghbt15	1,58	0,14	9(mean = 0.03,sd = 0.1)	11(mean = -0.03,sd = 0.07)	-0,86	0,4	9(mean = -0.0,sd = 0.14)	11(mean = 0.05,sd = 0.12)
Ghbt16	-0,08	0,93	9(mean = 0.06,sd = 0.08)	12(mean = 0.07,sd = 0.13)	-0,46	0,65	9(mean = 0.01,sd = 0.13)	12(mean = 0.04,sd = 0.19)
Koxy1	2,04	0,05	9(mean = 0.07,sd = 0.08)	16(mean = -0.0,sd = 0.1)	0,2	0,84	9(mean = -0.02,sd = 0.15)	16(mean = -0.03,sd = 0.09)
Koxy2	1,37	0,2	8(mean = 0.06,sd = 0.1)	18(mean = 0.01,sd = 0.07)	0,71	0,49	8(mean = -0.02,sd = 0.09)	18(mean = -0.04,sd = 0.06)
Koxy3	1,72	0,1	9(mean = 0.02,sd = 0.08)	16(mean = -0.05,sd = 0.12)	-0,47	0,65	9(mean = -0.07,sd = 0.18)	16(mean = -0.04,sd = 0.07)
Koxy4	1,06	0,31	9(mean = 0.04,sd = 0.09)	17(mean = 0.0,sd = 0.08)	1,07	0,31	9(mean = -0.01,sd = 0.11)	17(mean = -0.05,sd = 0.06)
Koxy5	1,52	0,15	9(mean = 0.03,sd = 0.11)	18(mean = -0.03,sd = 0.08)	0,51	0,62	9(mean = -0.02,sd = 0.1)	18(mean = -0.03,sd = 0.05)
Koxy6	1,76	0,11	9(mean = 0.06,sd = 0.11)	18(mean = -0.02,sd = 0.07)	1,42	0,18	9(mean = 0.01,sd = 0.09)	18(mean = -0.04,sd = 0.07)
Koxy7	2,4	0,03	9(mean = 0.06,sd = 0.11)	18(mean = -0.04,sd = 0.1)	1,38	0,2	9(mean = 0.03,sd = 0.11)	18(mean = -0.02,sd = 0.06)
Koxy8	0,82	0,42	9(mean = 0.03,sd = 0.1)	18(mean = -0.0,sd = 0.08)	2,27	0,04	9(mean = 0.04,sd = 0.08)	18(mean = -0.03,sd = 0.08)
Koxy9	1,71	0,11	9(mean = 0.05,sd = 0.14)	18(mean = -0.04,sd = 0.11)	1,43	0,18	9(mean = 0.04,sd = 0.12)	18(mean = -0.02,sd = 0.06)
Koxy10	0,26	0,8	9(mean = 0.03,sd = 0.12)	18(mean = 0.02,sd = 0.1)	1,57	0,13	9(mean = 0.05,sd = 0.09)	18(mean = -0.01,sd = 0.1)
Koxy11	1,85	0,09	9(mean = 0.07,sd = 0.17)	18(mean = -0.04,sd = 0.11)	0,87	0,4	9(mean = 0.01,sd = 0.11)	18(mean = -0.03,sd = 0.08)

Koxy12	0,77	0,46	9(mean = 0.02,sd = 0.13)	18(mean = -0.02,sd = 0.1)	1,29	0,21	9(mean = 0.01,sd = 0.08)	18(mean = -0.04,sd = 0.1)
Koxy13	2,1	0,05	9(mean = 0.04,sd = 0.1)	18(mean = -0.07,sd = 0.17)	0,81	0,43	9(mean = -0.01,sd = 0.12)	18(mean = -0.05,sd = 0.13)
Koxy14	0,54	0,6	9(mean = -0.01,sd = 0.1)	17(mean = -0.03,sd = 0.11)	1,7	0,11	9(mean = -0.01,sd = 0.07)	17(mean = -0.06,sd = 0.08)
Koxy15	1,28	0,22	9(mean = 0.03,sd = 0.08)	11(mean = -0.01,sd = 0.08)	1,47	0,16	9(mean = 0.03,sd = 0.09)	11(mean = -0.06,sd = 0.18)
Koxy16	1,52	0,15	9(mean = 0.04,sd = 0.1)	12(mean = -0.03,sd = 0.1)	1,04	0,31	9(mean = -0.0,sd = 0.07)	12(mean = -0.05,sd = 0.11)
Khbr1	-0,22	0,83	9(mean = -0.01,sd = 0.05)	16(mean = -0.01,sd = 0.07)	0,09	0,93	9(mean = 0.01,sd = 0.08)	16(mean = 0.01,sd = 0.07)
Khbr2	0,28	0,78	8(mean = -0.0,sd = 0.05)	18(mean = -0.01,sd = 0.05)	-0,68	0,51	8(mean = 0.02,sd = 0.04)	18(mean = 0.03,sd = 0.06)
Khbr3	-0,11	0,91	9(mean = 0.02,sd = 0.08)	16(mean = 0.03,sd = 0.05)	0,16	0,87	9(mean = 0.03,sd = 0.08)	16(mean = 0.03,sd = 0.04)
Khbr4	-0,3	0,76	9(mean = -0.01,sd = 0.05)	17(mean = -0.0,sd = 0.08)	-1,24	0,24	9(mean = -0.01,sd = 0.06)	17(mean = 0.03,sd = 0.06)
Khbr5	0,72	0,49	9(mean = 0.04,sd = 0.08)	18(mean = 0.02,sd = 0.03)	0,42	0,68	9(mean = 0.03,sd = 0.06)	18(mean = 0.02,sd = 0.04)
Khbr6	-0,61	0,55	9(mean = 0.0,sd = 0.06)	18(mean = 0.02,sd = 0.08)	-2,45	0,02	9(mean = -0.01,sd = 0.03)	18(mean = 0.03,sd = 0.06)
Khbr7	0,46	0,65	9(mean = 0.02,sd = 0.05)	18(mean = 0.01,sd = 0.03)	-1,02	0,33	9(mean = 0.0,sd = 0.06)	18(mean = 0.02,sd = 0.04)
Khbr8	1,46	0,17	9(mean = 0.04,sd = 0.08)	18(mean = -0.0,sd = 0.07)	-2,07	0,05	9(mean = -0.01,sd = 0.05)	18(mean = 0.04,sd = 0.09)
Khbr9	1,19	0,26	9(mean = 0.04,sd = 0.07)	18(mean = 0.02,sd = 0.04)	-0,91	0,37	9(mean = 0.01,sd = 0.02)	18(mean = 0.02,sd = 0.04)
Khbr10	1,99	0,06	9(mean = 0.07,sd = 0.09)	18(mean = -0.02,sd = 0.13)	0,33	0,74	9(mean = 0.03,sd = 0.07)	18(mean = 0.02,sd = 0.1)
Khbr11	-0,07	0,95	9(mean = 0.02,sd = 0.06)	18(mean = 0.02,sd = 0.04)	-0,02	0,98	9(mean = 0.02,sd = 0.05)	18(mean = 0.02,sd = 0.04)
Khbr12	1,23	0,23	9(mean = 0.04,sd = 0.08)	18(mean = -0.01,sd = 0.13)	-1,3	0,21	9(mean = 0.0,sd = 0.06)	18(mean = 0.05,sd = 0.12)
Khbr13	-1,4	0,18	9(mean = 0.01,sd = 0.07)	18(mean = 0.04,sd = 0.07)	-0,59	0,56	9(mean = 0.0,sd = 0.08)	18(mean = 0.02,sd = 0.05)
Khbr14	1,01	0,32	9(mean = 0.04,sd = 0.07)	17(mean = 0.0,sd = 0.12)	-1,51	0,14	9(mean = -0.01,sd = 0.06)	17(mean = 0.04,sd = 0.11)
Khbr15	1,6	0,13	9(mean = 0.03,sd = 0.05)	11(mean = -0.01,sd = 0.04)	-1,47	0,17	9(mean = -0.01,sd = 0.04)	11(mean = 0.05,sd = 0.14)

Khbr16	-0,65	0,53	9(mean = 0.02,sd = 0.09)	12(mean = 0.05,sd = 0.18)	-0,76	0,46	9(mean = -0.0,sd = 0.05)	12(mean = 0.04,sd = 0.17)
Khbo1	2,98	0,01	9(mean = 0.06,sd = 0.06)	16(mean = -0.01,sd = 0.06)	0,41	0,69	9(mean = -0.01,sd = 0.09)	16(mean = -0.02,sd = 0.06)
Khbo2	1,66	0,12	8(mean = 0.06,sd = 0.09)	18(mean = -0.0,sd = 0.07)	0,44	0,67	8(mean = -0.0,sd = 0.06)	18(mean = -0.01,sd = 0.06)
Khbo3	2,44	0,02	9(mean = 0.04,sd = 0.05)	16(mean = -0.02,sd = 0.08)	-0,5	0,63	9(mean = -0.03,sd = 0.13)	16(mean = -0.01,sd = 0.08)
Khbo4	0,95	0,35	9(mean = 0.03,sd = 0.07)	17(mean = -0.0,sd = 0.09)	0,35	0,73	9(mean = -0.01,sd = 0.07)	17(mean = -0.02,sd = 0.05)
Khbo5	2,72	0,02	9(mean = 0.07,sd = 0.08)	18(mean = -0.01,sd = 0.06)	0,54	0,6	9(mean = 0.01,sd = 0.15)	18(mean = -0.01,sd = 0.05)
Khbo6	1,78	0,09	9(mean = 0.06,sd = 0.07)	18(mean = 0.0,sd = 0.08)	0,11	0,92	9(mean = -0.0,sd = 0.09)	18(mean = -0.0,sd = 0.06)
Khbo7	3,3	0,01	9(mean = 0.09,sd = 0.09)	18(mean = -0.03,sd = 0.07)	0,84	0,42	9(mean = 0.03,sd = 0.11)	18(mean = -0.0,sd = 0.05)
Khbo8	2,41	0,03	9(mean = 0.07,sd = 0.08)	18(mean = -0.01,sd = 0.06)	0,64	0,53	9(mean = 0.03,sd = 0.08)	18(mean = 0.01,sd = 0.05)
Khbo9	1,91	0,08	9(mean = 0.1,sd = 0.17)	18(mean = -0.02,sd = 0.09)	1,3	0,22	9(mean = 0.05,sd = 0.11)	18(mean = -0.0,sd = 0.06)
Khbo10	1,57	0,14	9(mean = 0.1,sd = 0.17)	18(mean = -0.0,sd = 0.11)	1,76	0,11	9(mean = 0.08,sd = 0.12)	18(mean = 0.01,sd = 0.04)
Khbo11	1,93	0,08	9(mean = 0.09,sd = 0.17)	18(mean = -0.02,sd = 0.08)	0,78	0,46	9(mean = 0.03,sd = 0.13)	18(mean = -0.01,sd = 0.06)
Khbo12	1,64	0,13	9(mean = 0.06,sd = 0.15)	18(mean = -0.02,sd = 0.08)	0	1	9(mean = 0.01,sd = 0.08)	18(mean = 0.01,sd = 0.06)
Khbo13	1,62	0,12	9(mean = 0.05,sd = 0.11)	18(mean = -0.03,sd = 0.12)	0,53	0,61	9(mean = -0.01,sd = 0.12)	18(mean = -0.03,sd = 0.09)
Khbo14	1,54	0,15	9(mean = 0.03,sd = 0.11)	17(mean = -0.03,sd = 0.08)	0,08	0,94	9(mean = -0.02,sd = 0.08)	17(mean = -0.02,sd = 0.05)
Khbo15	2,18	0,05	9(mean = 0.06,sd = 0.09)	11(mean = -0.02,sd = 0.07)	0,73	0,48	9(mean = 0.01,sd = 0.09)	11(mean = -0.01,sd = 0.06)
Khbo16	0,67	0,51	9(mean = 0.05,sd = 0.09)	12(mean = 0.02,sd = 0.1)	0,04	0,97	9(mean = -0.0,sd = 0.06)	12(mean = -0.01,sd = 0.08)
Khbt1	2	0,06	9(mean = 0.05,sd = 0.07)	16(mean = -0.02,sd = 0.08)	0,46	0,65	9(mean = 0.01,sd = 0.08)	16(mean = -0.01,sd = 0.1)
Khbt2	1,46	0,17	8(mean = 0.06,sd = 0.11)	18(mean = -0.01,sd = 0.1)	-0,11	0,91	8(mean = 0.02,sd = 0.05)	18(mean = 0.02,sd = 0.1)
Khbt3	1,6	0,14	9(mean = 0.07,sd = 0.1)	16(mean = 0.01,sd = 0.06)	-0,4	0,7	9(mean = -0.0,sd = 0.13)	16(mean = 0.02,sd = 0.1)

