Will You Be My Love?
Match Outcome Prediction In Fluttr Dating App

A study submitted in partial fulfilment
of the requirements for the degree of
MSc Data Science

at

THE UNIVERSITY OF SHEFFIELD

by

Muhammad Fathi Fadlian
210132797

Word-length: *13224*

August 2022

Contents

Abstract

Background: With the increasing popularity, risks, and benefits from the usage of dating app, dating app has been a compelling subject to be learned. However, although with the abundance of features and filters of the dating app, some users still find it is hard to find a match. In addition, the availability of a good match predictor is assumed to be beneficial for both users and the platform. Hence, an automated method is proposed to predict the outcome of a match from the dating app using available variables from the real-life dataset.

Aims: This project aims to study a real-world dataset of real users with organic interactions and understand the factors and users' traits that can lead to a 'match' in the dating app. Along with that, we employ automated methods to estimate the likelihood of two users matching each other.

Methods: To achieve the aims of this study, the quantitative research method is conducted to describe and gain insight from the Fluttr dataset. The first step is to construct the data and the features. After conducting data pre-processing and feature engineering, an investigation of the combination of features and machine learning algorithm is conducted. The performance of each experiment then evaluated using Matthew's Correlation Coefficient

Results: In order to discover which, feature combinations and algorithm with good performance, this dissertation first carries out descriptive statistic and association test. After that, through model and combination of features comparison, it discovers that XGBoost is better compared to other algorithms. In addition, the use pf all three features (profile-based, topological, and interaction pattern) results in a better performance compared to other combinations.

Conclusions: From the findings, it tells that the XGBoost model which combined with the use of all three features (profile-based, topological, and interaction pattern) results the best MCC score. However, due to the sparsity of the data and the limited time, this study has not applied several other prospective methods. Thus, the future work can be focuses on the development of match prediction with graph-neural-networks utilising more comprehensive features (pictures, bio, etc).

Acknowledgement

وَمِنْ آيَاتِهِ أَنْ خَلَقَ لَكُمْ مِنْ أَنْفُسِكُمْ أَزْوَاجًا لِتَسْكُنُوا إِلَيْهَا وَجَعَلَ بَيْنَكُمْ مَوَدَّةً وَرَحْمَةً ۚ إِنَّ فِي ذَٰلِكَ لَآيَاتٍ لِقَوْمٍ يَتَفَكَّرُونَ

"And of His signs is that He created for you from yourselves mates that you may find tranquillity in them, and He placed between you affection and mercy. Indeed, that are signs for people who give thought"

The Quran 30:21

Many thanks to my supervisor, Dr Suvodeep Mazumdar, for providing continuous support when I felt lost in the research topic. Also, thanks to the staffs from Fluttr for providing this research opportunity. I would also like to thank my family and friends for encouraging me through the year of study. This research could not have been completed without their help and support.

1 Introduction

Currently, the increasing popularity of online social networks by people of various demographics has resulted in a high increase in the number of online dating companies and has spawned a more social acceptance of making use of these services. (Smith, 2005). Castro & Barrada (2020) also point out that online dating platforms have become a prevalent means by which people start contacting potential romantic or sexual partners of theirs. These studies are also strengthened by the increasing number of online dating services and users in the US (Cacioppo et al., 2013). Anderson et al. (2020) also asserted that over the past decade, using digital dating has become common, with 30% of Americans have used a dating website or app. Although (LeFebvre, 2018) argues that the primary purpose of dating platforms is to use them exclusively for casual sex, researchers in the last decade have found that people use these platforms for various reasons and that sex is not the main reason. Some of the reasons for individuals to use online dating platforms vary based on studies, including relationship factors (e.g., friendship, romantic love), entertainment factors (e.g., curiosity, boredom), and intrapersonal factors (e.g., self-validation, ease of communication) (Botnen et al., 2018a, 2018b; LeFebvre, 2018; Orosz et al., 2018; Ranzini & Lutz, 2017; Sumter et al., 2017a; Sumter & Vandenbosch, 2019; Timmermans & de Caluwé, 2017). In addition to its popularity, there are also positive aspects associated with the use of dating apps. The first positive impact of the dating app is its ability to decrease racial boundaries in a relationship. As asserted by Omolo, (2020), 'crossing the colour line' in a social relationship may signal social progress. However, Phan et al. (2021) asserted several risks associated with dating apps. It has been reported that dating apps have been associated with sexually transmitted infections (Couch et al., 2012; David & Cambre, 2016; Sumter et al., 2017b) as well as risky connections with others who may harm you physically, psychologically, or sexually (Couch et al., 2012). In some studies (Stonard et al., 2014), perpetrators are not the same across dating mediums; however, this is not certain, especially when it comes to personality differences like the Dark Tetrad and gender drive behaviour (Paulhus et al., 2018). In addition, With the increasing number of people using the online dating platform and the variety of factors that risk or benefit people using it, dating app/platform has been a compelling subject to be studied for various disciplines.

Works in the scope of online dating platforms mostly consist of recommender system algorithms, features selection and construction, and matching prediction. For instance, the work of recommender systems in the scope of dating apps was conducted by Rosenfeld et al. (2019), who developed the recommender system by utilising collaborative filtering. Another work by (S. Wang, 2018) asserted the usage of feature construction from users' 'face value' to predict the outcome of a match. These works employed by many online dating sites suggest the compatibility of their members based on a proprietary matchmaking algorithm. One of the topics is the matching prediction for the online dating platform. The matching algorithm is crucial for both the platform and the users. A GWI survey (2021) revealed that dating platform users would likely use the dating platform more if it provided personalized matches based on individual preferences. Vempaty (2016) also asserted that retaining existing users in a dating platform is cheaper than finding a new one. In addition, a better matching prediction means users are more likely to interact with each other, hence improving the users' satisfaction towards the online dating platform (Xia et al., 2014a). As asserted by Xia et al. (2014), when recommending dates on an online dating platform, one has to ensure the suggested users not only match the user's preferences but that they are also interested in the user and will reciprocate (i.e., reply to contact message from the user) the interest. A similar notion was also argued by Pizzato et al. (2006), who stated that different from many other recommendation techniques which are focused on predicting the user's attitude toward the given passive items (e.g., books, movies, etc.) based on their evaluation of other items.

However, users still struggle to select the right dating partner from the enormous number of candidates on the platform (Zang et al., 2017). The hardship of finding a partner in the dating app might be addressed for three reasons. The first reason is Overchoice or choice overload. This term was first coined by Toffler (1970), which asserted that a person might have a difficult time making a decision/choice when faced with many options. This phenomenon can be seen in the nature of online dating platforms. Sophisticated functionalities have been added to the platform, starting from the 'swipe' mechanism, various filters, and in-app games that increase the chance for the users to find 'the one'. While it is true that these functionalities give more chances for the users to meet their probable counterparts, these features also serve so many choices to the users. This phenomenon of the abundance of options makes users harder to decide due to the potential of choosing the wrong options. In addition, It becomes mentally draining to have too many approximately equally good options since they all have to be weighed against one another to determine which is the best (Iyengar & Lepper, 2000). The second reason is that users do not constantly interest in interacting with counterparts after the matches (J. Zhang & Yasseri, 2016). Moreover, in a study by J. Zhang & Yasseri (2016), only half of the interaction is replied by their matches. This phenomenon might also be attributed to the fear of matching with the wrong partner due to the abundance of options, like in the first reason (Pronk & Denissen, 2020). Hence, to answer the demand in the industry for a better matching prediction and to further expand the knowledge on the mentioned topic, this study aims to research a method to predict the matching outcome between users in an online dating platform.

To validate our proposal, the research is conducted in collaboration with Fluttr Limited. Fluttr Limited is a UK-based software company that built Fluttr, a newly launched online dating app (https://fluttrdating.com/) for both android phones and apple users. Fluttr focuses on creating a safe dating platform that prevents and minimises "Tinder Swindler-Style romance fraud" and identity theft. To objectify their aim, Fluttr ensures that all members complete a biometric identity verification before they can interact with other users. The collaboration with Fluttr is essential as they are one of the dating apps that pay more attention to security and ethics. In addition to that, collaboration with Flutter provides a real-world dataset. Although the synthetic dataset might result in better quality AI models, the synthetic dataset often does not represent the objective complexity embedded in the real-world dataset (Naber, 2022). Biases and faulty values in the synthetic dataset often have been removed by the data provider, resulting in clean data ready to be applied to machine learning algorithms. In addition, synthetic data is often already in an ideal form, so there is no need for a data construction process or feature engineering process; hence easier to be trained by the model. However, although real-world data have biases, faulty values, and inconsistencies, real-world data provide a clearer insight into the actual condition in the field. Data acquisition, even the automated one, is not free from faults and errors; hence, it is essential to account for the errors, faults, and biases in building machine learning models.

1.1 Terms and Naming

This essay will use several terms interchangeably following the context. The term "predicting the likelihood of match", "estimate the likelihood of match", "matching prediction", and "predict matching outcome" is used as the term for which this experiment is aimed. "Machine learning algorithms", "machine learning methods", and "machine learning models" are used interchangeably. The terms "features" and "variables" are also used interchangeably; however, the term "features" is more associated with a set of "variables" that describe a specific type of prevalence.

1.2 Project Aims

This project aims to study a real-world dataset of real users with organic interactions and understand the factors and users' traits that can lead to a 'match' in the dating app. Along with that, we employ automated methods to estimate the likelihood of two users matching each other.

1.3 Research Question

RQ1. How can we use automated methods to effectively predict matching outcome/likelihood based on the users' profile, topological factors, and interaction patterns between the users?
RQ2. What factors can help determine the likelihood of a match in a dating app?
RQ3. How can we use real-life data to predict the match outcome of users in a dating app?

1.4 Research Objectives

RO1. Assess the quality of the data to understand the data construction process better
RO2. Conduct Data and Feature Construction to create a form of data that is ready to be analysed
RO3. Conducting exploratory data analysis on the formed data to gain initial insight into the normality and the quality of the data
RO4. Conduct data cleansing and feature engineering to prepare the data such that it can directly be used for machine learning afterwards
RO5. Implementing multiple models (e.g., Random Forest, XGBoost) and combinations of features for predicting the likelihood of a match
RO6. Evaluate and compare the performance of models and the combination of features' for predicting the likelihood of a match in a dating app.

This dissertation is organised as follows: The first part is the introduction, where the aims, objective, and research question are discussed to form a cohesive direction of where the experiment is directed. The second part will focus on the literature review regarding match prediction, which will cover various methods, variables, and algorithms that are used for the matching prediction. In addition, the social, ethical and legal implications of dating apps will also be discussed. The methodologies will be discussed in the third section, which will cover the data collection, data construction, feature engineering, model development and the steps of the experiments. The result of the analysis will be discussed in the fourth part, which will cover both the exploratory data analysis as well as the result of the experiments. The fifth part will discuss the result of the previous part, the limitation, further exploration, and the conclusion of the experiment. Lastly, the bibliography and the code will be presented in the appendix.

2 Literature Review

As a means to gain a deeper understanding of the aspects of this research, the related literature has been examined and reviewed in order to gain more insight into these aspects. The majority of studies performed within the same subject focused on establishing new models for predicting matches between individuals and the creation of features for matching in dating apps. In addition to that, several projects have attempted to study the reciprocal recommender system for a dating app. It is essential to take into account that the primary reference used in this study (Zang et al., 2017; M. Zhang & Chen, 2018) was conducted within the last five years. As a result, it could be thought of as an indication of the fact that this field still possesses a substantial amount of scope for further

research and development. Therefore, the literature review will be discussed with respect to the following context:

2.1 Match Prediction in Dating App

Starting with the ground-breaking work by (Gale & Shapley, 1962), various economic models of marriage markets have been developed to predict the formation of relationships and measure the efficiency of the actual matches in the market. The model of match prediction produced is based on the parameters of mate preferences, matchmaking mechanism, the interaction pattern amongst the participants, and the participant and the matchmaking mechanism. Most studies conducted on the same subject of dating platforms primarily discuss providing online dating users with suggestions of potential romantic dates (Cai et al., 2010; Pizzato et al., 2010; Zhao et al., 2014). There are also some works that study the feature selection and construction of recommender systems and link prediction in dating platforms or social networks (Augusto Pizzato et al., 2006; Barrada et al., 2021; Nayak et al., 2010; Zang et al., 2017). However, specific for the link/matching prediction, there are two of the most recent studies conducted by Zhang & Chen (2018) use the graph artificial neural networks to predict the link between nodes in a network. The work of Xia et al. (2014) also discusses a method to model users' replying behaviour of users in an online dating platform in a graph-based model. Another study by Zang et al. (2017) also used several supervised machine learning methods (i.e., SVM, Naïve Bayes, Logistic Regression and Random Forest) to predict the link between heterosexual users. However, by assessing the previous works on predicting the likelihood of users' match, there are two general perspectives in formulating their data and problem. The first perspective is to formulate the data as a graph, consisting of the users described as a node, and the interaction between the users described as the edges of the graph. The second perspective uses a more conservative approach that formulates the data as regular tabular data, which consist of pairs of interacting entities with their respective features and interaction. The following section will discuss the example and the differences between the perspectives.

2.1.1 Graphical Approach

According to (Cai et al., 2012), the most recent dating platforms are closer to being classified as a social networks. Namely, application and dating platforms like Tinder, Hinge, and Fluttr. The nature of these platforms treats their users as nodes that have links between one another. In order to formulate the data from social networks as a graph, M. Zhang & Chen (2018) describe the users of the social network as nodes and the interaction between the users is described as the edges of the graph. The nodes are linked by the edges; thus, predicting the interaction between the nodes is equal to predicting the link/edges between the nodes. There has been much interest in link prediction in social networks since Liben-Nowell and Kleinberg (2003) first conducted research in this area, and has since been extensively studied. Based on a snapshot of a social network, the link prediction problem pursues to infer which new interactions, or in this case, match, among its members are likely to occur later. These links are formed if there is contact or match between the users. Thus, a link prediction machine learning method is considered more suitable to predict whether one node will have a link to other nodes or not. While key features of network structure have been extensively used in previous studies to predict links (e.g., node degree, graph distance, etc.), some network structure features have been employed for homogeneous networks and, therefore, cannot be applied directly to the bipartite nature of the heterosexual users of an online dating network. For example, as asserted by Xia et al. (2014), the common neighbour effect and Jaccard's coefficient do not directly apply since two males with common neighbours (females) will not communicate with each other on an online dating platform. In addition to this, M. Zhang & Chen (2018) introduced the SEAL graph neural networks framework, which not only able to predict the

link between the nodes using other existing link, but also accounts the features embedded by individual nodes. However, the graph-based approach requires more work on the data since the data that are available are mostly tabular data (Borisov et al., 2022), and we need to convert the

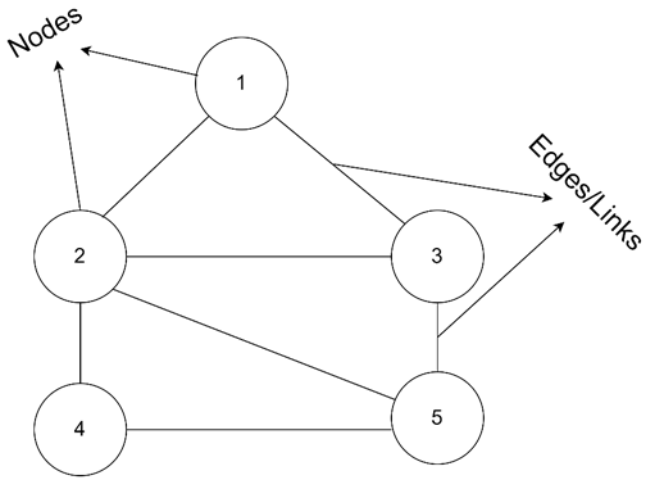

Figure 2.1 – Graph Structure

tabular data to graph format. In addition, regardless of the prediction outcomes, a limited number of algorithms can be applied to this type of data (Shwartz-Ziv & Armon, 2022). Moreover, the use of graph data limits the types of features that can be used in the analysis, as the matching prediction will not only use the interaction/link between the nodes/users but also will assess the usage of profile-based and topological features that are associated with individual nodes.