Khbt4	0,52	0,6	9(mean = 0.02,sd = 0.07)	17(mean = -0.0,sd = 0.15)	-0,62	0,55	9(mean = -0.02,sd = 0.09)	17(mean = 0.0,sd = 0.08)
Khbt5	2,46	0,03	9(mean = 0.12,sd = 0.11)	18(mean = 0.01,sd = 0.06)	0,53	0,61	9(mean = 0.04,sd = 0.2)	18(mean = 0.01,sd = 0.07)
Khbt6	0,87	0,39	9(mean = 0.06,sd = 0.07)	18(mean = 0.02,sd = 0.15)	-0,95	0,36	9(mean = -0.01,sd = 0.11)	18(mean = 0.03,sd = 0.1)
Khbt7	3,57	0	9(mean = 0.11,sd = 0.09)	18(mean = -0.01,sd = 0.06)	0,21	0,84	9(mean = 0.03,sd = 0.14)	18(mean = 0.02,sd = 0.06)
Khbt8	2,42	0,03	9(mean = 0.11,sd = 0.13)	18(mean = -0.01,sd = 0.11)	-0,76	0,46	9(mean = 0.02,sd = 0.11)	18(mean = 0.05,sd = 0.11)
Khbt9	1,89	0,09	9(mean = 0.14,sd = 0.22)	18(mean = -0.0,sd = 0.07)	1,07	0,3	9(mean = 0.07,sd = 0.11)	18(mean = 0.02,sd = 0.08)
Khbt10	1,9	0,08	9(mean = 0.16,sd = 0.25)	18(mean = -0.02,sd = 0.22)	1,29	0,22	9(mean = 0.11,sd = 0.17)	18(mean = 0.03,sd = 0.12)
Khbt11	1,77	0,11	9(mean = 0.12,sd = 0.19)	18(mean = 0.0,sd = 0.05)	0,62	0,55	9(mean = 0.05,sd = 0.16)	18(mean = 0.01,sd = 0.06)
Khbt12	1,7	0,11	9(mean = 0.11,sd = 0.2)	18(mean = -0.03,sd = 0.19)	-0,83	0,41	9(mean = 0.01,sd = 0.12)	18(mean = 0.06,sd = 0.17)
Khbt13	0,7	0,5	9(mean = 0.05,sd = 0.15)	18(mean = 0.02,sd = 0.08)	0,09	0,93	9(mean = -0.01,sd = 0.16)	18(mean = -0.01,sd = 0.07)
Khbt14	1,51	0,15	9(mean = 0.07,sd = 0.15)	17(mean = -0.03,sd = 0.17)	-0,87	0,39	9(mean = -0.02,sd = 0.12)	17(mean = 0.02,sd = 0.15)
Khbt15	2,36	0,03	9(mean = 0.09,sd = 0.13)	11(mean = -0.03,sd = 0.08)	-0,81	0,43	9(mean = 0.0,sd = 0.11)	11(mean = 0.04,sd = 0.12)
Khbt16	-0,11	0,91	9(mean = 0.07,sd = 0.15)	12(mean = 0.08,sd = 0.27)	-0,52	0,61	9(mean = -0.01,sd = 0.08)	12(mean = 0.03,sd = 0.25)

A dependent t-test was also made for comparing average response times of 27 participants for like (mean = 1121 ms, sd = 223 ms) and dislike (mean = 1117, sd = 227) decisions, which was not statistically significant to reject the null hypothesis of having same average response times, t (26) = 0.07, $p > 0.05$. These tests elicit that deploying response time into models might not be a good idea to predict the liking preference of individuals.

Correlation matrix was created for numeric and categorical features in the figure 3.4 below. It is based on several correlation coefficients due to different types of variables. Pearson correlation coefficient was used for numerical vs categorical and numerical vs numerical pairs, which measures linear correlation between pairwise sets

of these features. Result of this correlation is always between -1 and 1, representing negative and positive correlations respectively. It requires data to be normally distributed which is the case for features in this data. Second, Cramer V correlation coefficient was performed on liking and sex pair due to both being dichotomous variables. It came out to be 0.01 which indicates there is no relation between them. Finally, Spearman's correlation coefficient was used to calculate the correlation between Education vs other features since it is an ordinal feature, whose correlation values can be also interpreted from the figure 3.4. Some oxygenation features are highly correlated which may catch one's eye when looking at correlation heatmap. It can be seen from patterns of light-yellow colors and darker stacks continuing diagonally. This might indicate that there could be multicollinearity between those features, and hence it could deteriorate the performance of machine learning models. Fortunately, feature extraction methods will be implemented that may help eliminate this multicollinearity in those features. On the other hand, it would be beneficial to have highly correlated features while imputing missing values of them according to those correlations.

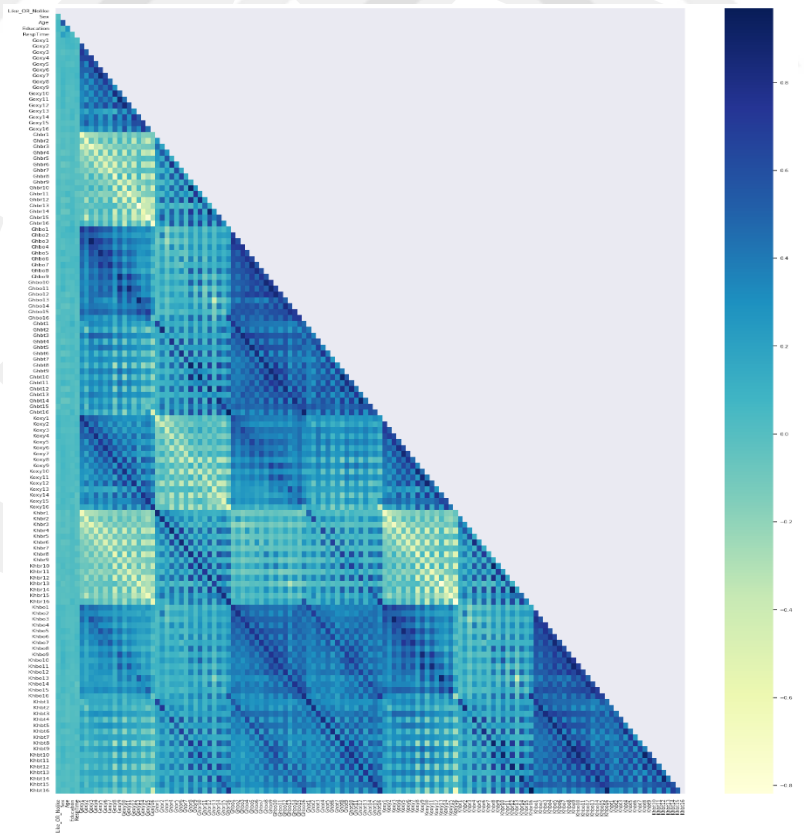


Figure 3.4: Correlation Heatmap of Features.

Top 15 correlated variables with respect to target is given in the figure 3.5 below. There isn't any strong correlation between dependent target variables and independent predictor variables. However, ensembles of predictors might have strong predictive power on this dependent variable which will be seen in the modeling phase. In addition, some features have high correlation between each other that would be beneficial while imputing missing values according to those correlations. On the other hand, highly correlated values can lead to multicollinearity which causes the model to be statistically ineffective. It can also cause high overfitting in models. Therefore, the best model can be defined as in which the predictor variables are highly correlated with dependent variables and at minimum within each other. Based on this, it is thought that feature extraction methods could eliminate this multicollinearity between predictor variables and improve the performance of algorithms that will be used.

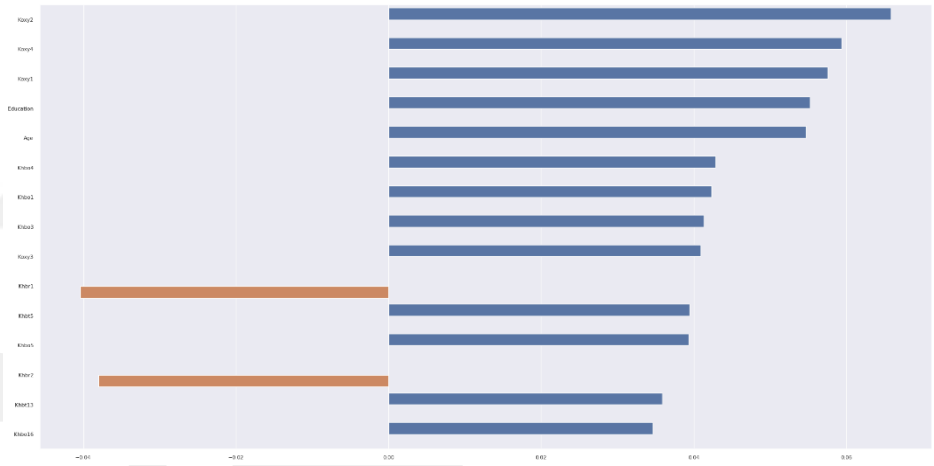


Figure 3.5: Top 15 Correlated Independent Variables for Target.

3.3. Data Preparation

In this section, the results and analysis of data preparation explained in the methodology chapter will be given.

3.3.1. Data Cleaning

At the beginning, 1.680 records existed which consisted of 28 participants' hemodynamic measurements. In some trials, several participants did not indicate any preference on them, in addition one participant did not indicate any preference at all, meaning all results of him were removed. Therefore, the record count was reduced

from 1.680 to 1.546. Finally, in one trial, the wrong button was pushed and this result was also excluded from the dataset, leaving 1.545 records available.

3.3.2. Missing value analysis

In table 3.3, missing value records and percentages for hemodynamic features were given. Especially, measurements taken from optodes 15 and 16 have the highest missing values compared to other optodes. It was seen that these missing values belong to mostly female participants, hence it was thought that their hair might have blocked the measurement process in these locations. On average, 6.9% of values were missing for all hemodynamic measurements, and table 3.3 shows features that have missing values above this threshold.

Table 3.3 : Missing Values of Hemodynamic Measurements

Feature	Missing Values	% of Values
Khbt16	503	32,1
Khbr16	503	32,1
Koxy16	503	32,1
Khbo16	503	32,1
Ghbo16	501	32
Ghbt16	501	32
Goxy16	501	32
Ghbr16	501	32
Koxy15	455	29,1
Khbt15	455	29,1
Khbo15	455	29,1
Khbr15	455	29,1
Ghbr15	449	28,7
Goxy15	449	28,7
Ghbo15	449	28,7
Ghbt15	449	28,7
Ghbo3	131	8,4
Goxy3	131	8,4
Ghbt3	131	8,4
Khbo3	131	8,4
Koxy3	131	8,4
Ghbr3	131	8,4
Khbt3	131	8,4
Khbr3	131	8,4
Khbo1	124	7,9
Khbt1	124	7,9

Khbr1	124	7,9
Koxy1	124	7,9
Ghbt1	121	7,7
Ghbo1	121	7,7
Ghbr1	121	7,7
Goxy1	115	7,3

Moreover, trial based missing values were also investigated. Any record had more than 25% of its values missing were removed from the data set as they might cause negative effects on ML algorithms, or on the contrary may have caused biased predictions because of imputations that would be put in their place. Imputed values could be repetitive and that may cause this bias effect. There were 58 records that had passed that missing value threshold and they were removed from the data set.

3.3.3. Outlier detection and removal

Outlier detection and removal had been done in an iterative approach. First, as mentioned in the methodology chapter, outlier values were masked as Nan to be later imputed. Since these imputations could cause biased results, a second approach was developed and chosen as an outlier removal method, which was capping outlier values to IQR boundaries. This way, both bias effects would be avoided and also important information that indicates activation of certain parts of the prefrontal cortex would be retained.

3.3.4. Imputation

As stated in the methodology chapter, various imputation methods were utilized i.e. mean, median, MICE, iterative imputation and KNN imputer. MICE and iterative imputation are in fact the same methodologies, but while MICE imputes values by using all other features depending on their correlation with the imputed feature, iterative imputation fills values by the closest three neighbor features in terms of correlation.

Original/non-imputed data has very high kurtosis and significant skewness for nearly every feature as it was stated in the data understanding section. Outlier removal reduced those properties by a significant amount. However, the choice of imputation was critical to avoid having a distribution other than or not close to gaussian.

Furthermore, interpretation of the missing values was important to represent the real properties of a feature which would significantly affect the predictive performance of the models.

In figure 3.6, distributions of all imputation methods were plotted for the Koxy16 feature which was one of the features that had the most missing values. Therefore, comparison was made based on this single feature for simplification of missing value analysis and also for the choice of imputation method. All of these data were standardized before plotting to make a better choice as the chosen dataset will be put into standardization before fitting it into the models. Skewness close to zero and kurtosis close to 3 was targeted as these values represent normal distribution. Original(baseline) dataset had not any skewness (0.06) but it had excess kurtosis (7.22) which represents a leptokurtic distribution. Dataset with mean and median imputations, which are on top of each other in figure 3.6, seem close to the original dataset, with a slight decrease in kurtosis, which were 6.88 and 6.89 respectively. Remaining methods seem close to each other, where neighbor imputation comes forward with kurtosis of 2.82. KNN imputation follows it with kurtosis of 3.25 and then iterative imputation with 3.32, and finally MICE with 4.38. Only iterative imputation has a slight skewness while others have almost zero skewness.

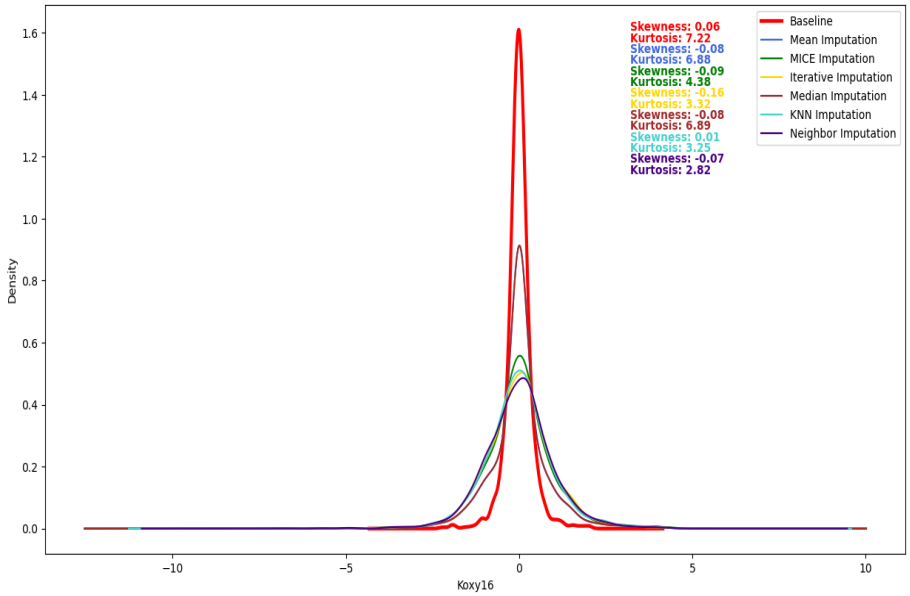


Figure 3.6: Various Imputed Distributions of Koxy16.

Since the distribution of the dataset belonging to neighbor imputation seems closer to normal distribution, it was selected as the choice of imputation method. Its closer competitors were KNN and iterative imputations, but they were not considered to be the preferred choice compared to neighbor imputation as for KNN, while it imputes values by looking at two neighbors same as neighbor imputation, it considers the neighborhood of records rather than features in the dataset. On the other hand, neighbor imputation depends on the neighborhood of optodes attached to fNIRS devices. For instance, Koxy1's two neighbors are Koxy2 and Koxy3, hence if the Koxy1 value of a trial was missing, it would be imputed with the mean of those two neighbors. Therefore, the neighborhood of features within a trial might represent those missing values in terms of closeness to their real measurements that we could not measure. For iterative imputation, it is based on a divide and conquer approach where several copies of a dataset are first created, then for each copy, each feature that has missing values are filled depending on the top three highly correlated features with that one. This is also a good approach, but it relies on the "neighborhood" of correlation, hence, since it was thought that imputation by neighborhood of optodes might represent the actual values of missing values better, hence neighbor imputation was preferred over iterative imputation.

3.3.5. Data clustering based on fNIRS features

Since some participants' hemodynamic responses might resemble each other, K-Means clustering was applied to the dataset with given participant numbers. In figure 3.7, silhouette scores of K-Means clustering with k ranging 2 to 19 are given. The oscillation and the peak on cluster number of 4 suggest that there should be that many groups where the participants are put into. However, it might be elicited that the silhouette score of 0.35 indicates that the clustering achieved would not be a good one. Nevertheless, since it was known that there might be a similarity of hemodynamic measurements between participants, cluster numbers were added to the dataset as a new feature.

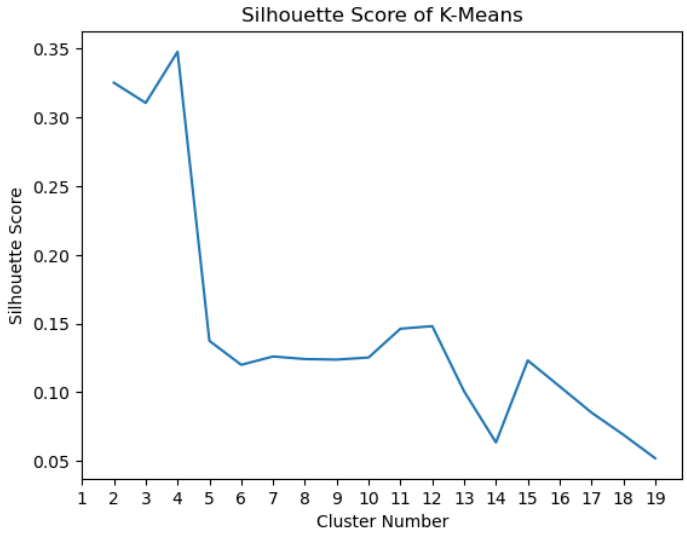


Figure 3.7: Silhouette Score Plot of K-Means Clustering with K from 1 to 19

3.3.6. Frontal-alpha-asymmetry Index

In order to form a FAA index, the left and right hemisphere's relative change in oxygenation signals have been calculated and formed as a new feature, raising the feature amount to 129.

3.3.7. One-hot encoding

Cluster groups were one-hot encoded and four new features were formed from that information. This was done due to some algorithms requiring categorical variables that should be encoded into integer values. With this being done, the feature number had risen to 133. Those four dummy features were named 'cl_1', 'cl_2', 'cl_3', 'cl_4'.

3.3.8. Standardization of the numeric data

StandardScaler of scikit-learn library was used to standardize numeric values which were hemodynamic responses. Therefore, with this scaling, the skewness and especially excess kurtosis were highly eliminated which can be seen in figure 3.3. Also looking at figure 3.6, with referencing 'Koxy16' feature, those elimination can be clearly seen; that was also the case for other features, too.

3.4. Modeling and Evaluation

In this section, classification models were built to predict the liking preference of individuals. For all models, leave-one-out cross-validation was applied to ensure that the results have reliability and test all participants' results. F1-score, precision and recall scores were reported for every model. Firstly, the main model's results were given to find the best performing ML algorithm. Then, feature extraction models' results were shown utilizing the best algorithm selected in the main model. Finally, the result of a model with feature selection was given to see the effect of feature selection on classification performance.

3.4.1. Main model results

For the main model, five machine learning algorithms were utilized to find the best among them for prediction of liking. In table 3.4, average F1, precision and recall scores are given for all of these algorithms with standard deviations in parentheses. Each subjects' trials were tested by leave-one out cross validation and the resulting scores were averaged which can be found in the table 3.4 below. Thus, all algorithms' scores consist of 27 individual's scores. Highest scores for F1, precision and recall are highlighted.

Table 3.4 : Average Scores of Algorithms

Algorithm	F1-Score	Precision	Recall
k-Nearest Neighbor	0.55 (0.1)	**0.55 (0.16)**	0.59 (0.11)
Support Vector Machines	0.65 (0.04)	0.51 (0.03)	**0.92 (0.09)**
Random Forest	0.62 (0.09)	0.52 (0.14)	0.82 (0.11)
XGBoost	0.6 (0.08)	0.53 (0.14)	0.74 (0.11)
LightGBM	**0.66 (0.08)**	0.53 (0.13)	**0.92 (0.08)**

For KNN, the highest F1-score achieved is 75% and the lowest is 37% with 55% as average score. It has the lowest average F1-score compared to other algorithms. Number of neighbor parameters was searched from 2 to 20 as stated in methodology to optimize this algorithm, and n = 17 was found to be most frequently selected as the

best number of neighbors. In figure 3.8, histogram of all participant's F1-scores were given to view the distribution of these scores.