2.1.2 Pairwise data approach

The work from (Zang et al., 2017) predicts the match outcome using various machine learning algorithms (i.e., SVM, Naïve Bayes, Logistic Regression and Random Forest) with three types of features: profile-based features, graph-based features, and topological features. It is implied that the work was using tabular data with pairs of users and their features as the predictors. Other work by Hitsch et al. (2006) also asserted the use of machine learning algorithms that works on tabular data for matching outcome prediction in an online dating platform. In addition, the use of tabular data widens the perspective of the analysis. The same pairwise tabular data has also been used to predict a sports match's outcomes. For instance, Guan & Wang (2022) and Herbinet (2018) researched the use of several machine learning algorithms and neural networks for predicting a football match. Compared to the graph approach that comes in short when capturing the users'/nodes' features, tabular data can accommodate the use of pairs' individual features for the prediction (Danisik et al., 2018). Zang et al. (2017) asserted that predicting the match outcome in a dating app can be classified as a binary classification problem. The binary labels are "MATCH" or "ACCEPTED" and "NO MATCH" or "NOT ACCEPTED". By breaking down the problem as a binary classification problem, we can use machine learning algorithms for the classification task.

2.2 Machine Learning Algorithms

In order to measure the performance of the algorithm in predicting matches in a dating app, a performance comparison of machine learning algorithms will be conducted in this study (RO5). Regardless of the format of the data, several machine learning algorithms have been applied to predict the match outcome in a dating app.

2.2.1 Random Forest and Gradient Boosting

The accuracy of an ensemble of machine learning models is often more significant than that of a single model. As a standard method, decision trees have received ensemble treatment through bagging methods such as random forests and gradient boosting (Bologna, 2021). A random forest (RF) is made up of multiple decision trees that are assembled with bootstrapping (random sampling replacement), a random selection of subspaces of features, and voting in order to construct predictions (Breiman, 2001). Overfitting issues can be overcome by aggregating the individual predictions of decision trees (Bbeiman, 1996). Thus, predictions of future data are improved as the variance of a single decision tree is reduced. Graduated boost (GB) differs from RF primarily in two ways. As opposed to RF which is built independently through a bagging technique to reduce variance, GB trees build an ensemble of trees sequentially to improve accuracy through bias reduction. Boosting processes use weighted resampling to increase weights for samples with low prediction accuracy at the end of each iteration (Quinlan, 1996; Y. Zhang & Haghani, 2015). Furthermore, whereas RF relies on majority or average votes to decide the final result, GB relies on sequential learning methods to minimise the residuals of a single decision tree model, which is then improved in the next iteration using the negative gradient of the loss function (Friedman, 2002).

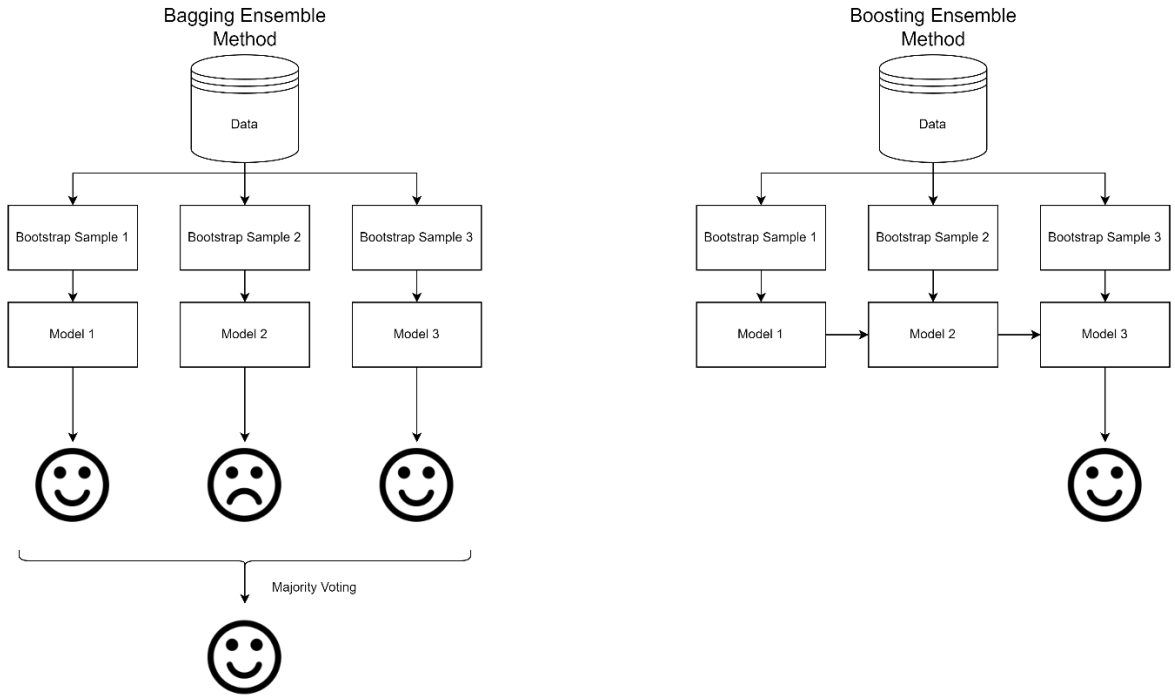

Figure 2.2 Bagging and Boosting algorithm

Random Forest and Gradient Boosting have been popular binary classification methods(Bahel, In, et al., 2020; Grgić et al., 2021; Paul et al., 2017). The study by Bahel et al. (2020) comparing various machine learning algorithm for binary classification tasks resulted in the excellent statistical performance of random forest. However, this algorithm is relatively slow and consumes more significant memory compared to others.

2.2.2 Support Vector Machines (SVM)

SVM is a supervised learning model that can be used for either classification or regression (Suthaharan, 2016). According to Cortes & Vapnik (1995) and Vapnik & Chervonenkis (1974), Statistical learning frameworks or VC theory are at the core of SVM prediction methods, which make for robust predictions. With SVM, data points can be categorized even when they cannot be linearly separated by mapping them into a high-dimensional feature space. Once the separator between the categories is identified, the data are transformed to make it possible to draw the separator as a hyperplane. A new record could then be assigned to a specific group based on the characteristics of its new data. In the SVM, the mathematical function used for the transformation is called as a kernel. There are five kernels that can be used in Scikit Learn Package: *linear, poly, RBF, sigmoid,* and *precomputed,* with *RBF* as the default kernel.

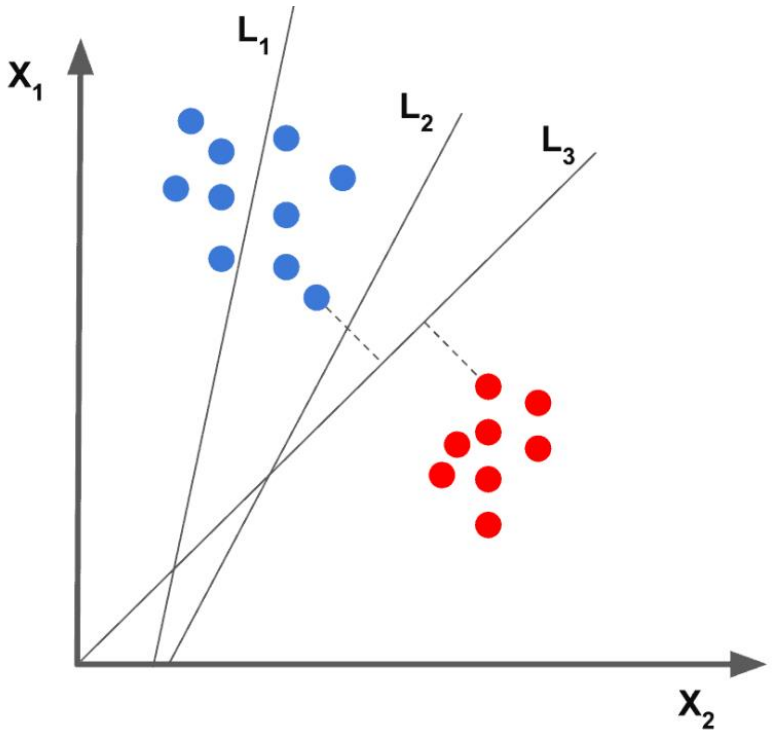

Figure 2.3 SVM (Mallick, 2021)

There have been numerous usages of SVM, particularly for binary classification tasks (Colas & Brazdil, 2006; Mathur & Foody, 2008). As asserted by Colas & Brazdil (2006), SVM outperformed other binary text classification algorithms. In addition to that, Zang et al. (2017) also found the same result where SVM outperformed other machine learning algorithms in predicting the outcome of a match in marriage consulting service.

2.2.3 Logistic Regression

Logistic Regression or Binary Logistic Model or Logit Model is a model of statistics that models the likelihood of an event out of two possible alternatives with log-odds (logarithm of the odds) from one or more predictors or independent variables. In logistic regression, we estimate the probability that an event will occur based on data on independent variables, such as whether two users successfully matched or not. Due to the probability of the outcome, the dependent variable is constrained to 0 and 1. The odds in logistic regression are transformed with a logit transformation, which is the probability of success divided by the probability of failure.

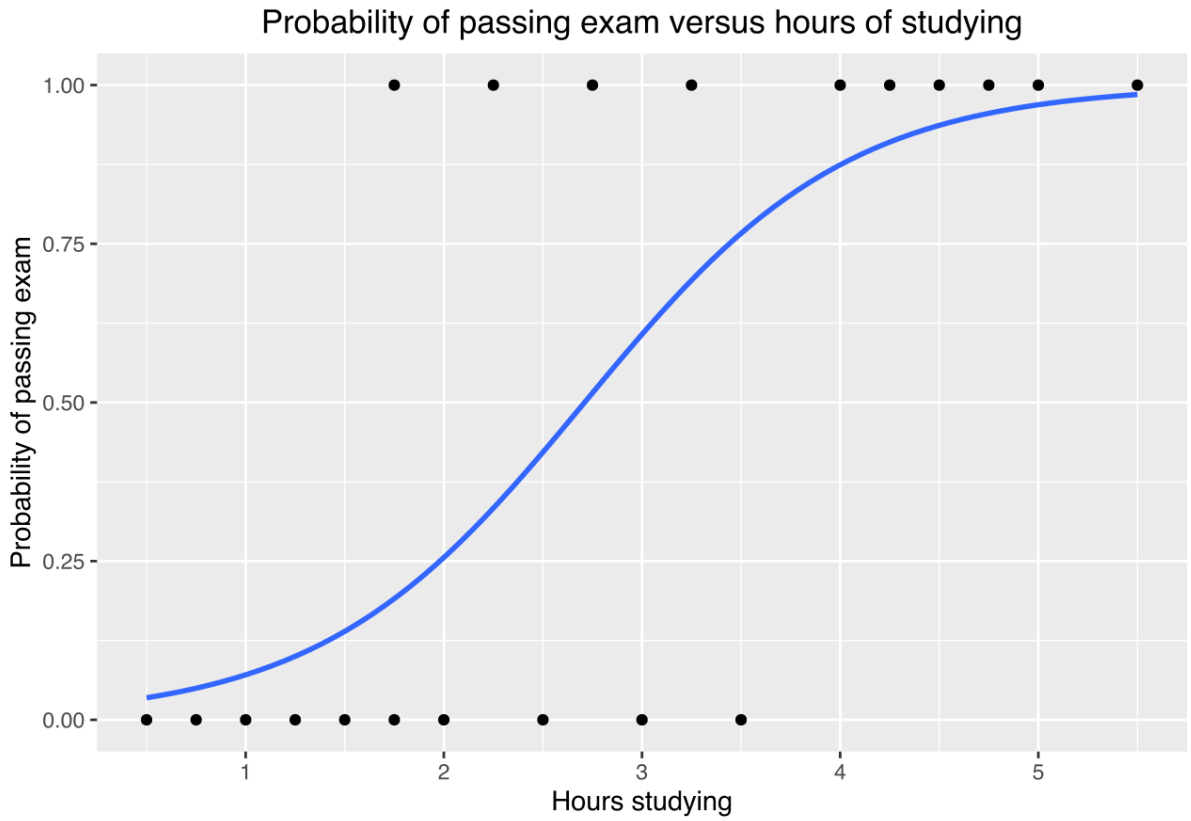

Figure 2.4 Example of Logistic Regression (Canley, n.d.)

Due to its properties mentioned before, Logistic Regression, by nature, works for binary classification tasks. There are various examples of the use of this algorithm; for instance Kirasich et al., (2018) and Mccormick et al. (2012) studied the use of logistic regression for binary classification task on heterogenous dataset. In addition, Feng et al. (2014) developed robust logistic regression algorithm for classification.

2.2.4 Gaussian Naïve Bayes

Generally, Gaussian Naive Bayes classifiers can be defined as simple "probabilistic classifiers" based on Bayes' theorem and strong (naive) independence assumptions between features. Although these types of algorithms are among the simplest Bayesian network models, they are capable of high accuracy when combined with kernel density estimation. According to (Herbinet, 2018), the advantage of using Gaussian Naïve Bayes for binary classification is that the algorithm is scalable. However, it is known that the predicted probabilities are not entirely accurate even if the classifier is robust enough to ignore the naive assumption (Herbinet, 2018).

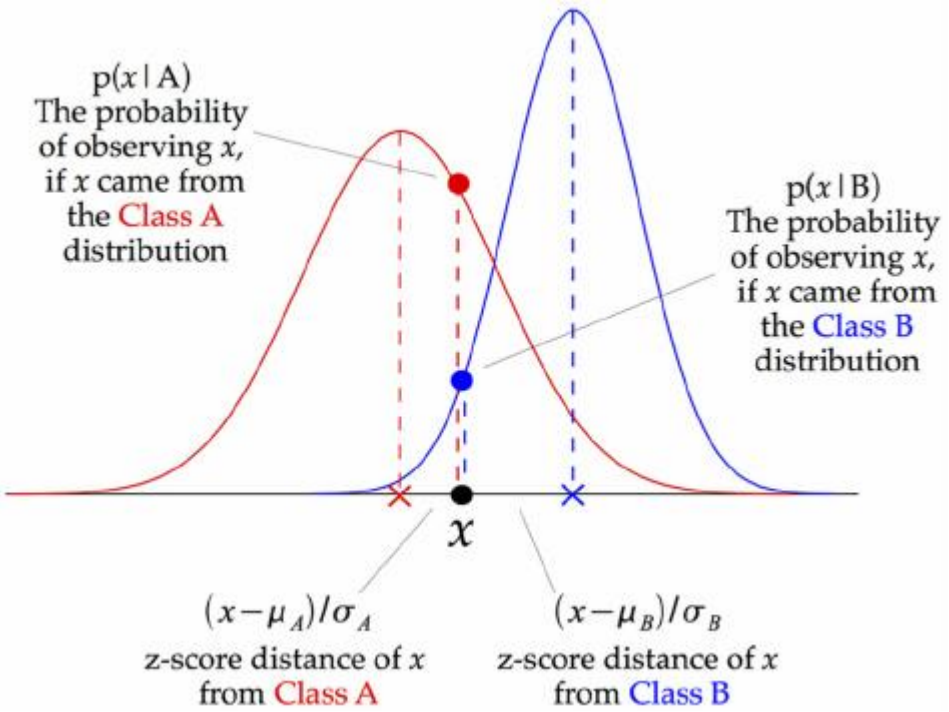

Figure 2.5 Gaussian Naive Bayes Illustration (Herbinet, 2018)

2.2.5 Multilayer Perceptron / Neural Networks

In a nutshell, MLPs (Multi-Layer Perceptron) are algorithms based on biological neural networks (Han et al., 2018; Hinton, 1989). Through the use of a graph of connected processing units, this model mimics the functioning of neurons. The NN is usually composed of three layers: an input layer, several or one hidden layers, and an output layer (S.-C. Wang, 2003). Each layer is connected by neurons. Weights are assigned to each connection between neurons. A weight value is assigned when the model is trained (Kavzoglu & Mather, 2003). During the training process, the objective is to minimize the error between predicted and actual values. (Kavzoglu & Mather, 2003). According to Abiodun et al. (2019) NNs and MLPs have shown promising results in recent years. This is due to the algorithm's properties that could solve nonlinear and coupled models (Fadlian et al., 2021; S.-C. Wang, 2003; Zheng, 2019)In addition to that, The use of this algorithm in classification and match prediction is also illustrated by numerous examples (Fanelli et al., 1993; Faris et al., 2016; Zheng, 2019).