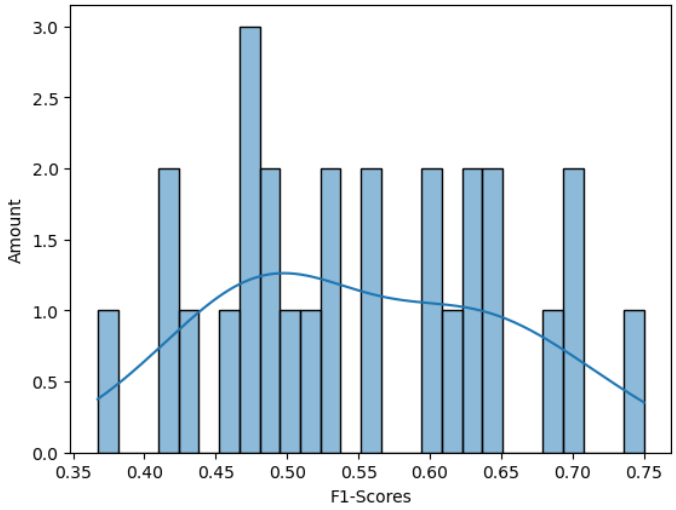


Figure 3.8: All Participants' F1-Scores for KNN algorithm.

Fourth highest average F1-score was obtained by RFC, with 62%. Its lowest and highest scores are 47% and 79% respectively as can bee seen in the histogram in figure 3.9. Best found parameters for the optimization of this algorithm are, criterion = 'entropy', max_depth = 5 and n_estimators = 500.

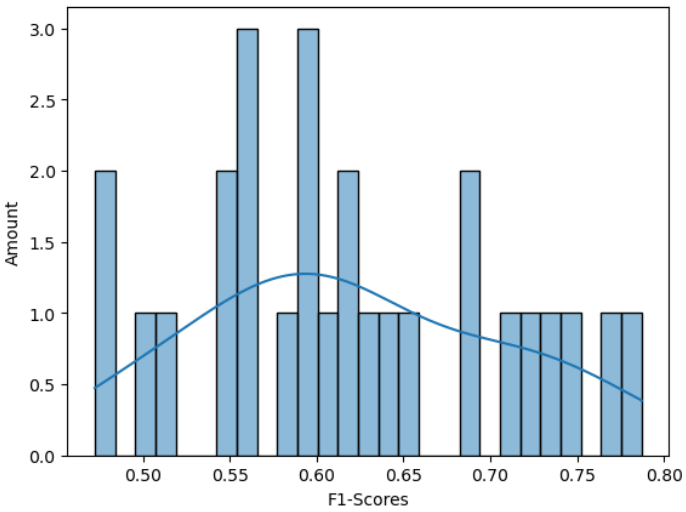


Figure 3.9: All Participants' F1-Scores for RFC algorithm.

Lowest and highest scores obtained with XGB algorithm are 47% and 80% respectively, and average score can be seen as 63%. Hyperparameter tuning for

optimization of this algorithm was mentioned in the methodology part. Best and most frequent parameters were found as; n_estimators(number of trees) = 100, min_child_weight = 4, max_depth = 3, learning_rate = 0.03, gamma = 10. Distribution of all participants' F1-scores is given in figure 3.10.

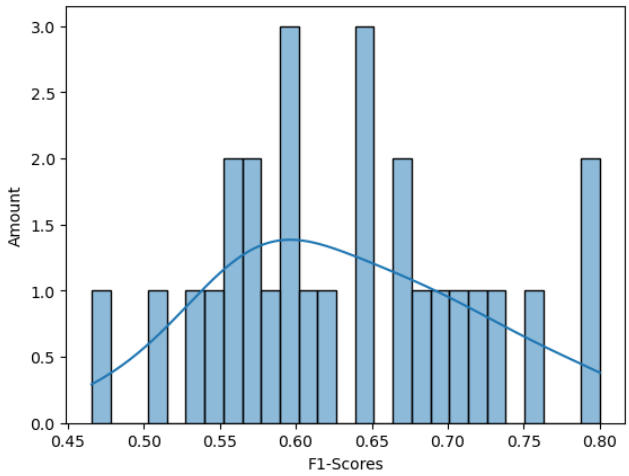


Figure 3.10: All Participants' F1-Scores for XGB algorithm.

SVM achieved the second best average F1-score, with 65%, and its lowest and highest scores are 50% and 68% respectively (figure 3.11). Lowest score obtained from a participant seems an outlier as it sits alone in that 50% range and the next score comes after it stands at 60%. Among all validation results, most frequent parameters were found as, C = 1 and gamma = 1.

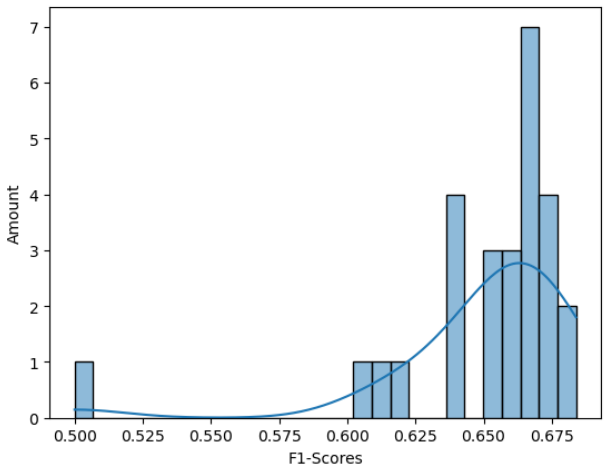


Figure 3.11: All Participants' F1-Scores for SVM algorithm.

LGBM has the best F1-score among all algorithms. However, SVM's score is really close to LGBM's, thus from these results alone, it cannot be said that either algorithm is superior to the other. Therefore, the results of the permutation and Wilcoxon test will be decisive to better compare these algorithms' scores. LGB's score distributions are also given in figure 3.12 below. Its highest score is 82% whereas its lowest is 46% within all participant scores. Best parameters that most frequently appear in cross-validation results can be given as path_smooth = 4, min_child_samples = 28, max_depth = 3, max_bin = 50, learning_rate = 0.03, and num_iterations(number of trees) = 100.

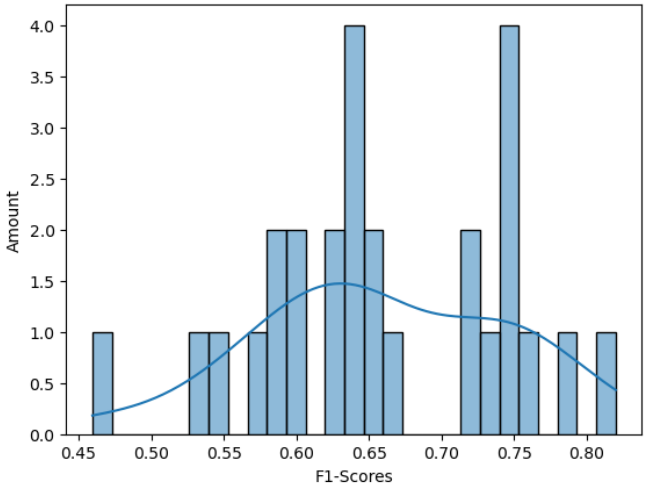


Figure 3.12: All Participants' F1-Scores for LGBM algorithm

Boxplots of all results of algorithms are given in figure 3.13 to better visualize the performances of them. Green triangles are mean score, and vertical lines represent median score. SVM's variance seems the lowest and its scores look tight except the outlier at 50%. KNN seems inferior compared to others when looking at these range of scores and the remaining algorithms seem close to each other meaning statistical tests will be more informative for a decision of superiority among them.

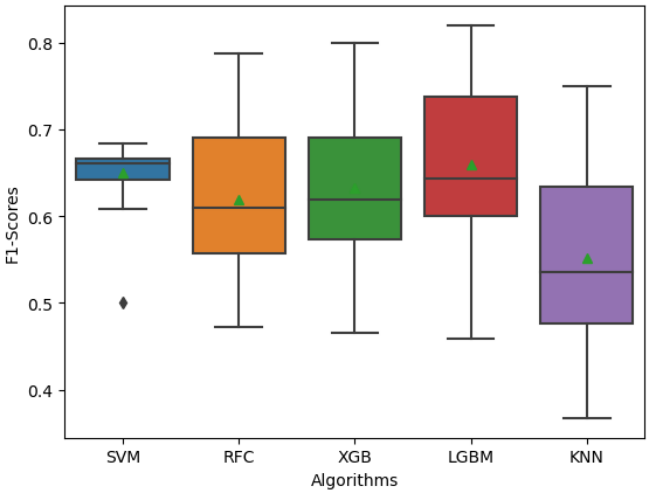


Figure 3.13: Boxplot of All Algorithms' Scores.

Wilcoxon signed-rank tests were applied to every pair of algorithms to test the statistical significance of the difference of F1-scores of all participants if there was any superiority between each pair of algorithms. The cut-off was chosen as '0.05', hence any p-value lower than that would indicate that the higher scoring algorithm's superiority was statistically significant. All pairs' p-values are given in the table 3.5 below. For p-values < 0.05, results have been made bold. It can be clearly seen that KNN having the lowest average score (55%) has been statistically confirmed ($p < .05$). XGB (63%) has a higher average score than RFC (62%), but statistical tests could not find any difference between each algorithm's participant scores ($p = .055$). Although SVM's average F1-score is better than XGB's, their participant scores are statistically indifferent ($p = .31$). Moreover, the same thing can also be said for the SVM-RFC pair, even though SVM (65%) has a higher average score than RFC (62%). On the other hand, LGBM seems the superior one compared to others ($p < .05$) except SVM ($p = 0.56$).

Table 3.5 : Wilcoxon Signed-Rank Test Results

p-values	KNN	SVM	RFC	XGB	LGBM
KNN	-	-	-	-	-
SVM	**0.000**	-	-	-	-
RFC	**0.004**	0.088	-	-	-
XGB	**0.002**	0.313	0.055	-	-
LGB	**0.000**	0.564	**0.000**	**0.004**	-

As a final evaluation for these five algorithms, permutation tests were performed. Dashed red line represents the original dataset's score on the distribution of permuted scores. To say the original score is better than by chance alone, the cut-off value has been taken as '0.05'. From figure 3.14 to 3.18, all five algorithms' permutation plots are present. KNN, RFC and XGB have achieved indifferent results by chance alone ($p > .05$). Therefore, it could be said that there is no dependency between trained features and predicted classes i.e., there is no real connection between them for these models. On the other hand, SVM and LGBM have statistically significant results indicating a good connection between respective features and labels ($p < .05$). In addition, their all-permuted classification results were lower than their actual results which also states the robustness of these algorithms in predicting the preference of liking using fNIRS data.

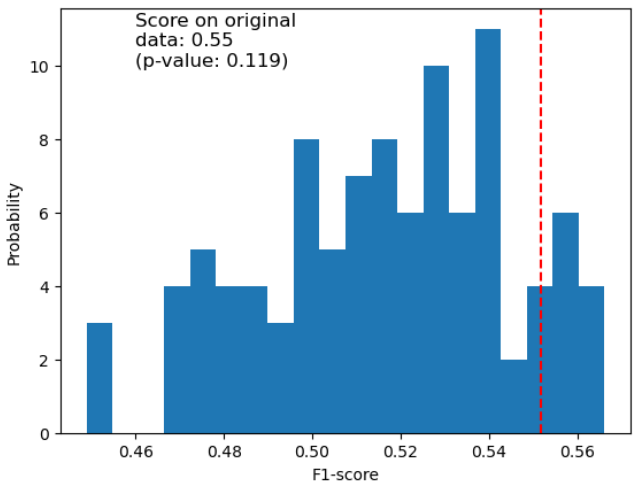


Figure 3.14: KNN's Permutation Test Results.

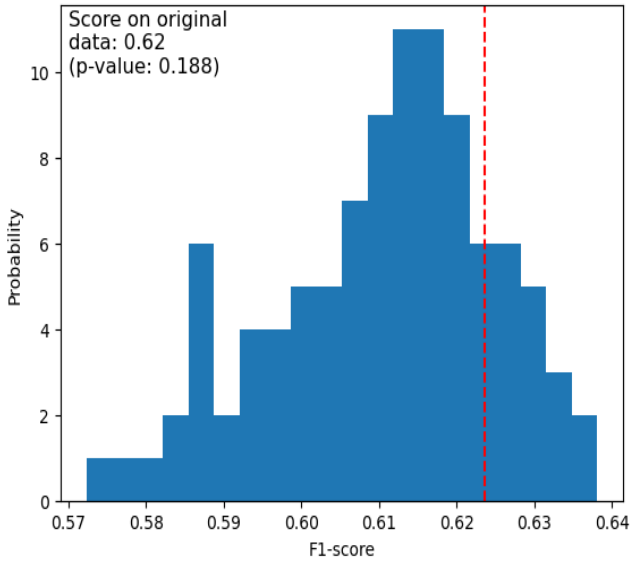


Figure 3.15: RFC's Permutation Test Results.

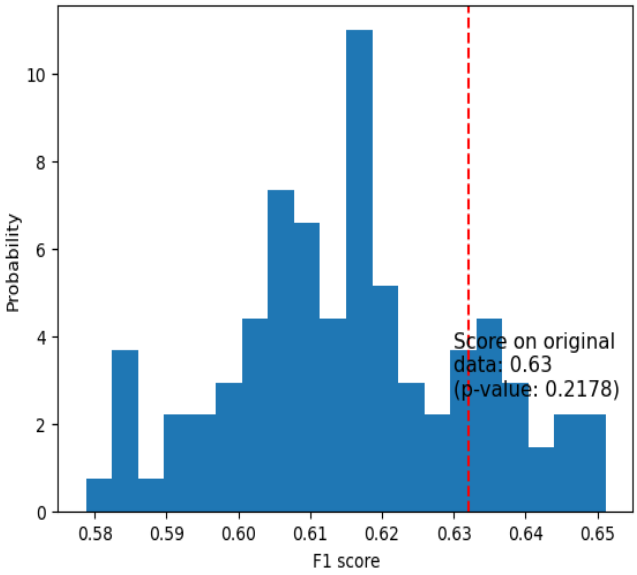


Figure 3.16: XGB's Permutation Test Results.

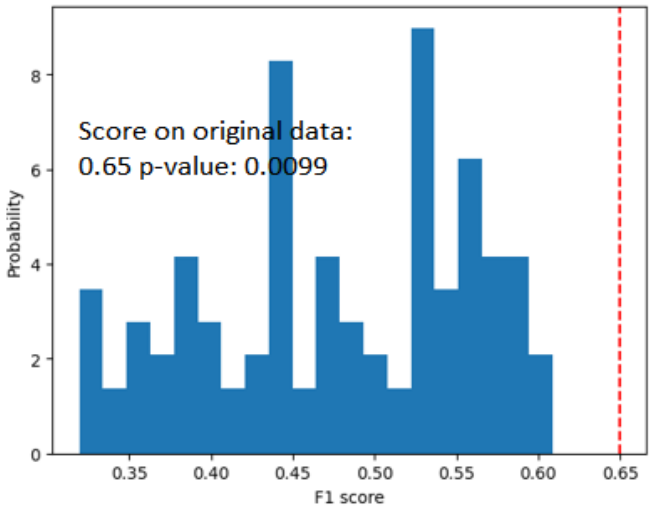


Figure 3.17: SVM's Permutation Test Results.

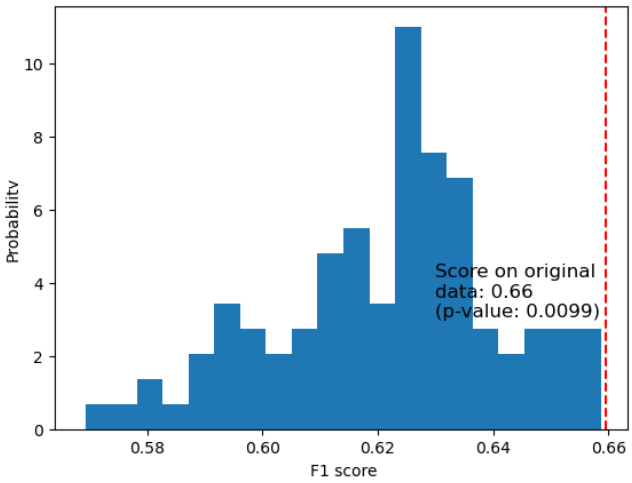


Figure 3.18: LBGM's Permutation Test Results.

LGBM had the highest average F1-score compared to other algorithms with SVM following it closely by 1% difference. Wilcoxon signed-rank test results also show that LGBM has significantly higher scores than every other algorithm except SVM, while its closest competitor SVM has significantly higher results only from KNN. Their permutation test results do not add any value to further comparison of them. However, there are enough results which indicate that LGBM might be a better choice for this kind of experiment.

3.4.2. Feature extraction models

For this section, the LGBM model has been selected as the best algorithm to apply feature extraction methodologies. As a secondary goal for this study, it was aimed to improve prediction scores of models by applying PCA, Isomap and t-SNE approaches which help reduce feature dimensionality and eliminate multicollinearity and compare these approaches' performances between them. Same result format has been followed to compare the implemented models as in the main model section.

In table 3.6., average F1, precision and recall scores of cross-validation results with their respective standard deviations have been given for the main model which is the base model selected in the previous main model section, and three models with feature extraction methods. Highest scores were highlighted in bold.

Table 3.6 : Average Scores of Feature Extraction and Main Models

Models	F1-Score	Precision	Recall
Main Model	**0.66 (0.08)**	**0.53 (0.13)**	**0.92 (0.08)**
PCA	0.62 (0.09)	0.52 (0.14)	0.82 (0.09)
ISOMAP	0.59 (0.09)	0.52 (0.15)	0.72 (0.1)
t-SNE	0.61 (0.09)	0.52 (0.14)	0.78 (0.11)

For LGBM models with Isomap applied, the highest F1-score achieved is 75% and the lowest is 43% with 59% as average score. It has the lowest average F1-score compared to the other three models. Best parameter couples that appear most frequently in cross-validation results are path_smooth = 4, min_child_samples = 23, max_depth = 3, max_bin = 50, learning_rate = 0.05, and num_iterations(number of trees) = 100. In figure 3.19, histogram of all participant's F1-scores is given to visualize the distribution of these scores.

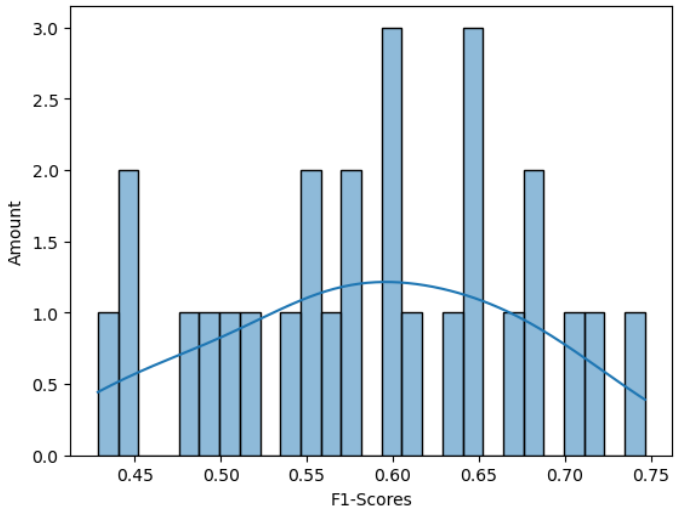


Figure 3.19: All Participants' F1-Scores for Isomap Method.

t-SNE model comes second between feature extraction approaches in terms of average F1-score with 61%. Its lowest and highest scores from cross validation results are 45% and 79% respectively, which are slightly higher than Isomap's results. Histogram of these cross-validation scores is given in figure 3.20. Three scores for respective participants are in 40% range and two of them are in 70% range. Most of the results are stacked around 60%. Hyperparameter tuning with grid search shows that the most frequent parameter couples for this model are path_smooth = 5, min_child_samples = 33, max_depth = 3, max_bin = 10, learning_rate = 0.03, and num_iterations(number of trees) = 100.

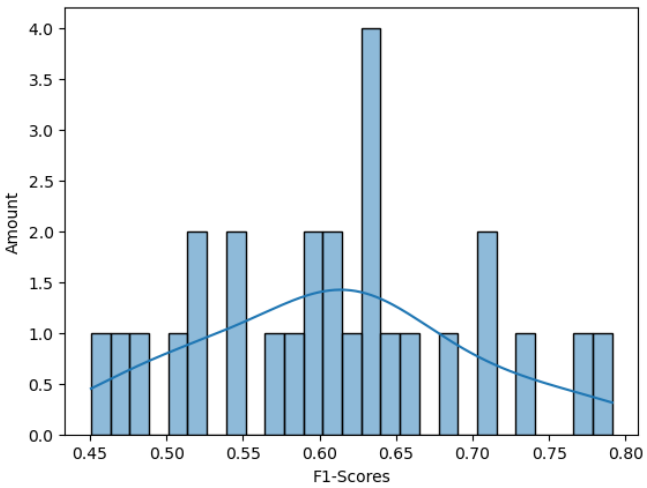


Figure 3.20: All Participants' F1-Scores for t-SNE Method.

As a final feature extraction model, PCA's cross-validation results are given in figure 3.21 below. PCA has the highest average F1-score among feature extraction methods with 62%. There is not much difference between t-SNE average score, hence Wilcoxon signed-rank test results will be critical in deciding which might be a better option. Lowest and highest scores of this model are 45% and 81% respectively. This indicates that PCA also has the highest individual score. Three of its scores are higher than 70% and its lowest score lies alone in the 40% range. Its score distribution has a slight positive skewness due to some of the scores pile up at 50% range below average score. Best parameters selected for this model are path_smooth = 3, min_child_samples = 35, max_depth = 3, max_bin = 10, learning_rate = 0.03, and num_iterations(number of trees) = 100.

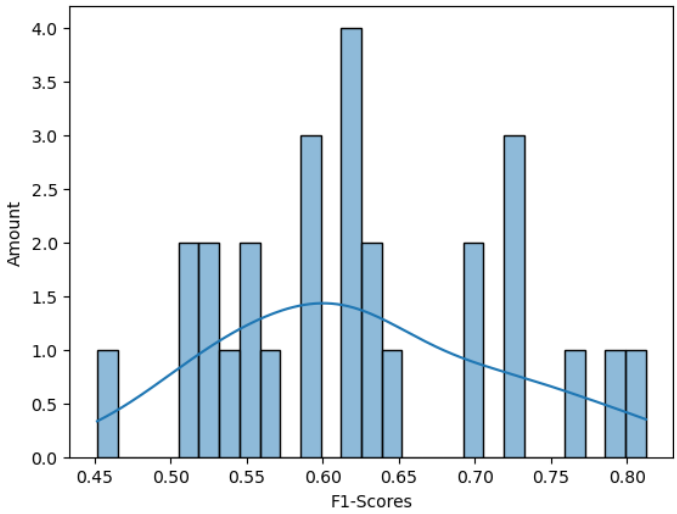


Figure 3.21: All Participants' F1-Scores for PCA method.