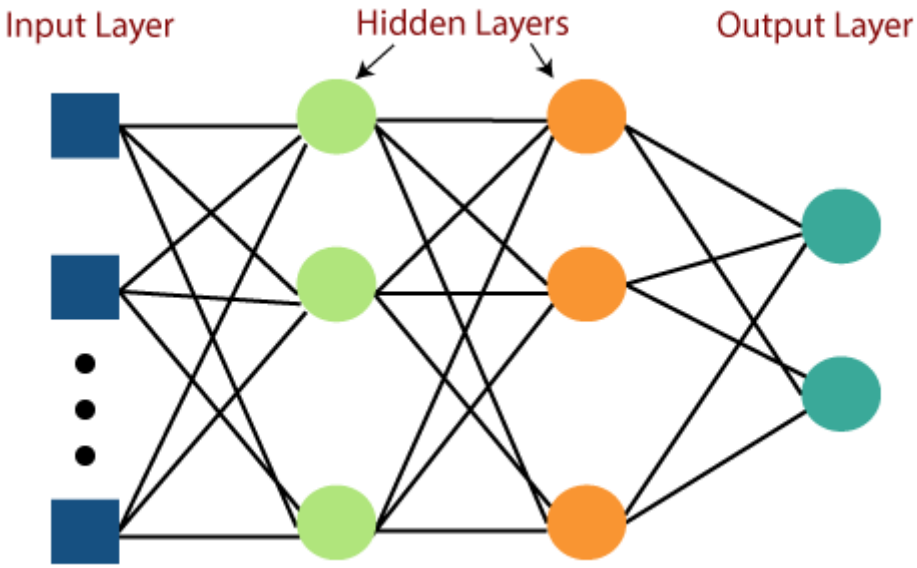

*Figure 2.6 MLP (*Multi-Layer Perceptron in TensorFlow - Javatpoint*, n.d.)*

There has been numerous machine learning tasks that can be solved with MLPs (S.-C. Wang, 2003). Not only for binary classification, MLP has been used for pattern recognition (Abiodun et al., 2019), Natural Language Processing (Alshemali & Kalita, 2020), and regression (Nghiep & Al, 2001). In addition to that, there are several types of MLPs which are suited to their specific usage (Choong & Lee, 2017; Ekman, 2021; S.-C. Wang, 2003; Zheng, 2019).

2.2.6 Graph Neural Networks (GNN)

In order to understand a GNN further, we need to take a closer look at the concept of a graph in computer science. A graph is a data structure that consists of two components: nodes and edges. A graph can be defined as $G = (V, E)$, where V is the representation of the set of nodes, and E are the edges between the nodes. The edges will be directed if there are any directional dependencies between the nodes. There are so many things that can be represented as a graph, such as social media networks and molecules (Menzli, 2021). A graph also can be represented in the form of an adjacency matrix with a dimension equal to $n\ x\ n$ where n is the number of nodes. If the nodes have features of their own (like a user profile), the node feature matrix is $n\ x\ f$ with f equal to the number of features (Sanchez-Lengeling et al., 2021).

However, when it comes to analysing graphs, there are many challenges for existing machine learning algorithms(Menzli, 2021). According to Menzli (2021) & Sanchez-Lengeling et al., (2021), This is due to the fact that Machine Learning tools and Deep Learning tools specialize in simple data types. Grid graphs with the same structure and size can be compared to images with the same structure and size. The text and speech are sequences, so they can be viewed as graphs of lines. In contrast, there are more complex graphs with no fixed form and unordered nodes whose neighbours may vary in number. As well as this, existing machine learning algorithms make the assumption that instances are independent of each other. The statement is false for graph data since each node is related to other nodes by a variety of links (Menzli, 2021).

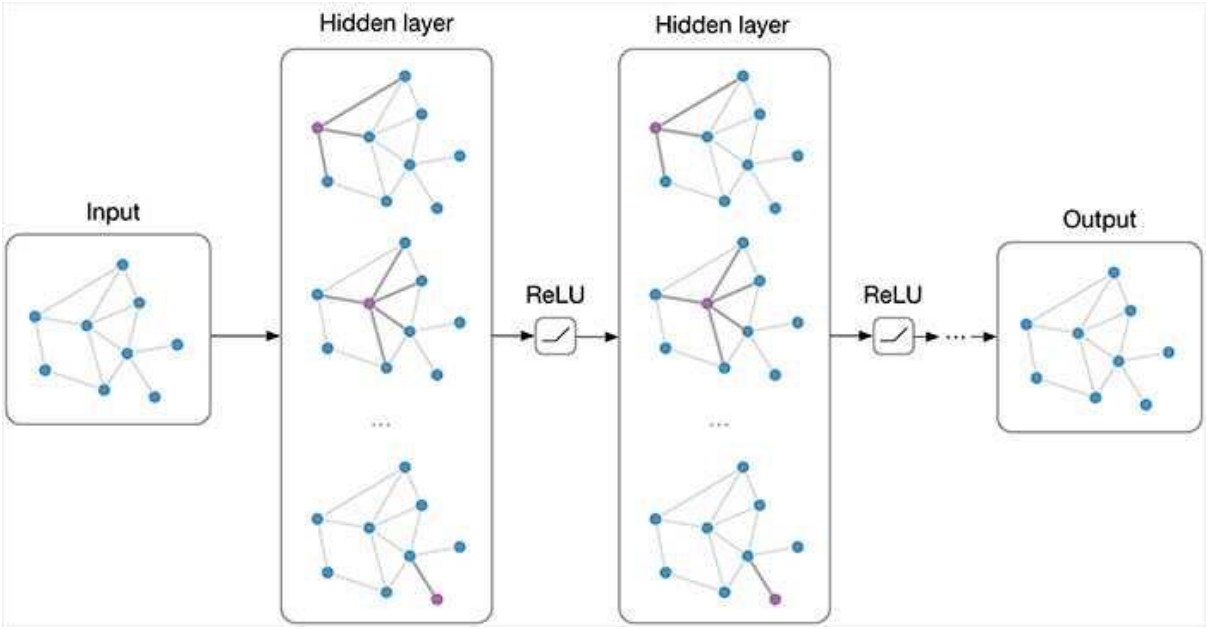

Figure 2.7 GNN (Karagiannakos, 2020)

Taking the previous graph description into account, a Graph Neural Network is a specialised class of deep learning designed to perform inference on graph data (Scarselli et al., 2009). GNNs are neural networks that can be applied directly to graphs and can predict nodes, edges, and graphs. In the case of graph data, GNNs can accomplish what Convolutional Neural Networks (CNNs) could not. However, the use of this algorithm requires additional work on the form of the data.

2.3 Feature Construction

The machine learning algorithm used for predicting the link between the nodes and binary classification needs features to be learned. However, with the evolving multi-media and profile forms on the platforms, the features included for the machine learning algorithm not only consist of the quantitative value but also include text from bios and profile pictures. Determining features that should be used for the learning model is one of the most critical steps in solving link prediction and binary classification. A study by Xia et al. (2014b) asserts that message sending and replying behaviour are highly affected by users' profile-based attributes. Behaviour in interaction, which reflects users' implicit preferences, provides insights regarding the users' actual preferences than their explicit preferences for their dating behaviour (Akehurst et al., 2012). In addition to that, topological features that describe the degree of incoming and outgoing links are used by researchers to represent the popularity and activity of an entity (Cai et al., 2012; Xia et al., 2014a). Based on the previous research, there are four pivotal categories of features, i.e., profile-based, topological, graph-based and interaction pattern features.

2.3.1 Profile-based Feature

(M. Zhang & Chen, 2018) analyse users' objective attractiveness based on their explicit features extracted from their profiles. In a user's profile, several types of information can be extracted, user demographics, users' physical attributes, ideal partner preferences, profile image, and bio or self-introduction article. However, to simplify the algorithm, this research only accounts for users' gender, type of relationship they are looking for, types of the gender they are looking for, and profile completeness, which reflects how the users engaged with the platform by giving the information regarding themselves.

In most experimental studies, physical attractiveness plays a significant role in matching, rather than the opposite, with daters preferring more attractive partners regardless of their own attractiveness (Curran & Lippold, 1975; Hitsch, Hortaçsu, & Ariely, 2010; Hitsch, Hortaçsu, Ariely, et al., 2010; Lee et al., 2008; Walster, 1970). In the context of online dating,(Hitsch, Hortaçsu, Ariely, et al., 2010) provided an innovative test of vertical preferences. According to their findings, e-mail sent to browsed profiles increases over time, regardless of the dater's own attractiveness (Hitsch, Hortaçsu, & Ariely, 2010), both for males and females daters.

According to *the matching hypothesis* (Walster et al., 1966), both men and women are strategic in their mate selection, seeking partners whose social desirability closely resembles theirs. The hypothesis is consistent with observed rates of marital homogamy, in which spouses share a wide variety of sociodemographic information and personal characteristics (Mare, 1991; Schwartz & Mare, 2012). Schwartz & Mare (2012) asserted that couples from the same sociodemographic class tend to have a more long-lasting relationship period and lower divorce rate.

In addition to sociodemographic and physical attractiveness, interest and hobbies also affect a romantic relationship. Lewandowski (2018) asserted that hobbies/interest is one of the main things that people consider when choosing partners. This claim was also strengthened by the previous

study by Johnson (1989) which argued that hobbies and interests are one of the variables associated with a relationship in an adult population.

2.3.2 Topological Feature

There have been researches on how attractiveness could possibly elevate the chance of people forming a romantic relationship (Asendorpf et al., 2011; M. Zhang & Chen, 2018). One of the ways that can be done to measure attractiveness is by inspecting the users' rate of getting a reply from their interactions (Nayak et al., 2010). Various prior studies were conducted to measure users' activities and popularity on social networks based on different types of links. However, as asserted by Zhang & Chen (2018), online dating social networks are different compared to other social networks. On social networking sites for online dating, users consider only their compatibility rather than others' activity levels. Therefore, we consider only the popularity factor to be our topological feature, which also relates to the attractiveness of the users. The topological link described by Zang et al. (2017) popularity values are divided into two categories. The first is the ratio of the replied contacts to the whole contacts. The second one is the rate of replying to contacts to the whole contacts a user receives. Although still using the same concept as Zang et al. (2017), we approach the problem differently following our dataset.

According to their findings, e-mail sent to browsed profiles increases over time, regardless of the dater's own attractiveness (Hitsch, Hortaçsu, & Ariely, 2010), both for males and females daters. However, the rate of sent messages is not the only factor that drives the popularity variable; the rate of incoming messages also needs to be considered in order to measure the popularity of an individual (Swani et al., 2017). These rates of receiving and sending the messages to other users could determine relationship satisfaction, according to Luo & Tuney (2015).

2.3.3 Interaction Pattern Feature

Research in relationship studies has long focused on how to reduce uncertainty and foster successful interactions at the very beginning of a relationship (Berger & Calabrese, 1975; Douglas, 1990). According to Furman & Shomaker (2008), there is an association between the patterns of interaction and the closeness of the adolescent romantic relationship. In addition, (Brand et al., 2012) and Snyder et al. (1977) asserted that men perform particular texting pats toward women, resulting in the women perceiving the man as more attractive. Snyder et al. (1977) also argued that the level of physical attractiveness would affect the interaction pattern between men and women. Both men and women tend to be more sociable and friendly when interacting with an attractive individual. Moreover, (Galliher et al., 2004) stated that a more significant number of positive interactions between the couple could increase the satisfaction in their relationship.

However, according to (Houser et al., 2008), the limited time and form of interaction could decrease the couples' judgement towards others on a dating platform. Neustaedter & Greenberg, (2012) also asserted that the lack of communication might result in a lower level of satisfaction between couples in a long-distance relationship (LDR). In addition, the harmony hypothesis by Gavin and Furman (1996) asserted that the positive interaction from one side of the party in the relationship is not enough to build a more satisfactory relationship; there has to be a balance of interaction between both parties.

2.3.4 Graph-based Features

Inspired by the recommender system, there are two ways we can extract people's preference features. The first one is by using the content-based method. Content-based preference features describe how users choose others based on others' content or explicit features acquired from their profiles. Using the same approach as Pizzato et al., (2006) and Zhang & Chen (2018) the

compatibility score of content-based preference features between two users is achieved by calculating the inner product of preference features and another user's profile features. However, due to the limited computing capability, only discrete features and text features are considered in the calculation. The second method is by collaborative filtering-based preference features. User preferences can be determined by whom they contact, as mentioned above. However, the use of these features requires a sufficient amount of data on the user's interest, hobby, and other variables that can be used to calculate content-based and collaborative filtering.

2.4 Social, Ethical, and Legal Implications

One of implementing a better match prediction is that this prediction method could potentially decrease the *overchoice* problem. As mentioned in the previous part, one of the negative sides of the dating app is that the platform suggests so many options for the users, which lead to the users' hardship in making of decision due to the overabundance of the options. By implementing a match prediction, the dating app could filter out the matches. However, sometimes finding the users the best match possible is not entirely in line with the platform's goal. Although Xia et al. (2014b) suggested that improving match outcomes could potentially increase the users' satisfaction towards the app, Wu et al. (2018) argued that finding the right person for a lasting relationship made less money for the dating apps compared to the casual slings. The argument is supported by the survey from GWI (2015), which stated that 54% of active dating app users are singles, while married and people in a relationship show less number compared to the prior category (30%).

In addition, although it is no direct legal impact of dating apps, a study by Benson (2021) shows that couples who are met online are six times more likely to be divorced compared to those who do not. With the increasing usage of dating apps/online dating platforms, the number of couples who will end up in a divorce situation will grow.

2.5 Section summary

After discussing the literature regarding the matching prediction in a dating app, several points can be concluded:

- The matching prediction can be approached with both graphical and tabular data.
- Various match prediction experiments have been conducted using combinations of machine learning algorithms.
- Four main kinds of features can be used as a predictor to determine the likelihood of a match in a dating app: profile-based features, topological features, graph-based features, and interaction pattern features.
- There are social, ethical, and legal implications of the use of the dating app, which could be affected by the result of this experiment.
- No experiment assesses the use of interaction pattern features alongside other types of features using various machine learning algorithms in predicting the match outcome.
- There is no experiment that combines the use of graph neural networks with various features to predict the likelihood of a link between the nodes.

3 Methodology

Taking the above literature and objectives into account, a specific sequenced methodology plan has been developed to achieve the objectives and aim of this research. At the initial stage, a kick-off meeting between students, supervisor, and Fluttr Limited representatives is conducted. The preliminary meeting is conducted to synchronise all stakeholders regarding the aim of the research, dataset requirement, non-disclosure agreement, and the expectation of both company and the

students. To this end, several meetings meeting with Fluttr Limited has been conducted to discuss the general dataset description, possible additional features/variable on the dataset, and the limitations of the dataset and the company. In parallel to this, a literature review regarding match prediction in an Online Dating Platform will be thoroughly studied. Hence, helpful case studies that address desired output, feature construction, deep learning algorithms architecture, and the evaluation matrices of the discussed topic can be identified.

The quantitative data analysis methods include exploratory data analysis, data cleaning and structuring, feature construction and selection, artificial neural network training, testing and cross-validation, and matrices evaluation. A descriptive statistical analysis will be conducted on the EDA phase to infer the normality and the distribution of the data. In addition to this, variance analysis will also be conducted to better understand the data spread. Furthermore, correlation analysis will be conducted to test the correlation between variables and the association between our target variable (match status) and other predictors. There are two types of correlation analysis that will be conducted; the first one is the Point-biserial correlation, to measure the correlation between a binary variable (target) and other continuous variables. The second one is the real-world data provided by the company does not come in an ideal form. Missing values and non-standard values are expected. The data cleaning and structuring phase needs to be conducted to overcome the shortage.

In accordance with the research objective to implement and evaluate several machine learning methods and a combination of features in predicting the match outcome (RO5 & RO6), this research will use the tabular data format. The use of the approach of tabular data format is due to its applicability with multiple machine learning algorithms (Shwartz-Ziv & Armon, 2022).

3.1 Ethics

It is assumed that this research is a low-risk project. A meeting with stakeholders has been conducted regarding the general review of the data. *Fluttr* Limited ensures that the data has been anonymised by their engineering and data team. A Non-Disclosure Agreement has been signed by the Fluttr Limited representative and me, as the student researcher. However, all the result of this research regarding the data, features, and log activities is still possible to be traced back and personalised by *Fluttr* Limited. The ethical aspect of this research is approved in the University Research Ethics Application System.

3.2 Experiment Setup

The experiment is conducted with Python within Anaconda Environment, Microsoft Excel, and MongoDB Compass. Initial data inspection is conducted with MongoDB compass; then the data are extracted using Python into CSV files. Then, the initial data preparation and data abstraction are conducted using Microsoft Excel and Python interchangeably, depending on the circumstances. For example, if there is a need for a quick visual approach, Microsoft Excel will be used. In other cases, Python is used for repetitive tasks such as configuring the users' pairs combinations. The version of Python is 3.8.13, and the version of Anaconda is 4.13. Several libraries are used to conduct the experiment. The main library for machine learning algorithms is Scikit Learn, which provides most of the machine learning algorithms (e.g., Logistic Regression, K Nearest Neighbour, SVM, Random Forest, Gaussian Naïve Bayes, and Multi-Layer Perceptron). In addition to Scikit Learn, the XGBoost library is also used for the XGBoost/Boosting classifier algorithm. Alongside machine learning methods, Scikit Learn also provides metric evaluation functionalities that will be used for our performance evaluation (e.g., MCC). Numpy and Pandas Library is also used as a library for

dataframe and numerical data handling. These libraries are widely used for data analytics, data science, and machine learning development (Whittle, 2021). In addition to that, Matplotlib and Seaborne are used for data visualisation. Python Stats package is used in this experiment for various statistical analyses like correlation tests.