Boxplot below in figure 3.22 enables better visual comparison between feature extraction and main models. Main model and PCA have almost the same range in terms of participants' F1-scores, with the main model having the highest average score. Isomap has the shortest range and also lower mean and median value for F1-scores compared to other models. Although the main model has the highest average score, Wilcoxon signed-rank test might give better idea by comparing it with other models with their respective participant scores if there is statistical difference between those results.

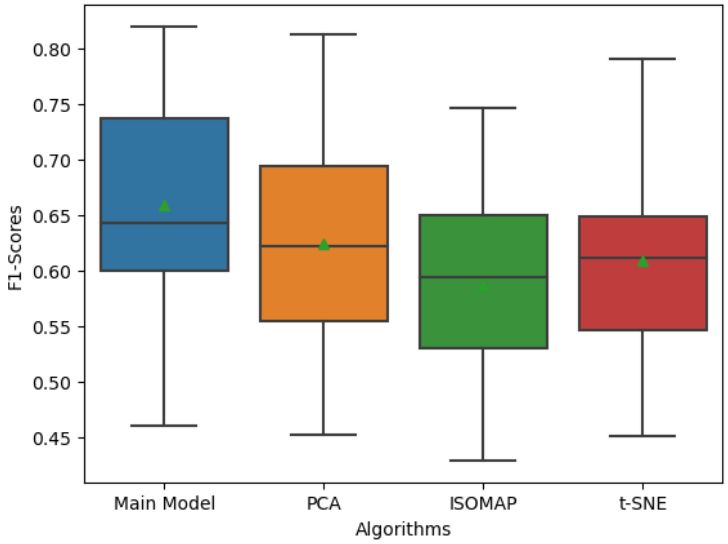


Figure 3.22: Boxplot of Feature Extraction and Main Models' Scores.

Wilcoxon signed-rank results are given as a matrix in table 3.7 below where upper diagonal results have been masked due to being symmetrical with lower scores. Significant p-values are highlighted with bold indicating there is statistically difference between their respective participant scores. Therefore, the highest average score achieved with the main model (66%) is supported by this test with $p < .05$. Another significant result is that the PCA (62%) model has statistically higher scores than the Isomap (59%) model ($p = .016$). On the other hand, there aren't any significant differences between PCA (62%) and t-SNE (61%), and also t-SNE (61%) and Isomap (59%) couples ($p > .05$).

Table 3.7 : Wilcoxon Signed-Rank Test Results for Feature Extraction and Main Models

p-values	Main Model	PCA	ISOMAP	t-SNE
Main Model	-	-	-	-
PCA	**0.003**	-	-	-
ISOMAP	**0.000**	**0.016**	-	-
t-SNE	**0.001**	0.239	0.136	-

As final evaluation for this section, permutation test results are given in figures 3.23-3.25 below. All three feature extraction models have not significant results for this test, indicating these models could not find a good relationship between features and labels ($p > .05$). On the other hand, the permutation test for the main model was

shown in previous section where it had statistically significant result meaning there is a strong link between its features and labels.

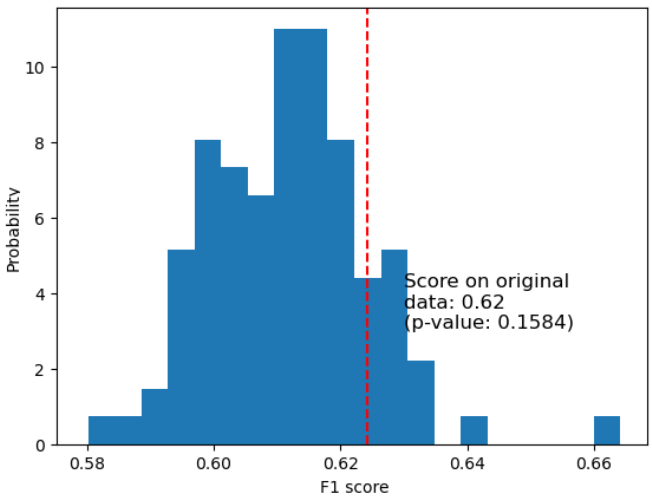


Figure 3.23: PCA Model's Permutation Test Results.

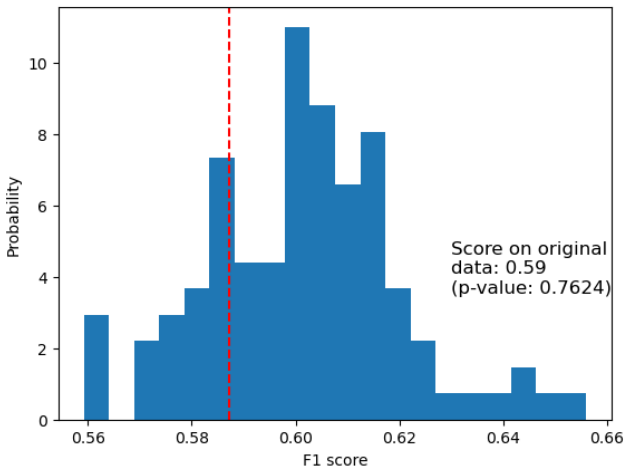


Figure 3.24: Isomap Model's Permutation Test Results.

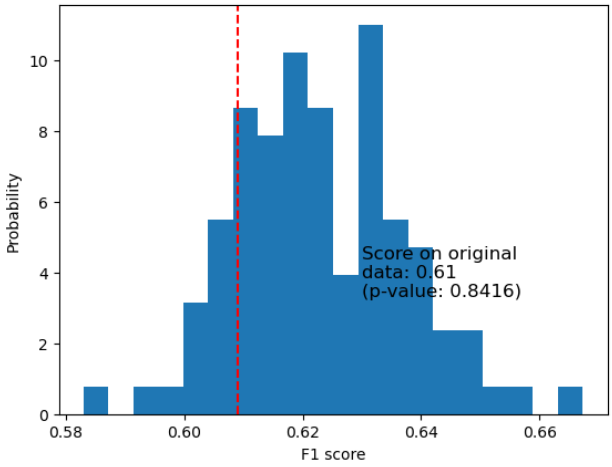


Figure 3.25: t-SNE Model's Permutation Test Results.

Both average scores, and statistical test results show that LGBM without any feature extraction applied might be the best binary classification model compared to feature extraction models. On the other hand, feature extraction models training times are lower than main models due to decreased feature dimension. PCA's and Isomap's training times were respectively 70% and 40% lower than the main model's training time. t-SNE could only reduce around 10% of training time of the main model while having lower scores. From these results, one might trade off performance over training time by using PCA if that is necessary.

In the next section the wrapper method's results will be shown by implementing the main model due to its higher score.

3.4.3. Wrapper model

Another way of removing irrelevant features to improve predictive models is using feature selection methods. A wrapper model has been formed in this study with the best performing algorithm in the main modeling section. In the feature extraction modeling section, it was found that the main model got a higher average score than feature extraction models, therefore a wrapper approach was implemented on it.

In table 3.8, there are average scores for the main model with and without wrapper method. Highest score for each evaluation metric was highlighted in bold. Wrapper model got the highest score for precision while the main model got the highest scores for f1-score and also recall. Therefore, the main model could predict

most of the liked visuals correctly with 92% accuracy whereas the wrapper model could only have 75% accuracy for positive class. Precision scores are close, hence these results caused the main model's f1-score to be higher than the f1-score of the wrapper model.

Table 3.8 : Average Scores of Main and Wrapper Models

Models	F1-Score	Precision	Recall
Main Model	**0.66 (0.08)**	0.53 (0.13)	**0.92 (0.08)**
Wrapper Approach	0.61 (0.09)	**0.54 (0.14)**	0.75 (0.1)

Half of the hemodynamic measurements were selected (64) in the feature selection phase, and with the rest of the features which were FAA index, and cluster features, cross-validation evaluation was made by testing every participant liking preference separately in each fold. F1-score distribution belonging to all these scores for the wrapper model is given in figure 3.26. Highest score achieved is 79% whereas lowest is 39%. There are two scores lower than 50% and most of them lie above 55%. Best parameters for this model were found as path_smooth = 4, min_child_samples = 39, max_depth = 7, max_bin = 100, learning_rate = 0.03, and num_iterations(number of trees) = 100.

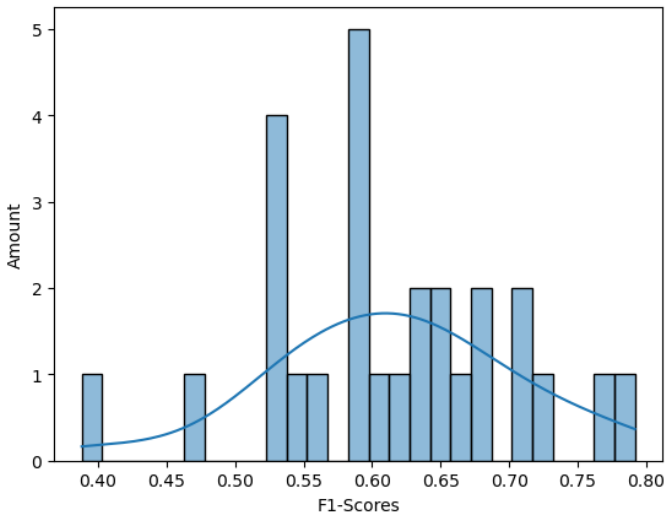


Figure 3.26: All Participants' F1-Scores for Wrapper Model.

Main model's median score is also higher than the wrapper model as can be seen in figure 3.27. Mean scores are shown with a green triangle on the boxplot for

each model. Main model achieved the highest score in terms of individual scores with 82% against the wrapper model's 79%. Moreover, the wrapper model's lowest score lies at 39% against the main model's 46% score.

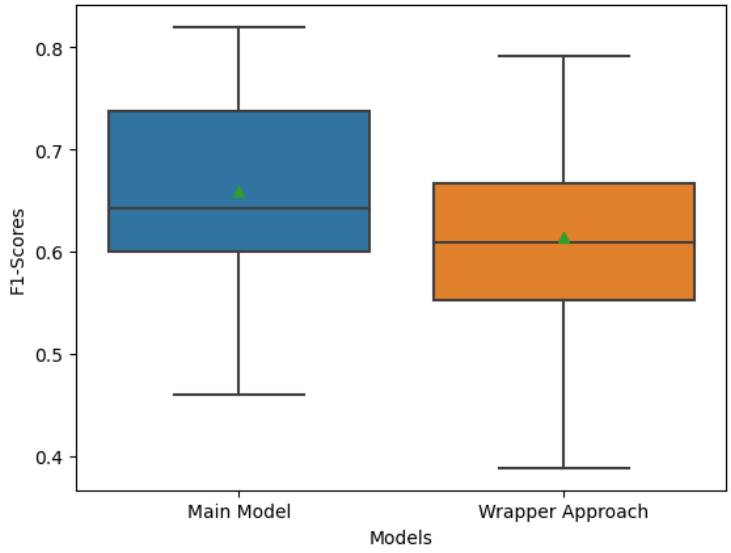


Figure 3.27: Boxplot of Wrapper and Main Models' Scores.

Wilcoxon signed-rank test was applied on participant scores of main and wrapper models to test if their respective scores differ, meaning if one's scores outperform others. Test result elicits that application of wrapper approach on main model diminishes predictive power of the model, as seen in the table 3.9 ($p < .05$). Therefore, it might be said that the main model's mean score of 66% is confirmed to be higher than the wrapper model's average score of 61%. The reason the wrapper approach could not deliver higher performance than the main model might be due to unselected features being important in prediction of liking. Due to computational power constraint, features to be selected could not be increased further to look into those potential features, and it would also increase the dimension of the features, getting it closer to the main model's dimensionality which is not preferred for this model.