3.3 Data Description

We use the secondary data provided by Fluttr Limited. The users' ids are anonymized; hence the research is classified as posing no risk. The data was given in the form of MongoDB and can be accessed with specific credentials (password). The database consists of 17 data tables, each representing a specific object related to the user or interaction between users. This study combines the documents of *matches, media, messages, reactions, users, userviews, and calls* to be able to extract required features and interaction patterns between the users. Each data table has its own primary key that identifies each entity; for instance, in the *users* table, there are 876 users that can be identified and can be differentiated between one another with a *userID.* Each user has their own *userID.* Each entity also possesses a unique identifier, regardless of its form (an object in real-life or an interaction). Hence, every call, reaction, likes, views, and uploaded document has its unique identifier in the form of id, which can be used to identify and differentiate between one another. In addition to the unique identifier of each entity (primary key), some of the table also have their foreign-key which is a reference to a unique identifier or a primary-key in another table. For example, in the table *matches,* there is a match between user A and user B, the match has its own unique identifier (*matchID*), and in addition to this, the table *matches* also provide us columns that enlist the unique identifier of the matching user (user A and user B). Hence, each match is associated with two users, which can be traced back to the *users* table. A UML diagram is provided as follows to better explain and visualise the relationship and the association between each table and entity.

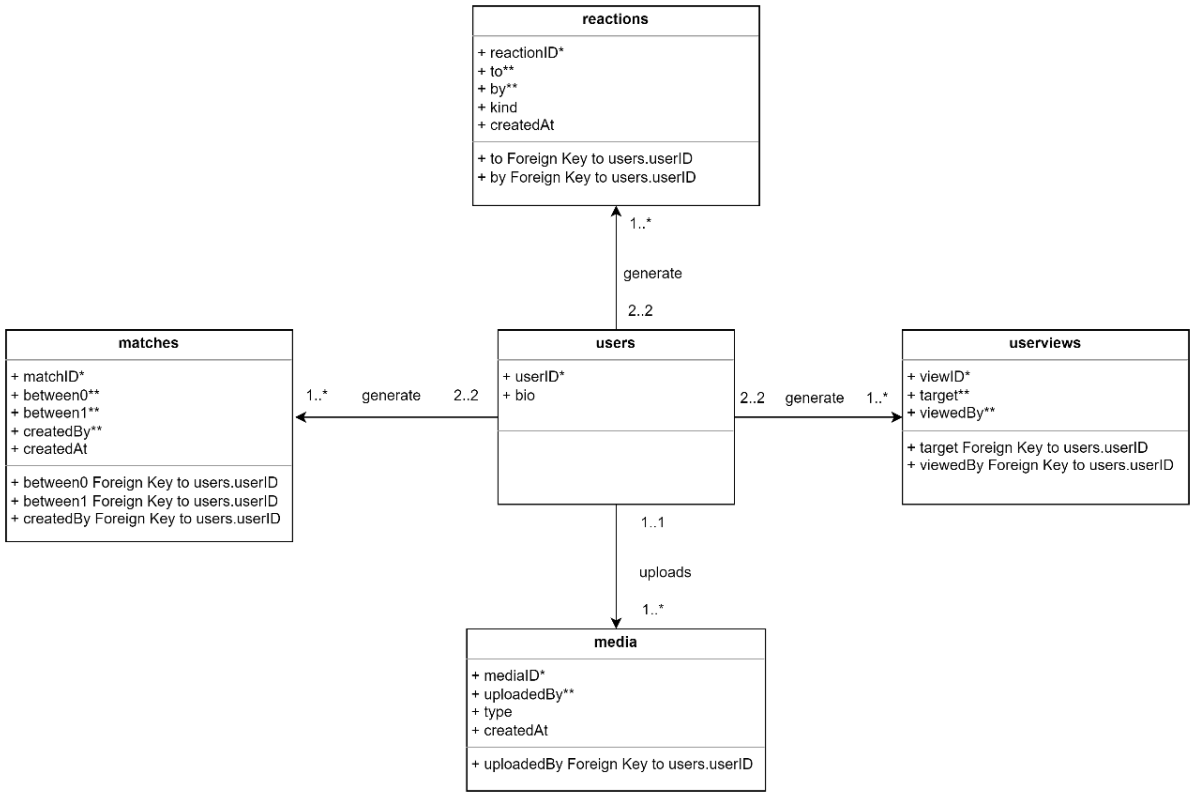

Figure 3.1 UML diagram of tables

Each box represents a table with the title of the box as the name of the table. The bullet points inside each box are the representation of a column name. column name with a star (*) is a primary key of the table, and column name with double stars (**) is a foreign key of the table. The arrow represents the relationship between the table with the number besides the table represent the cardinality of the table. For instance, the relationship between *users* table and *reactions* table is that users generate one or more reactions (hence the cardinality is 1..*), however, a reaction could only be generated by two interacting users (hence the cardinality is 2..2).

3.4 Experiment Steps

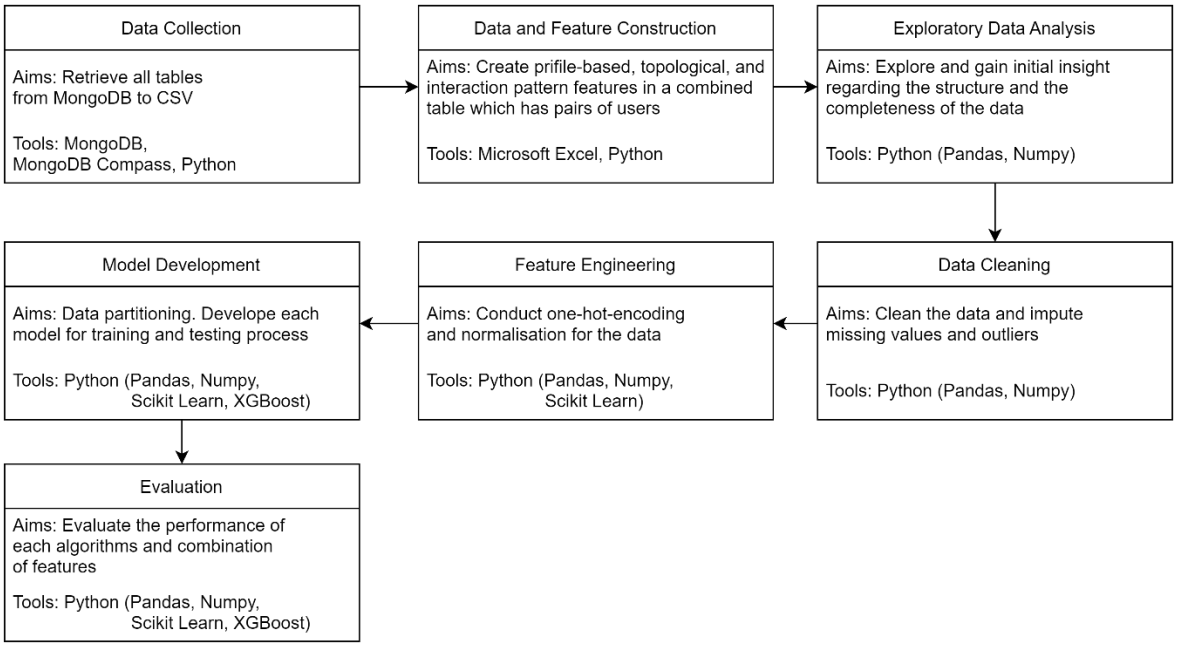

Figure 3.2 Experiment Steps

Before the experiment and analysis process, the data table needs to be combined into one comprehensive table containing features and variables mentioned in the previous section. The new combined table will consist of columns extracted from the original tables. The following diagram will demonstrate the process to better understand the combination table.

3.4.1 Feature Construction

Prior to the data construction in the main combination table, first, we need to construct our features. Due to the sparsity of the tables, we may not use every column of the tables. However, some variables are constructed from the existing column.

Profile-based features are extracted directly from user documents. However, due to the large amount of missing data (almost all columns missing 80-85% of values), we only use several items that describe the user's profile, such as gender, type of relationship they are looking, type of gender they are looking, and profile completeness. Gender and type of relationship preference can be

extracted from *users* table. Other variables are the composite variables that measure the completeness of several related features.

Table 3.1 Profile Based Features

Variable name	Description	Table of source
gender	Gender of the user	*users.gender*
lookinForGender	The type of gender the users preferred	*users.lookingForGender*
lookingForType	The type of relationship the users preferred	*users.lookingForGender*
bioCompleteness	The completeness of the users' bio. This is a composite variable that measures the completeness of the bio-related column of the users (e.g., bio, questions, my story)	*users*
physicalCompleteness	The completeness of the users' bio. This is a composite variable that measures the completeness of the physical-related column of the users (e.g., health, fitness, height, eye colour, weight)	*users*
demographyCompleteness	The completeness of the users' bio. This is a composite variable that measures the completeness of the demography-related column of the users (e.g., age, job)	*users*
interestCompleteness	The completeness of the users' bio. This is a composite variable that measures the completeness of the interest-related column of the users (e.g., hobbies,	*users*

Topological features are extracted by combining *reactions* and *userviews* documents. Each document provides the number of likes and views. In addition to this, the number of media uploaded by the users is also considered as the topological views, as this may result in the number of likes received by the users. By conducting the same approach as Zang et al. (2017)

Table 3.2 Topological Features

Variable name	Description	Table of source
reactionReceived	Number of likes received	*reaction.likes*
reactionGiven	Number of likes given	*users.lookingForGender*
mediaCount	Number of uploaded media	*media*
viewReceived	Number of views received	*usersviews*
viewGiven	Number of views given	*usersviews*

Interaction feature is gathered from userviews and reactions documents. However, the process of constructing this feature is different from the Topological feature. Interaction pattern features only account for the interaction conducted before the match. In order to understand better what kind of interaction can be conducted before the match, ideally, we need to analyse the app directly. However, due to several issues with the app's progress, we cannot examine the types of interaction directly from the app. Instead, we use the FAQ documents provided by Fluttr (Likes and Matches, 2021). According to the documents, there are several things that can be classified as interactions; however, our subject of interest is the interactions that can lead to a match. There are several interactions that are facilitated by the apps, such as views, likes, secret questions, and games. However, due to data insufficiency, the only interactions that can be used are *views* and *likes*.

Table 3.3 Interaction Pattern Features

Variable name	Description	Table of source
viewByUser	Number of views done before a match to the counterpart	*userviews*
matchType	Type of match (heterogenous, homogenous, etc.). This variable is obtained by joining the *match* and the *users* table	*match* *users.gender*

3.4.2 Data Construction

As mentioned in the previous section, predicting the match likelihood of two users can be considered as a binary classification with the target variable consisting of two values which indicate the outcome of the matching attempt, "ACCEPTED" and "NO MATCH". "ACCEPTED" means there is a match between the pairs, and "NO MATCH" means there is no match between the pairs. Hence, it is crucial to create a table consisting of pairs of users with the match outcome. In addition, this process aims to construct the data ready to be analysed in accordance with RO2. Each row in the combined table consists of a pair of two users with their respective topological and profile-based features. In addition to that, the interaction patterns between the user pair are also added to the rows. Lastly, a variable which indicates match success is added as the target variable for the rows. Although the match data is already provided in *the table, we still need the data indicating* the unsuccessful attempts of matches. The unsuccessful match attempts can be gathered from *reactions* table that consists of likes and the initial interaction between pair of users that can lead to a match. In addition to this, other interactions, such as views and likes, which indicates interaction pattern are also available in the form of user pairs.

First, all two possible combinations of the users (without repetition) are needed. To obtain this data, a python program which utilises the itertools package is built to list all possible two combinations of the 876 users. According to the formula $_nC_r = \frac{n!}{r!(n-r)!}$ In which $_nC_r$ is the number of combinations, n is total number of the set (number of users 876), and r number of choosing objects from the set (2). Based on the formula, we got 383250 number of possible two combinations of the users without repetition. Each pair will be given an identifier and a concatenate of the identifier of the two users in the pair. For example, pair combination number one consists of user A with unique id 1234 and user B with unique id 5678, then the unique identifier of the pair is 1234 – 5677. With this system of identifiers, each pair will not be the same as one another. Creating the unique id of each pair also can be used as a reference to lookup what kind of interaction is associated with the pairs. After this,

another unique identifier that consists of the combination of two users' unique id from every interaction was added to other tables. For instance, in table *reaction,* a like reaction is given by user A with unique id 1234 to user B with unique id 5678; a new unique identifier of the reaction is 1234-5678, although the like reaction already has a prior unique id. This unique id in another table can be used as a reference to look up the interaction between pair of users. After every two possible combinations of users has their unique id, and every interaction has its own unique id, which is built by merging the unique id of the two users, a complete table can be built by looking up the unique id of the pairs as the references. This process is conducted using Python and Microsoft excel interchangeably due to its complexity that requires both repetitive tasks (list the combinations) and visual cues for the tables. After these processes are completed, the data will form a new table which consists of pairs of users, their interaction, users' profile-based features, and users' topological features in every row. The approximate columns of the final combined table are listed with the table 3.4; however, due to the vast number of columns, only the name of the features is listed, not the specific column names.

Table 3.4 Combined table illustration

User0 profile-based features	User0 topological features	User1 profile-based features	User1 topological features	Interaction pattern features	Target/match outcome

The last part of the data construction is to cross the users so that the pair of userA and userB will later be added as userB and userA in order to balance the dataset. Figure 3.3 will explain how the crossing works.

UserID0	UserID1	matchID
1234	5678	12345678

UserID0	UserID1	matchID
1234	5678	12345678
5678	1234	56781234

Figure 3.3 Crossing process

Another approach using graph data is not implemented on this experiment due to its complexity. Graph-based approaches require more work on the data, because the data available are mostly tabular (Borisov et al., 2022), which must be converted to a graph format. In addition, this type of data can only be analysed using a limited number of algorithms, regardless of the outcome of the prediction (Shwartz-Ziv & Armon, 2022). Further, graph data can limit the types of features that can be analysed, as the matching prediction will not only take into account the interactions and links between nodes/users, but will also consider profile-based and topological features associated with each individual node.

3.4.3 Exploratory Data Analysis

In accordance with RO4 initial EDA is conducted to better understand the data and to get an initial insight into the data and as a reference of what kind of analysis needs to be conducted in the following steps (Jebb et al., 2017). First of all, we analyse the distribution of the data to gain insight into the normality and the balance of the data. The next step is to choose between a parametric test or a non-parametric test to test the association/relationship between the predictors and the target.

Based on the initial analysis of the form of the data, there will be two kinds of association/correlation tests conducted on the dataset. To measure the association between the categorical variables and our target variable, we use the chi-square test, a non-parametric test suitable for our data. To test the correlation of the continuous predictors and our target variable, we use point-biserial correlation, a non-parametric method to measure the correlation between continuous variables and a dichotomous/binary variable.

3.4.4 Data Cleaning

Given the complete data of the new table, a set of data cleaning processes and feature engineering needs to be conducted to ensure the quality of the data. First of all, we filter out the user pairs that consist of at least one user that is not an active user. This process ensures that no inactive users in involved in the assessment. Inactive users may result in biased performance and can add to the imbalance of the dataset (Z. Wang et al., 2021). In addition to this, the column filtration process also needs to be conducted to ensure that there is no identifier column in the table. The identifier column is not essential in this experiment since the study is not a time-series-related analysis (Shumway & Stoffer, 2010). In addition to that, maintaining the identifiers in the table will result in the biased performance of the algorithms (Z. Wang et al., 2021).

However, some of the data cleaning processes have been conducted before the exploratory data analysis step. This is due to the EDA process not only consisting of the descriptive task for the data but also statistical analysis regarding the features, which can result in biased statistical analysis results (Kwak & Kim, 2017). Hence, in the practicalities of this experiment, the data cleaning and the EDA is conducted back and forth to ensure that the data is ready to be processed by machine learning algorithm.

3.4.5 Feature Engineering

Implementing one-hot-encoding is essential for categorical data, especially when using linear or gradient-based models, which cannot process non-numerical data (Choong & Lee, 2017). Although there will be an effect of implementing one-hot-encoding to the algorithm due to the increased number of dimensions, the negative effect is minimum. The goal of one hot encoding is to convert categorical data into separate columns. For instance, there is a column named gender, with the value of 'man' and 'woman'; by using one-hot-encoding, we convert the column into two separate columns with the name 'men' and 'woman'. The rows that previously had the value of 'men' on the sex column will be valued as 1 in the new 'man' column and 0 in the new 'women' column.

userID	Gender Type
1	Men
2	Women
3	Non-Binary
4	Men

One-Hot-Encoding

userID	Gender Type (Men)	Gender Type (Woman)	Gender Type (Non-Binary)
1	1	0	0
2	0	1	0
3	0	0	1
4	1	0	0

Figure 3.4 One-hot-encoding illustration

In addition to one-hot-encoding, a normalisation step was also conducted on the data. As such, this process is vital to preserving the significance of each variable. That way, all variables will have the same distance between 0 and 1 for their continuous variables (Singh & Singh, 2020). The normalization is conducted by the following formula:

$$X_{normalised} = \frac{X - X_{minimum}}{X_{minimum} - X_{maximum}}$$

3.4.6 Model Development

To evaluate the model's performance better, the data is partitioned into trains and tested before being processed with the model (Steyerberg & Harrell, 2016). The importance of not using test data during the training phase will result in an adequate indicator of the algorithm's performance. Although there is no definitive of which fold is the most effective one, we use Marcot & Hanea's (2021) recommendation of k-fold cross-validation, which uses k = 5 in their study. Additionally, we use Stratified Sampling based on the matching result or target variable. (Shahrokh Esfahani & Dougherty, 2014) claim that this method ensures equal distribution of target variables between training and testing sets, thus improving prediction results.

There are seven algorithms that are tested in this experiment. Each algorithm has its own specification and parameters which determine how it works. All seven algorithms in this experiment used the default parameter provided by the packages. Although it is true that choosing the proper parameters often results in a better performance of the algorithm (Lavesson & Davidsson, 2006; Mantovani et al., 2015; Probst et al., 2018), the most recent study by Weerts et al. (2020) there are cases that using default parameters provided by the package results in a more efficient system. The reason is that most of the default parameters have been configured for a more generic and wide range of problems in machine learning. In addition, using the package's default parameter setting ensures that every algorithm and the combinations of features will be equally assessed.

The logistic Regression algorithm train-test part is conducted using Scikit Learn linear_model.LogisticRegression. The algorithm is conducted using default parameters from the package. The penalty parameter which dictates the norm of the penalty is *l2* which uses the sum of the squares of the parameters and Ridge Regression. 1e-4 is used as the *tol* parameter for the stopping threshold. C or the inverse of regularisation strength is 1; the smaller values of this parameter specify stronger regularisation. *Class_weight* parameter uses *None* value; hence, there is no adjustment on the weight of the target. *Lbfgs* is chosen as the solver of the algorithm due to its association with *l2* penalty parameter. The number of maximum iterations for the solver to converge or *max_iter* is 100.

The Random Forest algorithm is built using the Scikit Learn ensemble.RandomForestClassifier functionality of the package. This algorithm also uses the default setting of the parameters provided by the package. 100 is chosen as the *n_estimators,* which state the number of trees in the forest. The algorithm uses *gini* as the splitting criterion. According to Tangirala (2020), there is no significant difference between using gini, entropy and information gain for binary classification problems in a random forest algorithm. There is no maximum depth of the tree used in this algorithm. 2 is chosen as the minimum number of samples required for splitting in the node. *Class_weight* parameter uses *None* value; hence, there is no adjustment on the weight of the target. The algorithm uses bootstrapping method when building the trees. The model does not use a warm start when conducting the training process. This algorithm does not use any pruning method.

The K Nearest Neighbour uses functionality from Scikit Learn neighbors.KNeighborsClassifier. This algorithm uses five as the number of the nearest neighbours in the queries. *Uniform* setting is used for the *weights* parameter; hence, all points in each neighbourhood are weighted with the same weight. This algorithm uses *auto* setting as the method to compute the nearest neighbours; hence the algorithm will automatically decide the most appropriate computing method for the algorithm out of three other computation methods (*ball_tree, kd_tree, brute*). The *p* parameter, which states the power parameter of the Minkowski metric, is 2; this equals to the usage of the Euclidean distance measurement method to measure the distance between points in the neighbourhood.

The Support Vector Machine algorithm uses functionality from Scikit Learn svm.SVC. This algorithm uses one as the default *C* parameter, which is equal to the inverse of regularisation strength; the smaller values of this parameter specify stronger regularisation. The algorithm uses *RBF* as its default kernel type to pre-compute the kernel matrix from the data. *Degree* parameter, which states the degree of the polynomial kernel function, is 3. This algorithm uses a shrinking heuristic. In addition to that, there is no limit for the maximum number of iterations performed by the algorithm; hence, the *max_iter* parameter is -1.

Gaussian Naïve Bayes uses naïve_bayes.GaussianNB functionality from Scikit Learn Package. The *priors* parameter is not specified by the default setting; hence, there is no prior adjustment probabilities of the class. The default value for *var_smoothing* parameter is 1e-9, which dictates the portion of the most significant variance of all variables that will be added for the stability of the calculation.

The Multilayer Perceptron uses functionality from Scikit Learn neural_network.MLPClassifier. The MLP uses the default parameter setting from the platform. As 100 is chosen as the number of hidden layers, *relu* or rectified linear unit is chosen as the activation function of each node in the network. This experiment uses a stochastic gradient-based optimizer or *adam* as the solver. 0.0001 is chosen by default as the *alpha* or learning rate of the model. In addition, the learning rate of the model will be constant throughout the training process.

This experiment uses XGBoost as the example of boosting algorithm. Unlike other methods that use the Scikit Learn package, the XGBoost algorithm uses the XGBoost package. There are three parameters that can be set to perform the algorithm; there are General Parameters related to the booster used to boost; we can choose either a tree or linear model. The second is the Booster Parameters which depend on the type of boosting that is used. The third one is the Learning Task Parameters which dictate the learning scenario. This experiment uses *gbtree* as the value of *booster* parameter; hence the XGBoost will use the ensemble of trees instead of linear models. The *value for validate_parameters* is set by default to *True*, which dictates that XGBoost will perform input validation of the parameters to check the usage of the parameter. This experiment use the maximum number of *nthread*; this parameter does not directly affect the accuracy performance of the algorithm, instead affecting the speed and the effectivity of the training process by dictating the number of computing threads that will be used by the algorithm. 0.3 is the default learning rate of the algorithm. According to Attoh-Okine (1999), learning rate tuning is important since a higher learning rate can cause the algorithm to overfit, and a higher learning rate causes the model to be stuck in local minima/maxima. To prevent this issue, we set the *eta* parameter to 0.3, which dictates the size shrinkage of the learning rate so that the learning rate may adapt according to the number of iterations. This algorithm has *max_depth* of 6, which dictates the maximum depth of the trees.

As mentioned in the previous section, this experiment will use the tabular form of the data, hence, graph-neural-networks method cannot be used. However, further exploration might be conducted upon the graph-based data which is suitable for the experimentation using graph-neural-networks.

3.4.7 Model Implementation

Derived from the objective of the study mentioned in the previous section, there are mainly two kinds of experiment that will be conducted in this study. The first one is to test the performance of the algorithms for link/match prediction. The second one is to compare the effect of each feature to the performance of the link/matching prediction.

In order to complete the test on the algorithms and the combinations of features (RO5 & RO6), the experiment will run through all seven algorithms (Logistic Regression, Random Forest, K-Nearest Neighbour, Support Vector Machine, Gaussian Naïve Bayes, MLP, and XGBoost), with several combinations of features. Each algorithm will run with these combinations of features; interaction pattern only, profile-based only, topological only, topological + interaction pattern, topological + profile-based, profile-based + interaction pattern, interaction pattern + topological + profile-based. To maximise the efficiency of the experiment, a program which loops the combination of the features, and the algorithm is built.

3.4.8 Evaluation

In order to evaluate the performance of each algorithm, we use several metrics. Our algorithm is described using a confusion matrix since it can generally indicate its performance (Canbek et al., 2017). The areas under the curve and the receiver operating characteristic (ROC) are fairly popular and reliable (Jin Huang & Ling, 2005), but recent research has indicated that the AUC can be misleading and sensitive to the classes (Lobo et al., 2008; Tharwat, 2021). In addition to this, Accuracy is considered to be unreliable on the imbalance dataset. As asserted (Mortaz, 2020), accuracy places more weight on the common classes than on rare classes, which makes it difficult for a classifier to perform well on the rare classes. Thus, this study will use MCC as the most robust classification method because it scored high on several criteria (e.g., Base measure correlations, Imbalance uncorrelations, Distinctness, Output smoothness, Monotonicity, Consistency, Universal Discriminancy) (Chicco et al., 2021; Chicco & Jurman, 2020).

To classify the strength of the MCC score, we use a method by LaMorte (2021) which classify the MCC scores as follows:

Table 3.5 Classification of MCC scores (LaMorte, 2021)

MCC Score	Classification
>= 0.6	Strong positive correlation
0.4 – 0.59	Moderate positive correlation
0.2 – 0.39	Weak positive correlation
-1.9 – 1.9	No correlation
(-0.2) – (-0.39)	Weak negative correlation
(-0.4) – (-0.59)	Moderate negative correlation
<= (-0.6)	Strong negative correlation

3.5 Section Summary

- The tabular data format is used due to its simplicity and compatibility with the algorithms
- The main table used for the experiment consists of pairs of users with their profile-based features and topological features. In addition, interaction patterns between users in pairs are also included in the table.
- The data cleaning process was mainly conducted to filter out inactive users and faulty pairs from the main table
- The feature engineering process is conducted to normalise the data and to convert categorical data to numerical data using one-hot-encoding
- The EDA is conducted to gain initial insight about the data
- In total, there are 49 experiments from seven machine learning algorithms and seven combinations of features to find the best performing algorithm and combination of features.

4 Results

This section will show the result of the EDA and the main experiment. The EDA part will discuss the result for the balance of the dataset, as well as the distribution of categorical data. In addition to that chi-square test and point-biserial correlation result also will be presented. The result of the algorithm and features combination experiment will also be presented. Lastly the summary of the section will discuss the key findings from the results.

4.1 Exploratory Data Analysis

Initially, there were 876 users with 383250 combinations of pairs. However, after filtering the users and the pairwise combination based on the user's activity and the availability of adequate interaction between the user pairwise, the final data that will be processed is 259 users with 1092 pairwise. To further balance the dataset, another process of crossing the users' (user0 to user1, user1 to user0) is conducted. This resulted in 518 users with 2184 pairwise.

Table 4.1 Gender distribution of unique users

Gender Type	Count	Percentage
Gender Neutral	1	0.39%
Man	174	67.18&
Woman	84	32.43%
Total	259	

Table 4.2 MatchType distribution

Match type	Count	Percentage
Man-Man	56	2.56%
Man-Woman	2020	92.49%
Woman-GenderNeutral	2	0.09%
Woman-Woman	106	4.58%
Total	2184	

Based on the EDA, we have an extremely imbalanced dataset. The ratio of 'ACCEPTED' and 'NO MATCH' is more than 1:10. The number of 'ACCEPTED' is 174, and the number of 'NO MATCH' is 2010.

Table 4.3 Match outcome distribution

Match outcome	Count	Percentage
ACCEPTED	174	7.97%
NO MATCH	2010	92.03%

Despite the fact that the dating platform in our study is for both heterosexual and homosexual users, the ratio between so called "straight couples" and other is imbalance, with the number of woman-men pairwise outnumbered other to more than 90%.

The Chi-square test shows that there is association between the desired type of gender and the desired type of relationship with the. However, gender is not associated with the match outcome.

Table 4.4 Chi-square result

Variable Name	P-value	Significance
matchType	0.000	Significant association between match outcome and matchType
gender0	0.433	No significant association between match outcome and gender0
lookingForGender0	0.000	Significant association between match outcome and lookingForGender0
lookingForType0	0.000	Significant association between match outcome and lookingForType0
Gender1	0.433	No significant association between match outcome and gender1
lookingForGender1	0.000	Significant association between match outcome and lookingForGender1
lookingForType1	0.000	Significant association between match outcome and lookingForType1

Based on the point-biserial correlation test, there is an association between the target variable and our numerical data. Interestingly, all numerical data has a p value < 0.05, which indicates that there are association between the match outcome and the variables. The table below shows the top six variables.

Table 4.5 Top six point-biserial correlation result

Variable Name	R Score	R Squared	P Value	Significance
viewGiven1	-0.341735832	0.116783379	7.19E-61	significant association with target (reject H0)
viewGiven0	-0.341735832	0.116783379	7.19E-61	significant association with target (reject H0)
mediaCount0	-0.256359765	0.065720329	4.08E-34	significant association with target (reject H0)
mediaCount1	-0.256359765	0.065720329	4.08E-34	significant association (reject H0)
viewReceived0	-0.24805298	0.061530281	5.56E-32	significant association with target (reject H0)
viewReceived1	-0.24805298	0.061530281	5.56E-32	significant association with target (reject H0)

4.2 ML Algorithms and Feature Combinations

Following are the results of MCC for each model on different feature differentiations. Feature differentiations are based on the discussion in the previous section. To simplify the presentation of data, the name of the features is shortened as follows; Interaction Pattern -> IP, Topological -> TP, and Profile-Based -> PB.

Table 4.6 Experimet Results

	Features combination							
Algorithm	IP + TP + PB	IP	PB	TP	TP + IP	TP + PB	PB + IP	Average
LogReg	0.835	0.230	0.425	0.681	0.681	0.617	0.599	0.581
RF	0.904	0.856	0.333	0.703	0.706	0.686	0.640	0.69
KNN	0.618	0.873	0.419	0.640	0.640	0.595	0.572	0.622
SVM	0.742	0.533	0.285	0.616	0.616	0.593	0.569	0.565
GNB	0.511	0.372	0.416	0.473	0.473	0.477	0.562	0.469

MLP	0.886	0.709	0.52	0.668	0.668	0.645	0.764	0.695
XGB	0.937	0.838	0.439	0.687	0.687	0.670	0.904	0.737
Average	0.776	0.630	0.406	0.639	0.638	0.612	0.658	

It can be seen from Table 1. which features and feature combinations resulting the best overall MCC based on which feature/s are included in the experiment. Generally, Model that utilizes Topological feature has hinger MCC compared to other feature differentiation. In addition, XGBoost algorithm performed better compared to other algorithm based on the average MCC score obtained from every combination of features. Overall, the highest score is obtained by combining XGBoost algorithm with Interaction Pattern, Topological, and Profile-based features (TP+PB+IP) all together. While the lowest score is obtained by combining Logistic Regression algorithm with Interaction Pattern feature.

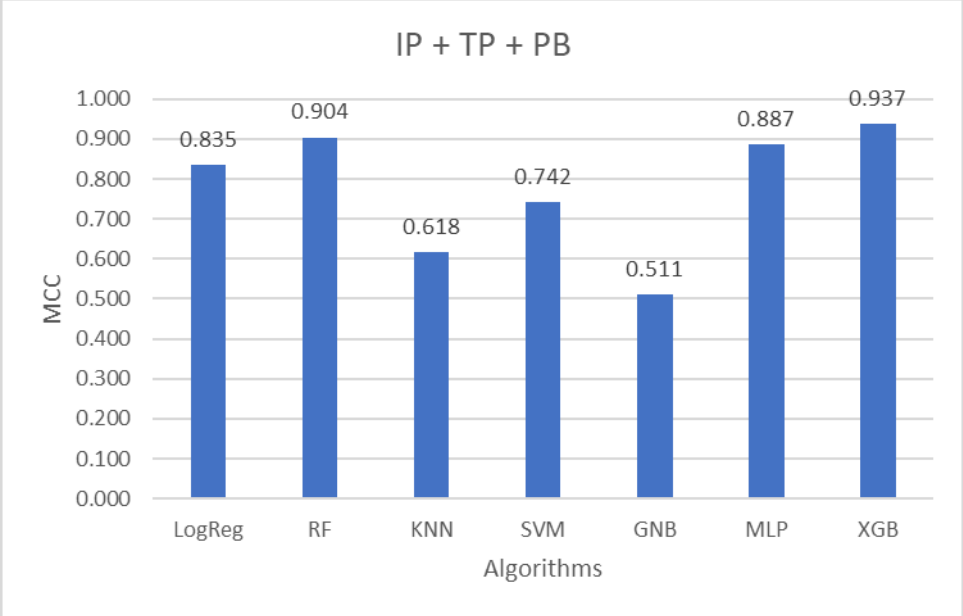

Figure 4.1 MCC Scores of IP+TP+PB

Based on the result of Table 1. The highest average score of features is obtained by combining all features (IP + TP + PB). Taking a deeper look into that, Figure 4.1 shows that amongst all experiments conducted with all features, XGBoost obtained the highest MCC score with 0.973. Random Forest, MLP, Logistic Regression and SVM placed in the second, third, fourth, and fifth with MCC scores 0.904, 0.88, 0.835, and 0.742, respectively. All results from XGBoost, MLP, Random Forest, Logistic Regression, and SVM are considered to have a strong Matthew's correlation coefficient score according to (LaMorte, 2021) due to their score being above 0.7.