Table 3.9 : Wilcoxon Signed-Rank Test Result for Wrapper and Main Models

Wilcoxon Signed Rank Test	Main - Wrapper Models
p-value	0.00026

Permutation result for the wrapper model is given in figure 3.28. Mean score of 61% has been statistically tested if the model could find relevance between features and classes when doing the classification. It appears that the wrapper model could not find a good link between them, indicating the obtained mean score is not better than by chance alone ($p > .05$).

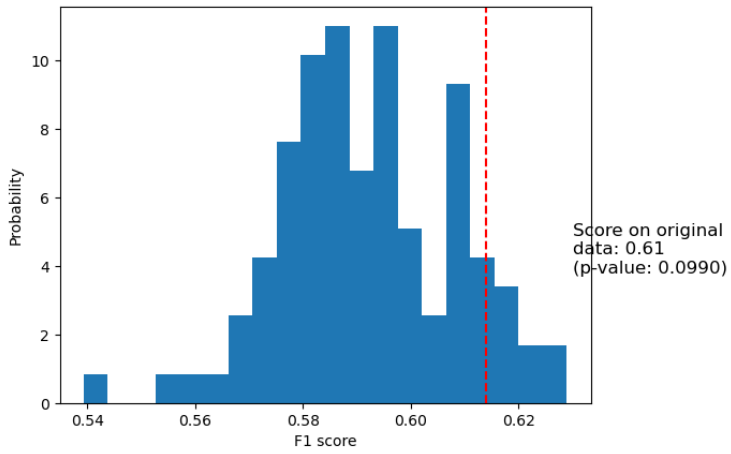


Figure 3.28: Permutation Test Result for Wrapper Model

As a summary for this thesis, firstly, studies related to classification models that are formed by measuring hemodynamic activity by using fNIRS with the help of machine learning algorithms are investigated. Following this, theoretical background of decision-making and emotional processes, role of different brain regions related to preference of liking, and machine learning approaches are explained.

In order to form a model, firstly measurements taken from experiment trials were analyzed. Exploratory data analysis was made by deducing statistical properties of the data set, analyzing missing values, presence of outliers and looking into correlation between all features and labels, i.e., whether visual stimuli was liked or not. After those analysis, data cleaning, outlier removal, missing value imputation were done with this order. IQR method was used to spot and remove outliers that may diminish the performance of the models, from the dataset. Two approaches have been implemented for outlier detection, both using the IQR method but while one masks the outliers, the other caps it to the lower and upper boundaries calculated by IQR. Later one was selected as the outlier removal method as masking of the outliers meant creating more NA values which may deteriorate the classification process. For the imputation section, various imputation methodologies were applied and compared,

such as mean, median, MICE, iterative, KNN and neighbor imputations. Neighbor imputation was selected in this iteration as the most appropriate imputation approach as its features' overall distribution is closest to normal distribution compared to other ones, which is considered to be important while implementing machine learning algorithms.

Following those data engineering parts, as some of the individuals fNIRS measurements might resemble each other, K-Means clustering was applied to cluster those similar participant measurements. Due to the difference in activity of the left and right hemispheres of the brain which could discriminate whether an individual felt positive emotions that might be related to preference of its liking in this experiment, an index was created from this phenomenon that is called frontal alpha asymmetry. As the final data preparation processes, one-hot encoding and standardization of the numeric data was applied.

Generally, precision score came lower compared to recall score which rises to 92%. For this study, predicting the positive class i.e., liking of a stimulus, might be more important than predicting the negative class (dislike), hence making recall more important than precision as for the future works that could make this fNIRS method to be more applicable in a practical real-life environment, prediction of liking of a product could be used as commercial or as recommendation engines to the specific groups. This means that showing a product that is thought to be liked for that target group could come out wrong as in the low precision cases but favoring higher accuracy for liking class over predicting dislikes wrong might contribute more to the success of the potential commercial.

For the main goal, results elicit that LGBM was the superior algorithm, hence chosen for the next iteration as the main model. Both of its mean f1-score, Wilcoxon signed rank test and permutation test made it to be the best model among others. Proceeding to the feature extraction models, there was not enough proof that PCA or t-SNE was better than one or other, whereas both outperform the Isomap model. On the other hand, the main model without any of those methods applied, came out to be a better model by looking at statistical tests. Therefore, wrapper model phase was done without feature extraction methods, and wrapper approach also could not further increase performance of the main model, which elicits that all hemodynamic

measurements coming taken from different parts of prefrontal cortex, and their multicollinearity might contribute to the predictive power of the model collaboratively, making main model with the LGBM algorithm, the best choice.

3.4.4. VIF feature reduction model

An additional model has been developed with collinear features eliminated with variance inflation factor (VIF). This model has been formed with the LGBM algorithm and PCA. Due to data having multicollinear property, and from the results of models with PCA, it was thought that this collinearity was not dealt with. Therefore, an extra method that deals with this situation has been implemented before doing PCA.

VIF is a method that measures multicollinearity in a dataset. It is generally applied to regression analysis. However, since classification problems might also be negatively affected by multicollinearity, it was applied to this binary classification problem.

As stated in the data understanding section, there was 128 hemodynamic features that were used in previous models. Contrary to use all of them, average hemodynamic measurements of image viewing, and decision-making parts were calculated for only Hbo and Hbr features, since these two have been used more extensively than Hbt and oxy parameters. Therefore, total of 32 features were left and they were analyzed by calculating their respective VIF values to eliminate collinear features. VIF value equal to 1 means that an independent variable is not correlated to the remaining ones, where VIF between 1 and 5 means there is a moderate correlation to others, and higher than 5 indicates that it is strongly correlated with others. In table 3.10, features with VIF lower than 5 has been given and selected as input for this model.

Table 3.10 : Features Having Low Collinearity with Others (VIF < 5)

feature	VIF
hbr1	1.70
hbr2	2.94
hbr3	2.84
hbr5	3.62
hbr7	2.93
hbr9	3.30
hbr11	4.02
hbr13	3.92
hbr15	3.05
hbo1	4.96

With VIF, 10 features were selected and put into the model. Standardization and PCA applied before putting into the LGBM classifier, with the same order as in the PCA model in the feature extraction models section. Explained variance has been taken as 85% in PCA, corresponding to selecting top five principal components. F1-score appeared to be as 0.65, stating that there was no improvement in performance compared to the best model with score 0.66.

3.4.5. Lasso feature reduction models

As a last model, a different approach has been implemented for feature selection or reduction, called Least Absolute Shrinkage and Selection Operator (Lasso). It is a regularization and also variable selection technique. Regularization adds a penalty term to the loss function to avoid overfitting. It discourages the model from learning overly complex relationships between features and labels, thus avoiding poor generalization. Lasso uses L1 to achieve regularization.

All of 128 hemodynamic measurements have been put into Lasso to spot the most important ones between them. First, all features have been standardized and then Gridsearch with 5-fold cross validation was used to tune the ‘alpha’ hyperparameter of the Lasso between the values 0.1 and 10 by increasing 0.1 each time. It is a coefficient that determines L1 regularization. Therefore, a higher alpha value results in stronger regularization, meaning a larger penalty for the magnitude of the model coefficients, whereas lower alpha leads to a weaker regularization that makes the model fit into training data more closely.

After searching for the best hyperparameter, it was found as 1.9, and with it 27 out of 128 features have been excluded from the dataset. Then, LGBM was used for the binary classification of liking preference, with the same setup in the main model i.e., same hyperparameter tuning with Gridsearch, and LOGOCV. On top of main model setup (with 101 features), three other approaches were also implemented. Those were PCA applied to the features selected by Lasso (14 features left after PCA with keeping 85% of variance), PCA applied to the features selected by Lasso and then additional selection of the principal components by wrapper method (15 features after PCA and wrapper, with keeping 95% of variance), and finally after Lasso and PCA additional feature selection has been made on principal components with Lasso again (8 feature left after PCA and Lasso with keeping 85% of variance). Therefore, there has been totally four models developed in this section and the corresponding average f1-scores of the models from cross validation results came out to be 0.64, 0.65, 0.6, and 0.62 respectively, indicating there has been still no improvement compared to the best model i.e., the main model with LGBM that is in the main model results section.

CONCLUSION

The main goal of this thesis was to create a model that predicts the preference of liking of individuals who were shown various visual stimuli using various machine learning algorithms i.e., k-Nearest Neighbor, Support Vector Machines, Random Forests, XGBoost, and LightGBM. Comparisons of them were made using statistical tests such as Wilcoxon signed-rank test and permutation tests. Secondary goal was to apply feature extraction methodologies which were Principal Component Analysis, t-Stochastic Neighbor Embedding and Isometric Mapping, to further improve the predictive power of the model. Finally, the last goal was to implement a wrapper approach to select the best features that come from the best model in the previous steps to have a more solid performance. The goals mentioned were all iterative processes which started from the main goal and continued to the last goal by selecting the best approaches in each process.

After data preparation steps which include data cleaning, outlier detection and removal, missing value imputation, adding FAA index as a new feature, data clustering with K-Means algorithm, one-hot encoding transformation and standardization of numeric features, the main model was formed using five ML algorithms. Among ML algorithms, LGBM appeared to be the best choice which was confirmed with its highest average f1-score compared to others that was tested with statistical tests such as Wilcoxon Signed-Rank test that tested the statistical significance of the difference in f1-scores of models, and permutation test that elicits if the models' results are better than by chance alone. Applying feature extraction and feature selection methods could not contribute additional improvements to the model in terms of predictive power, making the main model with the LGBM algorithm is the best approach.

As a future work, in addition to the usage of fNIRS devices, electroencephalogram (EEG) and heart rate monitors might be used in the future works for an attempt to enhance performance of the implemented models. Moreover, eye-tracking devices could be taken advantage of which might be implemented as a complementary approach to the fNIRS measurements to improve classification performance in future works.

REFERENCES

[1] M. P. Çakir, T. Çakar, Y. Girisken, and D. Yurdakul, "An investigation of the neural correlates of purchase behavior through fNIRS," *Eur J Mark*, vol. 52, no. 1–2, pp. 224–243, Feb. 2018, doi: 10.1108/EJM-12-2016-0864.

[2] M. Y. Köksal, T. Çakar, E. Tuna, and Y. Girişken, "Liking Prediction Using fNIRS and Machine Learning: Comparison of Feature Extraction Methods," in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, 2022, pp. 1–4.

[3] M. Kumagai, "Extraction of Personal Preferences Implicitly using NIRS," 2012.

[4] J. Hennrich, C. Herff, D. Heger, and T. Schultz, "Investigating deep learning for fNIRS based BCI," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, Nov. 2015, vol. 2015-November, pp. 2844–2847. doi: 10.1109/EMBC.2015.7318984.

[5] S. Kaheh, M. Ramirez, J. Wong, and K. George, "Neuromarketing using EEG Signals and Eye-tracking," Dec. 2021, pp. 1–4. doi: 10.1109/conecct52877.2021.9622539.

[6] M. Ramirez, S. Kaheh, and K. George, "Neuromarketing Study Using Machine Learning for Predicting Purchase Decision," in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2021, pp. 560–564.