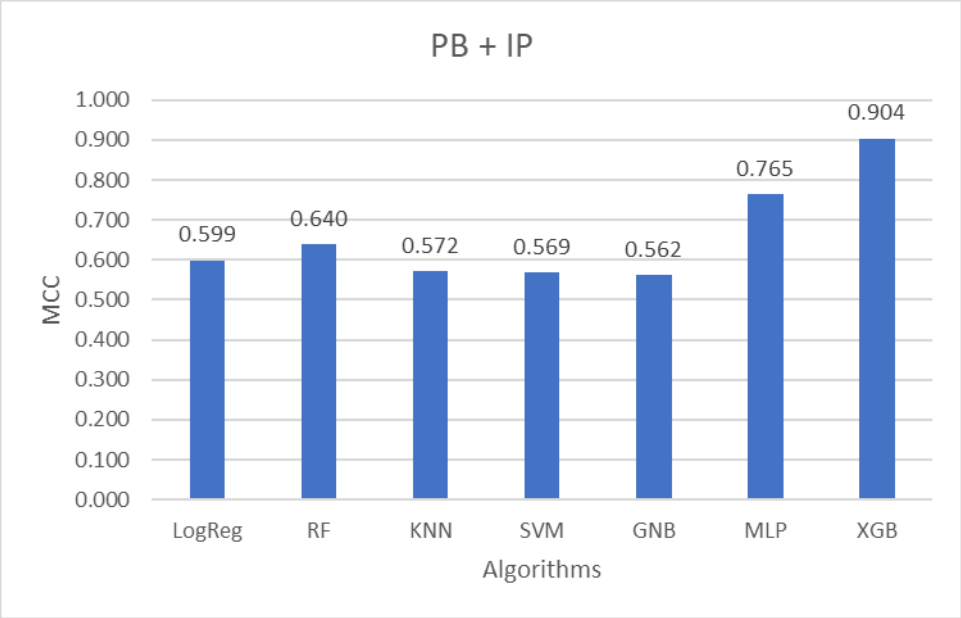

Figure 4.2 MCC Scores PB+IP

The second highest combination of features by the average score of algorithms' MCC is the combination of Profile-Based and Interaction Pattern Features (PB + IP). Figure 4.2. shows that XGBoost, again, ranked the highest compared to other algorithms with 0.904 MCC score. MLP is in second place with 0.765; Random Forest is in third place with a 0.64 MCC score. However, the PB+IP result only shows XGBoost, Random Forest, and MLP with a strong Matthew's correlation coefficient score. Other results show only a moderate MCC score.

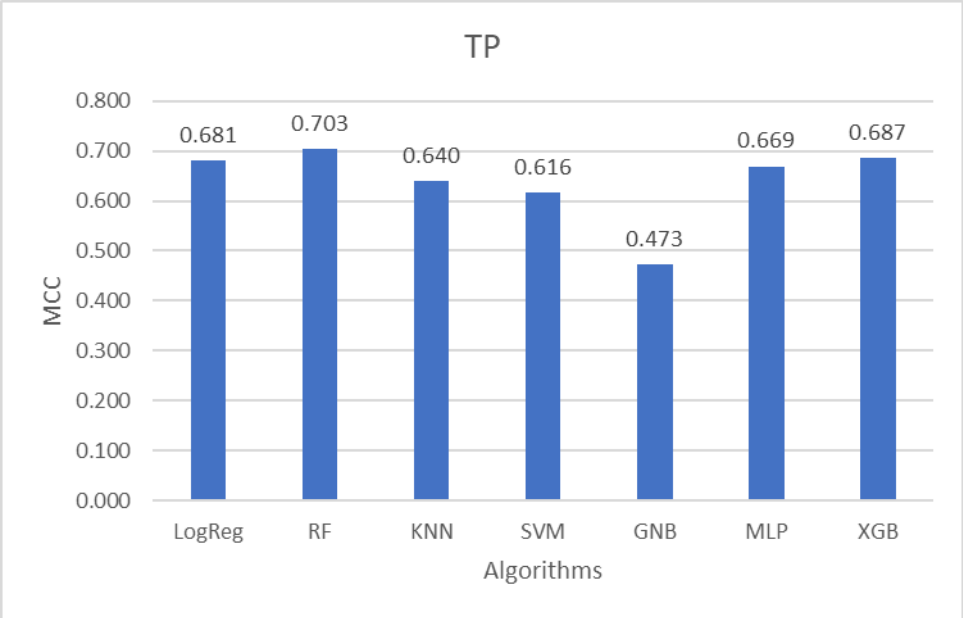

Figure 4.3 MCC Scores of TP

The third highest combination of features by the average score of algorithms' MCC is Topological Features only (TP) shown by figure 4.3. Unlike the previous results where XGBoost always performed better compared to other algorithms, the highest performing algorithm in the topological feature setting is Random Forest (MCC: 0.703). Random Forest also is the only algorithm with a strong MCC score; other algorithms are considered as having moderate correlation, with Gaussian Naïve Bayes performing the worst with only 0.473 MCC score, which is considered as weak correlation.

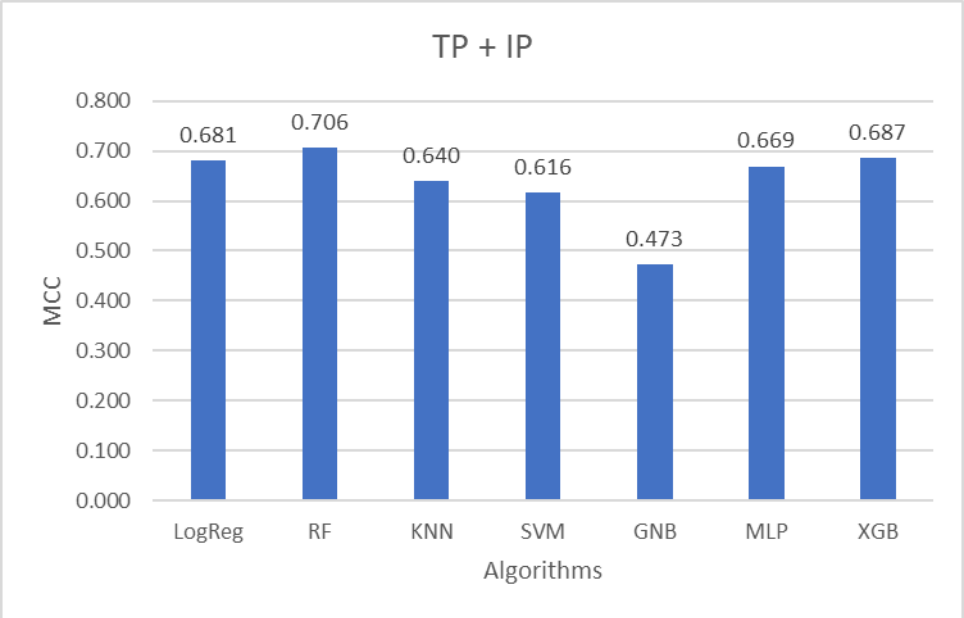

Figure 4.4 MCC Scores of TP+IP

The fourth highest combination of features by the average score of algorithms' MCC is the combination of Topological and Interaction Pattern Features (TP + IP) shown by figure 4.5. Again, Random Forest ranked the highest compared to other algorithms with a 0.706 MCC score. Random Forest is the only algorithm with a strong MCC score; other algorithms are considered to have moderate correlations, with Gaussian Naïve Bayes performing the worst with only 0.473 MCC score, which is considered weak correlation.

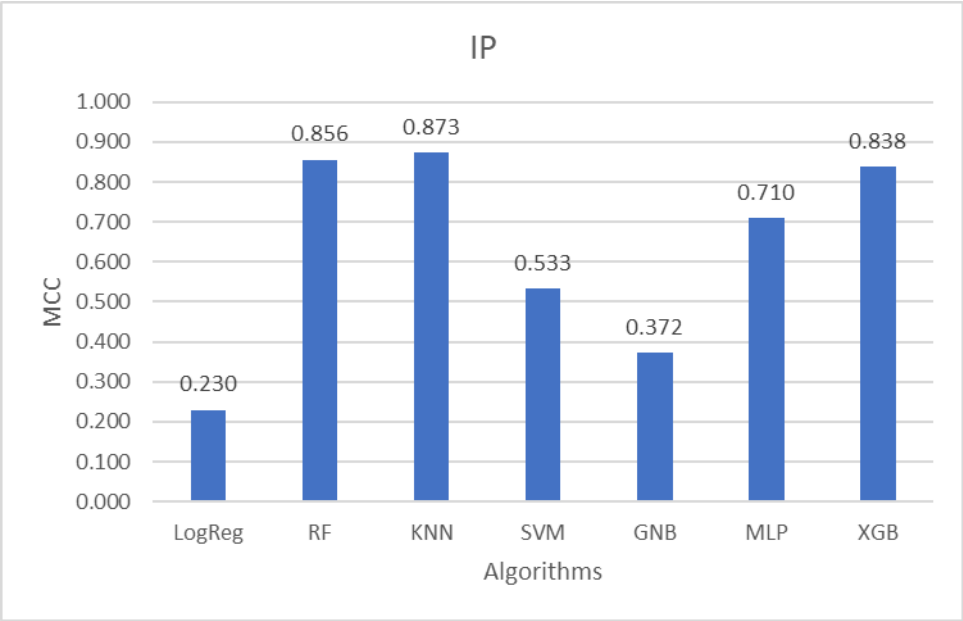

Figure 4.5 MCC Scores of IP

The models which only used Interaction Pattern Features (IP) ranked fifth based on the average MCC score. Figure 4.6 shows that K Nearest Neighbour (KNN) ranked the highest compared to other algorithms with a 0.873 MCC score. Followed by Random Forest and XGBoost with 0.856 and 0.838 MCC scores, respectively. KNN, Random Forest, MLP, and XGBoost obtained a strong Matthew's Correlation Coefficient score. SVM ranked fourth (MCC: 0.533) with a moderate MCC score. Lastly, Logistic Regression and Gaussian Naïve Bayes ranked last with weak MCC scores (0.23 & 0.372).

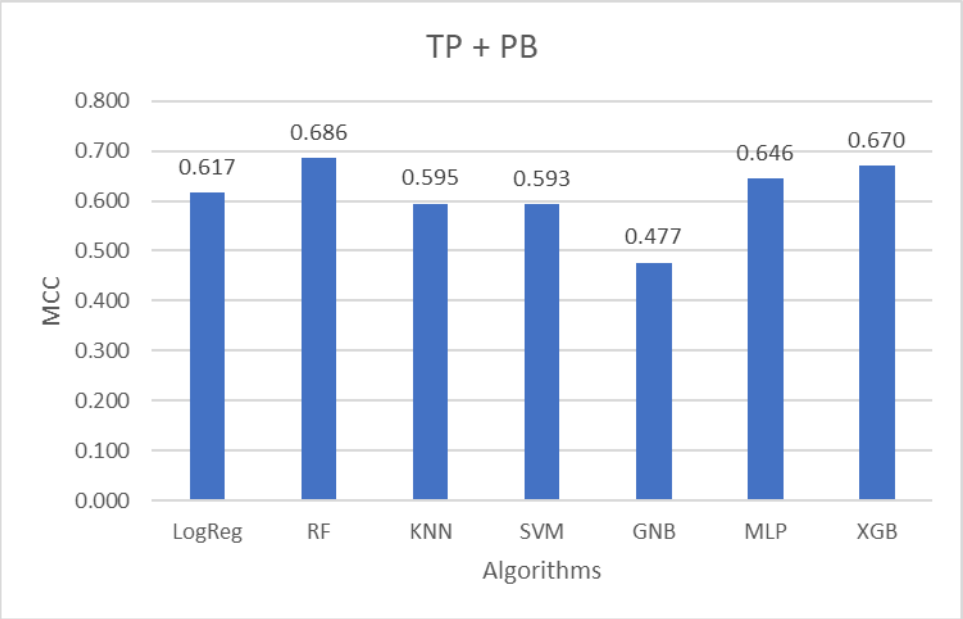

Figure 4.6 MCC Scores TP+PB

The combination of Topological and Profile-based features (TP + PB) ranked sixth based on the average MCC score of algorithms. As shown by Figure 4.6, Random Forest ranked the highest, followed by XGBoost, MLP, and Logistic Regression, with MCC scores of 0.686, 0.67, 0.646, and 0.617, respectively. In the fourth and fifth place, there are KNN and SVM with MCC slight difference in MCC scores (0.595 and 0.593). The last place is filled by Gaussian Naïve Bayes with a weak MCC score (0.477).

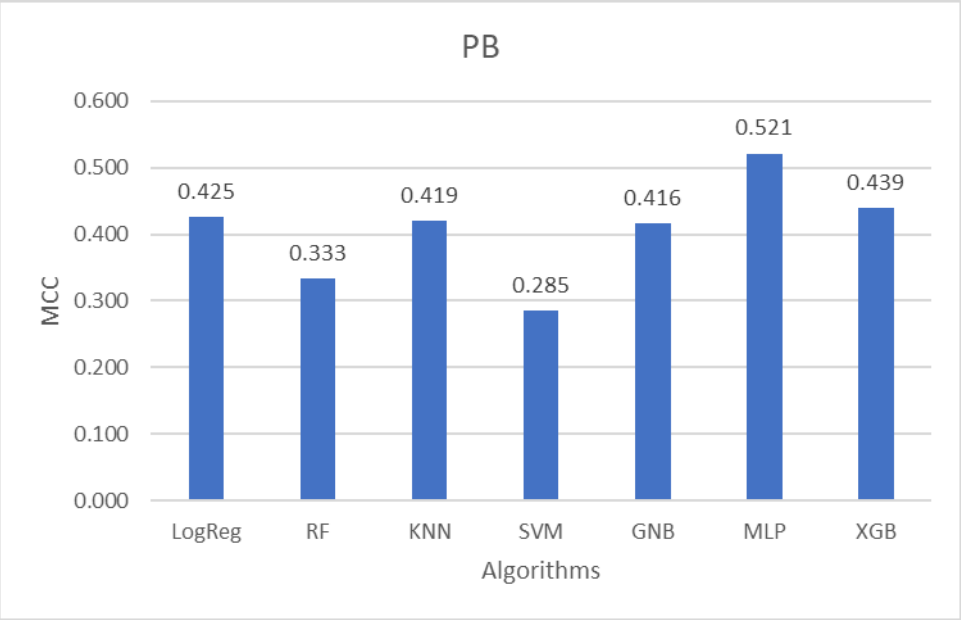

Figure 4.7 MCC Scores of PB

The use of the Profile-based (PB) only features ranked the last based on the average MCC score of the algorithms. As shown by Figure 4.7, the usage of profile-based only features provides weak MCC scores for all algorithms except MLP. The highest performing algorithm is MLP, with a moderate MCC score (MCC: 0.521).

4.3 Section Summary

- The experiment is conducted on a highly imbalanced dataset with the ratio of "ACCEPTED" and "NO MATCH" more than 1:10.
- All numerical variables have a significant association with the target variable.
- All categorical variables except for gender have a significant association with the target variable.
- Based on the average MCC score, the experiment which combines all TP, PB, and IP features obtained the highest MCC score (0.776).
- Based on the average MCC score, the experiment conducted with XGBoost obtained the highest MCC score (0.737).

5 Discussion and Conclusion

5.1 Result Analysis

Overall, XGBoost has the highest average of MCC compared to other algorithms. Random Forest and MLP come second and third place. Although there is evidence that random forest performed better compared to XGBoost (e.g., Kabiraj et al. (2020)), XGBoost almost always performed better compared to the latter in a classification task. According to Chen & Guestrin (2016) and (Gupta (2022), there are several reasons why XGBoost, a boosting algorithm is, performed better compared to Random Forest, a bagging algorithm. The first reason is attributed to XGBoost's approach to direct pruning before during the actual modelling steps. XGBoost uses the difference of 'similarity score' between the parent and the children node of the trees. In this case, overfitting could be overcome by just stopping the construction of the tree to a greater depth if the difference between the mentioned nodes is minimal. On the other hand, If the majority of trees in the random forest have the similar sample, the Random Forest might overfit the data, thus reducing the model's performance.

Although XGBoost has the highest average MCC score compared to other algorithms, there are cases where Random Forest got a higher score compared to XGBoost. For instance, although XGBoost got the highest MCC score in the experiment, utilising IP + TP + PB features, PB, and PB+IP, Random Forest got higher scores than XGBoost in IP, TP, TP+IP, TP+PB features combinations. The difference in the performance is a natural thing for the machine learning algorithm, with each algorithm has its own speciality depending on the task, case, shape of the data in the higher dimension plane, and the features included in the training process (Ahmad et al., 2011; Alzubi et al., 2018; Bahel, Pillai, et al., 2020; Canbek et al., 2017, 2021; Han et al., 2018; Joharestani et al., n.d.; Lin et al., 2007; Montebruno et al., 2020; Sarker, 2021; Shichkin et al., 2018; Sokolova & Lapalme, 2009; Zang et al., 2017). In addition, Random Forest might perform better than XGBoost due to the advantages of bagging algorithm over boosting algorithms in imbalanced datasets (Khoshgoftaar et al., 2011; B. Wang & Pineau, 2013). In addition, this might be attributed to the model tuning method. As asserted by (Freeman et al., 2016), a different result will be obtained with a different model, model tuning parameters, and variables.

Relatively similar results are shown in the experiment that was conducted using IP, TP, and IP+TP. This might be attributed to the multicollinearity of the variables. According to Farrar & Glauber (1967), multicollinearity is a condition where one predictor in regression is correlated and can be predicted by using another variable with sure accuracy. Although multicollinearity might affect the interpretability of statistical analysis (Belsley et al., 1980), it has no direct effect on the cross-validated prediction accuracy (Lieberman & Morris, 2014).

When comparing IP, TP and PB individually, PB comes short compared to IP and TP. While IP and TP got a moderate level of MCC score (0.63 and 0.64), PB only got 0.408. This might be attributed to some of the PB features are not significantly associated with the match outcome. The variable of *gender0* and *gender1* do not have any association with match outcome based on the chi-square test conducted in EDA part.

The better performance shown by the experiment that combines all features (TP + PB + IP) is in line with the result from Zang et al. (2017) which stated that the combination of all features obtained the best result compared to other combinations. However, there is a different aspect, while Zang et al. (2017), uses graph-based features alongside topological and profile based, this experiment use interaction pattern features. The first reason is associated with the statistical properties. The second reason is attributed to the nature of the romantic relationship itself.

5.2 Ethical Implications

The number of non-binary people in the experiment is very small. Only 1 user with genderNeutral identity compared to man (533) and woman (558). In addition, the type of match in the dataset are mostly made up from man-woman matches. This could result other type of matches (i.e., man-man, woman-woman, man-genderNeutral) and gender identities could be underrepresented.

5.3 Limitations

This experiment is conducted with a sparse dataset. The incompleteness, especially for the variable that formed profile-based features, is almost more than 80% of the total column.

The availability of the profile pictures and uploaded media also limits the methods that can be applied to the experiment. According to Curran & Lippold (1975) and Walster et al. (1966), physical attractiveness plays a vital role in a romantic relationship. However, the currently available data only have a limited number of columns that describe physical appearances and attractiveness. In addition to that, the sparsity also occurred in other profile-based features such as sociodemographic, interest and hobbies, and bio-related variables. Due to this, the experiment does not directly use each profile-based features, Instead, the completeness of each profile-based features is used as a substitute. Although this condition could partially predict the match outcome, the MCC scores show weak correlation with the target. This experiment assumes that the completeness of the features could partially capture the features as a whole.

This experiment also assumed that the only interaction prior to the match are views and likes only. In the real dataset, there are several other interactions that could be conducted prior to the match, such as mystery game and questions in the profile. However, due to the sparsity of the data and the vary limited number on the mentioned interaction, this experiment only account likes and views as the interaction before the match.

In addition, the time factor also prevents the experiment from using a more advanced approach like graph-neural networks. This is due to the more extensive process that needed to be conducted in order to model the data into graph-based data.

5.4 Further Exploration

We may consider using the graph neural networks approach. Since Liben-nowell & Kleinberg (2003) there have been several experiments that study the link prediction in a social network. Sanchez-Lengeling et al. (2021) also asserted that the interaction in dating app platforms also could be classified as a social network interaction, and the analysis around the interaction can be conducted by utilising graph neural networks. In addition to that, (M. Zhang & Chen, 2018) introduced the SEAL (Subgraphs, Embeddings and Attributes for Link prediction) framework for predicting the link between nodes in a social network which, unlike other graph neural networks that only account for graph-based data, SEAL introduce the features of each node. With this idea in mind, SEAL much resembles the condition of a graph-based analysis of social networks and dating apps.

In addition to the implementation of GNN and graph-based data, further exploration also can further assess the result of each algorithm's parameter configuration. For instance, a different *kernel* parameter option. Moreover, a more thorough exploratory data analysis needs to be conducted in further research to assess the availability and effect of multicollinearity between variables in match prediction. In addition, feature selection processes such as LDA and PCA could be implemented to reduce the number of variables; thus, the calculation process will be more effective.

Further exploration of a complete dataset could also be conducted. The complete data does mean not only the minimum number of missing values of data but also the diversity of the data. For instance, instead of measuring physical appearances by only using the values of height, eye colour, and hair colour, further exploration could implement a facial recognition system to determine the attractiveness of the users based on the uploaded photos/media. As asserted by Kagian et al. (2008), Stepanek et al. (2018), and Yu et al. (2014) which proposed an evaluation of facial attractiveness using machine learning. Indeed, further ethical and security aspects need to be considered prior to the experiments. In addition, the availability of more diverse bios of the users could also be used to explore the effect and the semantics of user bios to match outcomes.

A thorough evaluation of the comparison of XGBoost, MLP, and Random Forest Performance in predicting a match outcome needs to be discussed. Although the overall XGBoost performances are better based on the average MCC score across different feature combinations, Random Forest and MLP's performance tops XGBoost in several cases. Both potential arguments of why XGBoost might performed better in one case and lose in others have already been exercised in the previous section. However, there are still no definitive answers to the inconsistencies. Future research can focus on the comparison between the algorithm under the same set of features with fine-tuned algorithm parameters.

5.5 Conclusion

Experiments to find an automated method to predict match outcome from dating app has been proposed. The automated method uses machine learning algorithms with several combinations of features. The experiment was conducted in collaboration with Fluttr dating app, which provides the primary datasets, which were later combined to form a combined table. The combined table is a tabular-form data which is seen as more suitable for this experiment. The experiment shows that using all the features together (TP + PB + IP) produces the best result compared to other feature combinations. In addition, although there are some inconsistencies in the result, XGBoost performed better compared to other algorithms. Accordingly, the best result in predicting match outcome is produced by combining the XGBoost algorithm with topological, profile-based, and interaction pattern features.

6 References

Abiodun, O. I., Kiru, M. U., Jantan, A., Omolara, A. E., Dada, K. V., Umar, A. M., Linus, O. U., Arshad, H., Kazaure, A. A., & Gana, U. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access*, *7*, 158820–158846. https://doi.org/10.1109/ACCESS.2019.2945545

Ahmad, A., Mustapha, A., Zahadi, E. D., Masah, N., & Yahaya, N. Y. (2011). Comparison between Neural Networks against Decision Tree in Improving Prediction Accuracy for Diabetes Mellitus. In *CCIS* (Vol. 188, pp. 537–545). https://doi.org/10.1007/978-3-642-22389-1_47

Akehurst, J., Koprinska, I., Yacef, K., Pizzato, L., Kay, J., & Rej, T. (2012). Explicit and Implicit User Preferences in Online Dating. In *LNAI* (Vol. 7104).

Alshemali, B., & Kalita, J. (2020). Improving the Reliability of Deep Neural Networks in NLP: A Review. *Knowledge-Based Systems*, *191*, 105210. https://doi.org/10.1016/j.knosys.2019.105210

Alzubi, J., Nayyar, A., & Kumar, A. (2018). Machine Learning from Theory to Algorithms: An Overview. *Journal of Physics: Conference Series*, *1142*, 012012. https://doi.org/10.1088/1742-6596/1142/1/012012

Asendorpf, J. B., Penke, L., & Back, M. D. (2011). From Dating to Mating and Relating: Predictors of Initial and Long–Term Outcomes of Speed–Dating in a Community Sample. *European Journal of Personality*, *25*(1), 16–30. https://doi.org/10.1002/per.768

Attoh-Okine, N. O. (1999). Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. *Advances in Engineering Software*, *30*(4), 291–302. https://doi.org/10.1016/S0965-9978(98)00071-4

Bahel, V., In, S. P., & Malhotra, M. (2020). *A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance*.

Bahel, V., Pillai, S., & Malhotra, M. (2020). A Comparative Study on Various Binary Classification Algorithms and their Improved Variant for Optimal Performance. *2020 IEEE Region 10 Symposium (TENSYMP)*, 495–498. https://doi.org/10.1109/TENSYMP50017.2020.9230877

Bbeiman, L. (1996). *Bagging Predictors* (Vol. 24).

Belsley, D., Kuh, E., & Welsch, R. (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity (Wiley Series in Probability and Statistics)* (1st ed.). Wiley-Interscience.

Benson, H. (2021). *Relative Strangers: The Importance of Social Capital for Marriage*. www.marriagefoundation.org.uk

Berger, C. R., & Calabrese, R. J. (1975). *SOME EXPLORATIONS IN INITIAL INTERACTION AND BEYOND: TOWARD A DEVELOPMENTAL THEORY OF INTERPERSONAL COMMUNICATION*.

Bologna, G. (2021). A Rule Extraction Technique Applied to Ensembles of Neural Networks, Random Forests, and Gradient-Boosted Trees. *Algorithms*, *14*(12), 339. https://doi.org/10.3390/a14120339

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). *Deep Neural Networks and Tabular Data: A Survey*.

Botnen, E. O., Bendixen, M., Grøntvedt, T. V., & Kennair, L. E. O. (2018a). Individual differences in sociosexuality predict picture-based mobile dating app use. *Personality and Individual Differences*, *131*, 67–73. https://doi.org/10.1016/j.paid.2018.04.021

Botnen, E. O., Bendixen, M., Grøntvedt, T. V., & Kennair, L. E. O. (2018b). Individual differences in sociosexuality predict picture-based mobile dating app use. *Personality and Individual Differences*, *131*, 67–73. https://doi.org/10.1016/j.paid.2018.04.021

Brand, R. J., Bonatsos, A., D'Orazio, R., & DeShong, H. (2012). What is beautiful is good, even online: Correlations between photo attractiveness and text attractiveness in men's online dating profiles. *Computers in Human Behavior*, *28*(1), 166–170. https://doi.org/10.1016/j.chb.2011.08.023

Breiman, L. (2001). *Random Forests* (Vol. 45). https://doi.org/https://doi.org/10.1023/A:1010933404324

Cacioppo, J. T., Cacioppo, S., Gonzaga, G. C., Ogburn, E. L., & Vanderweele, T. J. (2013). Marital satisfaction and break-ups differ across on-line and off-line meeting venues. *Proceedings of the National Academy of Sciences of the United States of America*, *110*(25), 10135–10140. https://doi.org/10.1073/pnas.1222447110

Cai, X., Bain, M., Krzywicki, A., Wobcke, W., Kim, Y. S., Compton, P., & Mahidadia, A. (2012). *Reciprocal and Heterogeneous Link Prediction in Social Networks*.

Canbek, G., Sagiroglu, S., Temizel, T. T., & Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. *2017 International Conference on Computer Science and Engineering (UBMK)*, 821–826. https://doi.org/10.1109/UBMK.2017.8093539

Canbek, G., Taskaya Temizel, T., & Sagiroglu, S. (2021). BenchMetrics: a systematic benchmarking method for binary classification performance metrics. *Neural Computing and Applications*, *33*(21), 14623–14650. https://doi.org/10.1007/s00521-021-06103-6

Canley. (n.d.). *Logistic Regression*. https://en.wikipedia.org/wiki/Logistic_regression#/media/File:Exam_pass_logistic_curve.svg

Castro, Á., & Barrada, J. R. (2020). Dating apps and their sociodemographic and psychosocial correlates: A systematic review. *International Journal of Environmental Research and Public Health*, *17*(18), 1–25. https://doi.org/10.3390/IJERPH17186500

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen's Kappa and Brier Score in Binary Classification Assessment. *IEEE Access*, *9*, 78368–78381. https://doi.org/10.1109/ACCESS.2021.3084050

Choong, A. C. H., & Lee, N. K. (2017). Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method. *2017 International Conference on Computer and Drone Applications (IConDA)*, 60–65. https://doi.org/10.1109/ICONDA.2017.8270400

Colas, F., & Brazdil, P. (2006). Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In M. Bramer (Ed.), *Artificial Intelligence in Theory and Practice* (pp. 169–178). Springer US. https://doi.org/10.1007/978-0-387-34747-9_18

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Couch, D., Liamputtong, P., & Pitts, M. (2012). *What are the real and perceived risks and dangers of online dating? Perspectives from online daters Health risks in the media*. https://doi.org/10.1080/13698575.2012.720964

Curran, J. P., & Lippold, S. (1975). The effects of physical attraction and attitude similarity on attraction in dating dyads. *Journal of Personality*, *43*(3), 528–539. https://doi.org/10.1111/j.1467-6494.1975.tb00720.x

Danisik, N., Lacko, P., & Farkas, M. (2018). Football Match Prediction Using Players Attributes; Football Match Prediction Using Players Attributes. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. http://www.dictionary.com/browse/football-,

David, G., & Cambre, C. (2016). Screened Intimacies: Tinder and the Swipe Logic. *Social Media + Society*, *2*(2), 205630511664197. https://doi.org/10.1177/2056305116641976

Douglas, W. (1990). Uncertainty, information-seeking, and liking during initial interaction. *Western Journal of Speech Communication*, *54*(1), 66–81. https://doi.org/10.1080/10570319009374325

Ekman, M. (2021). *Learning Deep Learning: Theory and Practice of Neural Networks, Computer Vision, Natural Language Processing, and Transformers Using TensorFlow* (1st ed.). Addison-Wesley Professional.

Fadlian, M. F., Azhari, M. B., & Kusumoputro, B. (2021). Data acquisition of X-plane's aircraft through matlab for neural network based identification system. *AIP Conference Proceedings*, *2376*, 060002. https://doi.org/10.1063/5.0066213

Fanelli, S., di Martino, M., & Protasi, M. (1993). An efficient algorithm for the binary classification of patterns using MLP-networks. *IEEE International Conference on Neural Networks*, 936–943. https://doi.org/10.1109/ICNN.1993.298683

Faris, H., Aljarah, I., & Mirjalili, S. (2016). Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence*, *45*(2), 322–332. https://doi.org/10.1007/s10489-016-0767-1

Farrar, D. E., & Glauber, R. R. (1967). *Multicollinearity in Regression Analysis: The Problem Revisited* (Vol. 49, Issue 1). https://about.jstor.org/terms

Feng, J., Xu, H., Mannor, S., & Yan, S. (2014). Robust Logistic Regression and Classification. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 27). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2014/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper.pdf

Freeman, E. A., Moisen, G. G., Coulston, J. W., & Wilson, B. T. (2016). Random forests and stochastic gradient boosting for predicting tree canopy cover: comparing tuning processes and model performance. *Canadian Journal of Forest Research*, *46*(3), 323–339. https://doi.org/10.1139/cjfr-2014-0562

Friedman, J. H. (2002). Stochastic gradient boosting. In *Computational Statistics & Data Analysis* (Vol. 38). www.elsevier.com/locate/csda

Furman, W., & Shomaker, L. B. (2008). Patterns of interaction in adolescent romantic relationships: Distinct features and links to other close relationships. *Journal of Adolescence*, *31*(6), 771–788. https://doi.org/10.1016/j.adolescence.2007.10.007

Gale, D., & Shapley, L. S. (1962). College Admissions and the Stability of Marriage. In *Source: The American Mathematical Monthly* (Vol. 69, Issue 1).

Galliher, R. v., Welsh, D. P., Rostosky, S. S., & Kawaguchi, M. C. (2004). Interaction and Relationship Quality in Late Adolescent Romantic Couples. *Journal of Social and Personal Relationships*, *21*(2), 203–216. https://doi.org/10.1177/0265407504041383

Grgić, V., Mušić, D., & Babović, E. (2021). Model for predicting heart failure using Random Forest and Logistic Regression algorithms. *IOP Conference Series: Materials Science and Engineering*, *1208*(1), 012039. https://doi.org/10.1088/1757-899X/1208/1/012039

Guan, S., & Wang, X. (2022). Optimization analysis of football match prediction model based on neural network. *Neural Computing and Applications*, *34*(4), 2525–2541. https://doi.org/10.1007/s00521-021-05930-x

Gupta, A. (2022). *XGBoost versus Random Forest - Geek Culture*. https://medium.com/geekculture/xgboost-versus-random-forest-898e42870f30

Han, T., Jiang, D., Zhao, Q., Wang, L., & Yin, K. (2018). Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery. *Transactions of the Institute of Measurement and Control*, *40*(8), 2681–2693. https://doi.org/10.1177/0142331217708242

Herbinet, C. (2018). *Predicting Football Results Using Machine Learning Techniques*. Department of Computing, Imperial College London.