[7] B. Yılmaz, S. Korkmaz, D. B. Arslan, E. Güngör, and M. H. Asyalı, "Like/dislike analysis using EEG: determination of most discriminative channels and frequencies," *Comput Methods Programs Biomed*, vol. 113, no. 2, pp. 705–713, 2014.

[8] S. M. H. Hosseini, Y. Mano, M. Rostami, M. Takahashi, M. Sugiura, and R. Kawashima, "Decoding what one likes or dislikes from single-trial fNIRS measurements," *Neuroreport*, vol. 22, no. 6, pp. 269–273, 2011.

[9] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif Intell*, vol. 97, no. 1–2, pp. 273–324, 1997.

[10] K. Steele and H. O. Stefánsson, "Decision Theory," in *The Stanford Encyclopedia of Philosophy*, Winter 2020., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2020.

[11] D. Kahneman, *Thinking, fast and slow*. Macmillan, 2011.

[12] P. Slovic, B. Fischhoff, and S. Lichtenstein, "Behavioral decision theory," *Annu Rev Psychol*, vol. 28, no. 1, pp. 1–39, 1977.

[13] M. Pelowski, M. Oi, T. Liu, S. Meng, G. Saito, and Saito H., “Understand after like, viewer’s delight? A fNIRS study of order-effect in combined hedonic and cognitive appraisal of art,” *Acta Psychol (Amst)*, vol. 170, pp. 127–138, 2016.

[14] D. E. Berlyne, “Studies in the new experimental aesthetics: Steps toward an objective psychology of aesthetic appreciation. - PsycNET,” 1974. https://psycnet.apa.org/record/1975-07344-000 (accessed Feb. 23, 2022).

[15] G. C. Cupchik and J. László, *Emerging visions of the aesthetic process: In psychology, semiology, and philosophy*. Cambridge University Press, 1992.

[16] H. Leder, C. C. Carbon, and A. L. Ripsas, “Entitling art: Influence of title information on understanding and appreciation of paintings,” *Acta Psychol (Amst)*, vol. 121, no. 2, pp. 176–198, Feb. 2006, doi: 10.1016/j.actpsy.2005.08.005.

[17] G. C. Cupchik, O. Vartanian, A. Crawley, and D. J. Mikulis, “Viewing artworks: Contributions of cognitive control and perceptual facilitation to aesthetic experience,” *Brain Cogn*, vol. 70, no. 1, pp. 84–91, Jun. 2009, doi: 10.1016/J.BANDC.2009.01.003.

[18] D. Summers, *The judgment of sense: Renaissance naturalism and the rise of aesthetics*, vol. 5. Cambridge University Press, 1990.

[19] G. C. Cupchik, “I am, therefore I think, act, and express both in life and in art,” in *Art and identity*, Brill, 2013, pp. 67–91.

[20] R. Reber *et al.*, “Processing Fluency and Aesthetic Pleasure: Is Beauty in the Perceiver’s Processing Experience?,” 2004.

[21] R. Reber, P. Wurtz, and T. D. Zimmermann, “Exploring ‘fringe’ consciousness: The subjective experience of perceptual fluency and its objective bases,” *Conscious Cogn*, vol. 13, no. 1, pp. 47–60, Mar. 2004, doi: 10.1016/S1053-8100(03)00049-7.

[22] R. Reber, T. A. Fazendeiro, and P. Winkielman, “Processing Fluency as the Source of Experiences at the Fringe of Consciousness,” 2002. [Online]. Available: http://psyche.cs.monash.edu.au/v8/psyche-8-10-reber.html

[23] T. Vogel, R. R. Silva, A. Thomas, and M. Wänke, “Truth is in the mind, but beauty is in the eye: Fluency effects are moderated by a match between fluency source and judgment dimension,” *J Exp Psychol Gen*, vol. 149, no. 8, pp. 1587–1596, Aug. 2020, doi: 10.1037/xge0000731.

[24] Peter Politser, *Neuroeconomics: A guide to the new science of making choices*. New York: Oxford University Press Inc., 2008.

[25] T. Grossmann, “The role of medial prefrontal cortex in early social cognition,” *Front Hum Neurosci*, vol. 7, p. 340, 2013.

[26] P. Yuan and N. Raz, "Prefrontal cortex and executive functions in healthy adults: a meta-analysis of structural neuroimaging studies," *Neurosci Biobehav Rev*, vol. 42, pp. 180–192, 2014.

[27] W.-Z. Liu *et al.*, "Identification of a prefrontal cortex-to-amygdala pathway for chronic stress-induced anxiety," *Nat Commun*, vol. 11, no. 1, pp. 1–15, 2020.

[28] P. Domenech and E. Koechlin, "Executive control and decision-making in the prefrontal cortex," *Curr Opin Behav Sci*, vol. 1, pp. 101–106, 2015.

[29] C. A. Hutcherson, H. Plassmann, J. J. Gross, and A. Rangel, "Cognitive regulation during decision making shifts behavioral control between ventromedial and dorsolateral prefrontal value systems," *Journal of Neuroscience*, vol. 32, no. 39, pp. 13543–13554, Sep. 2012, doi: 10.1523/JNEUROSCI.6387-11.2012.

[30] S. Funahashi, "Working memory in the prefrontal cortex," *Brain Sciences*, vol. 7, no. 5. MDPI AG, May 01, 2017. doi: 10.3390/brainsci7050049.

[31] A. Koizumi, D. Mobbs, and H. Lau, "Is fear perception special? Evidence at the decision-making and metacognitive levels," *Soc Cogn Affect Neurosci*, vol. 11, no. 11, pp. 1772–1782, Nov. 2016, doi: 10.1093/scan/nsw084.

[32] Z. Cattaneo, C. Ferrari, S. Schiavi, I. Alekseichuk, A. Antal, and M. Nadal, "Medial prefrontal cortex involvement in aesthetic appreciation of paintings: a tDCS study," *Cogn Process*, vol. 21, no. 1, pp. 65–76, Feb. 2020, doi: 10.1007/s10339-019-00936-9.

[33] T. Çakar, K. Rızvanoğlu, Ö. Öztürk, D. Z. Çelik, and İ. Gürvardar, "The use of neurometric and biometric research methods in understanding the user experience during product search of first-time buyers in E-commerce," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10288 LNCS, pp. 342–362. doi: 10.1007/978-3-319-58634-2_26.

[34] G. Vecchiato *et al.*, "On the use of EEG or MEG brain imaging tools in neuromarketing research," *Comput Intell Neurosci*, vol. 2011, 2011.

[35] G. Vecchiato *et al.*, "Changes in brain activity during the observation of TV commercials by using EEG, GSR and HR measurements," *Brain Topogr*, vol. 23, no. 2, pp. 165–179, 2010.

[36] F. Irani, S. M. Platek, S. Bunce, A. C. Ruocco, and D. Chute, "Functional near infrared spectroscopy (fNIRS): an emerging neuroimaging technology with important applications for the study of brain disorders," *Clin Neuropsychol*, vol. 21, no. 1, pp. 9–37, 2007.

[37] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, "Machine learning: A review of classification and combining techniques," *Artif Intell Rev*, vol. 26, no. 3, pp. 159–190, Nov. 2006, doi: 10.1007/s10462-007-9052-3.

[38] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, “Data cleaning: Overview and emerging challenges,” in *Proceedings of the 2016 international conference on management of data*, 2016, pp. 2201–2206.

[39] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artif Intell Rev*, vol. 22, no. 2, pp. 85–126, 2004.

[40] C. L. Blake and C. J. Merz, “UCI repository of machine learning databases, 1998.” 1998.

[41] A. F. Siegel, *Statistics and data analysis: an introduction*. Wiley, 1988.

[42] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, “Informal identification of outliers in medical data,” in *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, 2000, vol. 1, pp. 20–24.

[43] V. Barnett and T. Lewis, “Outliers in statistical data,” *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, 1984.

[44] D. B. Skalak and E. L. Rissland, “Inductive Learning in a Mixed Paradigm Setting.,” in *AAAI*, 1990, pp. 840–847.

[45] G. H. John, “Robust Decision Trees: Removing Outliers from Databases.,” in *KDD*, 1995, vol. 95, pp. 174–179.

[46] A. R. T. Donders, G. J. M. G. van der Heijden, T. Stijnen, and K. G. M. Moons, “A gentle introduction to imputation of missing values,” *J Clin Epidemiol*, vol. 59, no. 10, pp. 1087–1091, 2006.

[47] A. Jadhav, D. Pramod, and K. Ramanathan, “Comparison of performance of data imputation methods for numeric dataset,” *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, 2019.

[48] J. L. Schafer and J. W. Graham, “Missing data: our view of the state of the art.,” *Psychol Methods*, vol. 7, no. 2, p. 147, 2002.

[49] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. “ O’Reilly Media, Inc.,” 2019.

[50] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[51] L. Breiman, “Bagging predictors,” *Mach Learn*, vol. 24, no. 2, pp. 123–140, 1996.

[52] R. Wirth and J. Hipp, “CRISP-DM: Towards a standard process model for data mining,” in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 2000, vol. 1, pp. 29–39.

[53] R. C. Oldfield, “The assessment and analysis of handedness: the Edinburgh inventory,” *Neuropsychologia*, vol. 9, no. 1, pp. 97–113, 1971.

[54] H. Ayaz *et al.*, “Registering fNIR data to brain surface image using MRI templates,” in *2006 international conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 2671–2674.

[55] H. Obrig *et al.*, “Near-infrared spectroscopy: does it function in functional activation studies of the adult brain?,” 2000.

[56] H. Ayaz, P. A. Shewokis, S. Bunce, K. Izzetoglu, B. Willems, and B. Onaral, “Optical brain monitoring for operator training and mental workload assessment,” *Neuroimage*, vol. 59, no. 1, pp. 36–47, Jan. 2012, doi: 10.1016/j.neuroimage.2011.06.023.

[57] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.

[58] Law E, “Impyute,” 2017. https://impyute.readthedocs.io/en/master/

[59] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[60] A. Ercan, B. Karan, and T. Çakar, “Köpek Gezdirici Segmentasyonu Dog Walker Segmentation,” in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, 2022, pp. 1–4.

[61] K. Kirasich, T. Smith, and B. Sadler, “Random forest vs logistic regression: binary classification for heterogeneous datasets,” *SMU Data Science Review*, vol. 1, no. 3, p. 9, 2018.

[62] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.

[63] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[64] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, pp. 273–297, 1995.

[65] T. Bozkan, T. Çakar, A. Sayar, and S. Ertuğrul, “Customer Segmentation and Churn Prediction via Customer Metrics,” in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, 2022, pp. 1–4.

[66] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Ann Stat*, pp. 1189–1232, 2001.

[67] LightGBM, “LightGBM Documentation,” 2022. https://lightgbm.readthedocs.io/en/latest/index.html# (accessed Jan. 20, 2022).

[68] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *2014 science and information conference*, 2014, pp. 372–378.

[69] I. Guyon and A. Elisseeff, “An introduction to feature extraction,” in *Feature extraction*, Springer, 2006, pp. 1–25.

[70] J. B. Tenenbaum, V. de Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science (1979)*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[71] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE.,” *Journal of machine learning research*, vol. 9, no. 11, 2008.

[72] P. Good, *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media, 2013.,

[73] D. W. Zimmerman, “A note on preliminary tests of equality of variances,” *British Journal of Mathematical and Statistical Psychology*, vol. 57, no. 1, pp. 173–181, 2004.