Hinton, G. E. (1989). Connectionist learning procedures. *Artificial Intelligence*, *40*(1–3), 185–234. https://doi.org/10.1016/0004-3702(89)90049-0

Hitsch, G. J., Hortaçsu, A., & Ariely, D. (2006). *MIT Sloan School of Management What Makes You Click?-Mate Preferences and Matching Outcomes in Online Dating What Makes You Click?-Mate Preferences and Matching Outcomes in Online Dating **. http://ssrn.com/abstract=895442

Hitsch, G. J., Hortaçsu, A., & Ariely, D. (2010). Matching and Sorting in Online Dating. *American Economic Review*, *100*(1), 130–163. https://doi.org/10.1257/aer.100.1.130

Hitsch, G. J., Hortaçsu, A., Ariely, D., Hitsch, G. J., Hortaçsu, A., & Ariely, D. (2010). *What makes you click?-Mate preferences in online dating*. *8*, 393–427. https://doi.org/10.1007/s11129-010-9088-6

Houser, M. L., Horan, S. M., & Furler, L. A. (2008). Dating in the fast lane: How communication predicts speed-dating success. *Www.Sagepublications.Com)*, *25*(5), 749–768. https://doi.org/10.1177/0265407508093787

Iyengar, S. S., & Lepper, M. R. (2000). When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, *79*(6), 995–1006. https://doi.org/10.1037/0022-3514.79.6.995

Jebb, A. T., Parrigon, S., & Woo, S. E. (2017). Exploratory data analysis as a foundation of inductive research. *Human Resource Management Review*, *27*(2), 265–276. https://doi.org/10.1016/j.hrmr.2016.08.003

Joharestani, M. Z., Cao, C., Ni, X., Bashir, B., & Talebiesfandarani, S. (n.d.). *PM 2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data*. https://doi.org/10.3390/atmos10070373

Johnson, M. A. (1989). Variables Associated with Friendship in an Adult Population. *The Journal of Social Psychology*, *129*(3), 379–390. https://doi.org/10.1080/00224545.1989.9712054

Kabiraj, S., Raihan, M., Alvi, N., Afrin, M., Akter, L., Sohagi, S. A., & Podder, E. (2020). Breast Cancer Risk Prediction using XGBoost and Random Forest Algorithm. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–4. https://doi.org/10.1109/ICCCNT49239.2020.9225451

Kagian, A., Dror, G., Leyvand, T., Meilijson, I., Cohen-Or, D., & Ruppin, E. (2008). A machine learning predictor of facial attractiveness revealing human-like psychophysical biases. *Vision Research*, *48*(2), 235–243. https://doi.org/10.1016/j.visres.2007.11.007

Kavzoglu, T., & Mather, P. M. (2003). The use of backpropagating artificial neural networks in land cover classification. *International Journal of Remote Sensing*, *24*(23), 4907–4938. https://doi.org/10.1080/0143116031000114851

Khoshgoftaar, T. M., van Hulse, J., & Napolitano, A. (2011). Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *41*(3), 552–568. https://doi.org/10.1109/TSMCA.2010.2084081

Kirasich, K. ;, Smith, T. ;, & Sadler, B. (2018). Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets. In *SMU Data Science Review* (Vol. 1, Issue 3). https://scholar.smu.edu/datasciencereviewAvailableat:https://scholar.smu.edu/datasciencereview/vol1/iss3/9http://digitalrepository.smu.edu.

Kwak, S. K., & Kim, J. H. (2017). Statistical data preparation: management of missing values and outliers. *Korean Journal of Anesthesiology*, *70*(4), 407. https://doi.org/10.4097/kjae.2017.70.4.407

LaMorte, W. W. (2021). *The Correlation Coefficient (r)*. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module9-Correlation-Regression/PH717-Module9-Correlation-Regression4.html#:%7E:text=Possible%20values%20of%20the%20correlation,little%2C%20if%20any%2C%20correlation.

Lavesson, N., & Davidsson, P. (2006). Quantifying the Impact of Learning Algorithm Parameter Tuning. *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, 395–400.

Lee, L., Loewenstein, G., Ariely, D., Hong, J., & Young, J. (2008). If I'm Not Hot, Are You Hot or Not? *Psychological Science*, *19*(7), 669–677. https://doi.org/10.1111/j.1467-9280.2008.02141.x

LeFebvre, L. E. (2018). Swiping me off my feet: Explicating relationship initiation on Tinder. *Journal of Social and Personal Relationships*, *35*(9), 1205–1229. https://doi.org/10.1177/0265407517706419

Lewandowski, G. W. (2018). *The influence of past relationships on subsequent relationships: The role of the self*. https://www.researchgate.net/publication/290264995

Liben-nowell, D., & Kleinberg, J. (2003). The Link Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology*, *58*. https://doi.org/10.1002/asi.20591

Lieberman, M. G., & Morris, J. D. (2014). *Multicollinearity and Classification Prediction Multiple Linear Regression Viewpoints* (Vol. 40, Issue 1).

Lin, K.-L., Lin, C.-Y., Huang, C.-D., Chang, H.-M., Yang, C.-Y., Lin, C.-T., Tang, C. Y., Frank Hsu, D., Lin, C.-Y., Chang, H.-M., & Hsu, D. F. (2007). Feature Selection and Combination Criteria for Improving Accuracy in Protein Structure Prediction. *IEEE TRANSACTIONS ON NANOBIOSCIENCE*, *6*(2). https://doi.org/10.1109/TNB.2007.897482

Luo, S., & Tuney, S. (2015). Can texting be used to improve romantic relationships?—The effects of sending positive text messages on relationship satisfaction. *Computers in Human Behavior*, *49*, 670–678. https://doi.org/10.1016/j.chb.2014.11.035

Mallick, S. (2021). *Support Vector Machines (SVM) | LearnOpenCV #*. https://learnopencv.com/support-vector-machines-svm/

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., & Carvalho, A. C. P. L. F. (2015). *To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning; To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning*. https://doi.org/10.1109/IJCNN.2015.7280644

Marcot, B. G., & Hanea, A. M. (2021). What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? *Computational Statistics*, *36*(3), 2009–2031. https://doi.org/10.1007/s00180-020-00999-9

Mare, R. D. (1991). Five Decades of Educational Assortative Mating. *American Sociological Review*, *56*(1), 15–32. https://doi.org/10.2307/2095670

Mathur, A., & Foody, G. M. (2008). Multiclass and Binary SVM Classification: Implications for Training and Classification Users. *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, *5*(2). https://doi.org/10.1109/LGRS.2008.915597

Mccormick, T. H., Raftery, A. E., Madigan, D., & Burd, R. S. (2012). Dynamic Logistic Regression and Dynamic Model Averaging for Binary Classification. *Biometrics*, *68*. https://doi.org/10.1111/j.1541-0420.2011.01645.x

Menzli, A. (2021). *Graph Neural Network and Some of GNN Applications: Everything You Need to Know*. https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications

Montebruno, P., Bennett, R. J., Smith, H., & Lieshout, C. van. (2020). Machine learning classification of entrepreneurs in British historical census data. *Information Processing & Management*, *57*(3), 102210. https://doi.org/10.1016/j.ipm.2020.102210

Mortaz, E. (2020). Imbalance accuracy metric for model selection in multi-class imbalance classification problems. *Knowledge-Based Systems*, *210*. https://doi.org/10.1016/j.knosys.2020.106490

Multi-layer Perceptron in TensorFlow - Javatpoint. (n.d.). https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow

Naber, M. (2022). *The Advantages of Synthetic Data Over Real Data*. https://neptune.ai/blog/the-advantages-of-synthetic-data-over-real-data

Nayak, R., Zhang, M., & Chen, L. (2010). A social matching system for an online dating network: A preliminary study. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 352–357. https://doi.org/10.1109/ICDMW.2010.36

Neustaedter, C., & Greenberg, S. (2012). Intimacy in long-distance relationships over video chat. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 753–762. https://doi.org/10.1145/2207676.2207785

Nghiep, N., & Al, C. (2001). Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. *Journal of Real Estate Research*, *22*(3), 313–336. https://doi.org/10.1080/10835547.2001.12091068

Omolo, J. (2020). *Crossing the Colour Line*. Amazon Digital Services LLC - KDP Print US.

Orosz, G., Benyo, M., Berkes, B., Nikoletti, E., Gál, É., Tóth-Király, I., & Bőthe, B. (2018). The personality, motivational, and need-based background of problematic Tinder use. *Journal of Behavioral Addictions*, *7*(2), 301–316. https://doi.org/10.1556/2006.7.2018.21

Paul, D., Su, R., Romain, M., Sébastien, V., Pierre, V., & Isabelle, G. (2017). Feature selection for outcome prediction in oesophageal cancer using genetic algorithm and random forest classifier. *Computerized Medical Imaging and Graphics : The Official Journal of the Computerized Medical Imaging Society*, *60*, 42–49. https://doi.org/10.1016/j.compmedimag.2016.12.002

Paulhus, D. L., Curtis, S. R., & Jones, D. N. (2018). Aggression as a trait: the Dark Tetrad alternative. *Current Opinion in Psychology*, *19*, 88–92. https://doi.org/10.1016/j.copsyc.2017.04.007

Phan, A., Seigfried-Spellar, K., & Choo, K. K. R. (2021). Threaten me softly: A review of potential dating app risks. *Computers in Human Behavior Reports*, *3*, 100055. https://doi.org/10.1016/J.CHBR.2021.100055

Pizzato, L., Rej, T., Chung, T., Koprinska, I., & Kay, J. (2006). *RECON: A Reciprocal Recommender for Online Dating*. ACM.

Probst, P., Boulesteix, A.-L., & Bischl, B. (2018). *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*.

Pronk, T. M., & Denissen, J. J. A. (2020). A Rejection Mind-Set: Choice Overload in Online Dating. *Social Psychological and Personality Science*, *11*(3), 388–396. https://doi.org/10.1177/1948550619866189

Quinlan, J. R. (1996). Boosting first-order learning. In A. K. Arikawa Setsuo and Sharma (Ed.), *Algorithmic Learning Theory* (pp. 143–155). Springer Berlin Heidelberg.

Ranzini, G., & Lutz, C. (2017). Love at first swipe? Explaining Tinder self-presentation and motives. *Mobile Media and Communication*, *5*(1), 80–101. https://doi.org/10.1177/2050157916664559

Rosenfeld, M. J., Thomas, R. J., & Hausen, S. (2019). Disintermediating your friends: How online dating in the United States displaces other ways of meeting. *Proceedings of the National Academy of Sciences*, *116*(36), 17753–17758. https://doi.org/10.1073/pnas.1908630116

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. (2021). A Gentle Introduction to Graph Neural Networks. *Distill*, *6*(8). https://doi.org/10.23915/distill.00033

Sarker, I. H. (2021). *Machine Learning: Algorithms, Real-World Applications and Research Directions*. *2*, 160. https://doi.org/10.1007/s42979-021-00592-x

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Schwartz, C. R., & Mare, R. D. (2012). The Proximate Determinants of Educational Homogamy: The Effects of First Marriage, Marital Dissolution, Remarriage, and Educational Upgrading. *Demography*, *49*(2), 629–650. https://doi.org/10.1007/s13524-012-0093-0

Shahrokh Esfahani, M., & Dougherty, E. R. (2014). Effect of separate sampling on classification accuracy. *Bioinformatics*, *30*(2), 242–250. https://doi.org/10.1093/bioinformatics/btt662

Shichkin, A. v., Buevich, A. G., & Sergeev, A. P. (2018). Comparison of artificial neural network, random forest and random perceptron forest for forecasting the spatial impurity distribution. *AIP Conference Proceedings*, *1982*, 020005. https://doi.org/10.1063/1.5045411

Shumway, R., & Stoffer, D. (2010). *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)* (3rd ed. 2011). Springer.

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Smith, A. D. (2005). Exploring online dating and customer relationship management. *Online Information Review*, *29*(1), 18–33. https://doi.org/10.1108/14684520510583927

Snyder, M., Tanke, E. D., & Berscheid, E. (1977). Social perception and interpersonal behavior: On the self-fulfilling nature of social stereotypes. *Journal of Personality and Social Psychology*, *35*(9), 656–666. https://doi.org/10.1037/0022-3514.35.9.656

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Stepanek, L., Kasal, P., & Mestak, J. (2018). Evaluation of facial attractiveness for purposes of plastic surgery using machine-learning methods and image analysis. *2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom)*, 1–6. https://doi.org/10.1109/HealthCom.2018.8531195

Steyerberg, E. W., & Harrell, F. E. (2016). Prediction models need appropriate internal, internal–external, and external validation. *Journal of Clinical Epidemiology*, *69*, 245–247. https://doi.org/10.1016/j.jclinepi.2015.04.005

Stonard, K. E., Bowen, E., Lawrence, T. R., & Price, S. A. (2014). The relevance of technology to the nature, prevalence and impact of Adolescent Dating Violence and Abuse: A research synthesis. *Aggression and Violent Behavior*, *19*(4), 390–417. https://doi.org/10.1016/j.avb.2014.06.005

Sumter, S. R., & Vandenbosch, L. (2019). Dating gone mobile: Demographic and personality-based correlates of using smartphone-based dating applications among emerging adults. *New Media and Society*, *21*(3), 655–673. https://doi.org/10.1177/1461444818804773

Sumter, S. R., Vandenbosch, L., & Ligtenberg, L. (2017a). Love me Tinder: Untangling emerging adults' motivations for using the dating application Tinder. *Telematics and Informatics*, *34*(1), 67–78. https://doi.org/10.1016/j.tele.2016.04.009

Sumter, S. R., Vandenbosch, L., & Ligtenberg, L. (2017b). Love me Tinder: Untangling emerging adults' motivations for using the dating application Tinder. *Telematics and Informatics*, *34*(1), 67–78. https://doi.org/10.1016/j.tele.2016.04.009

Suthaharan, S. (2016). Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* (pp. 207–235). Springer US. https://doi.org/10.1007/978-1-4899-7641-3_9

Swani, K., Milne, G. R., Brown, B. P., Assaf, A. G., & Donthu, N. (2017). What messages to post? Evaluating the popularity of social media communications in business versus consumer markets. *Industrial Marketing Management*, *62*, 77–87. https://doi.org/10.1016/j.indmarman.2016.07.006

Tangirala, S. (2020). Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*. In *IJACSA) International Journal of Advanced Computer Science and Applications* (Vol. 11, Issue 2). www.ijacsa.thesai.org

Timmermans, E., & de Caluwé, E. (2017). Development and validation of the Tinder Motives Scale (TMS). *Computers in Human Behavior*, *70*, 341–350. https://doi.org/10.1016/j.chb.2017.01.028

Toffler, A. (1970). *Future Shock*. Random House, Inc.

Vapnik, V., & Chervonenkis, A. (1974). *Theory of Pattern Recognition*. Nauka.

Walster, E. (1970). The effect of self-esteem on liking for dates of various social desirabilities. *Journal of Experimental Social Psychology*, *6*(2), 248–253. https://doi.org/10.1016/0022-1031(70)90090-9

Walster, E., Aronson, V., Abrahams, D., & Rottman, L. (1966). Importance of physical attractiveness in dating behavior. *Journal of Personality and Social Psychology*, *4*(5), 508–516. https://doi.org/10.1037/h0021188

Wang, B., & Pineau, J. (2013). *Online Ensemble Learning for Imbalanced Data Streams*. http://arxiv.org/abs/1310.8004

Wang, S. (2018). *Calculating dating goals: data gaming and algorithmic sociality on Blued, a Chinese gay dating app*. https://doi.org/10.1080/1369118X.2018.1490796

Wang, S.-C. (2003). Artificial Neural Network. In *Interdisciplinary Computing in Java Programming* (pp. 81–100). Springer US. https://doi.org/10.1007/978-1-4615-0377-4_5

Wang, Z., Tsai, C.-F., & Lin, W.-C. (2021). Data cleaning issues in class imbalanced datasets: instance selection and missing values imputation for one-class classifiers. *Data Technologies and Applications*, *55*(5), 771–787. https://doi.org/10.1108/DTA-01-2021-0027

Weerts, H. J. P., Mueller, A. C., & Vanschoren, J. (2020). *Importance of Tuning Hyperparameters of Machine Learning Algorithms*. http://arxiv.org/abs/2007.07588

Wu, Y., Zhang, K., & Padmanabhan, V. (2018). Matchmaker Competition and Technology Provision. *Journal of Marketing Research*, *ISSN*, 396–413. https://doi.org/10.1509/jmr.16.0423

Xia, P., Jiang, H., Wang, X., Chen, C., & Liu, B. (2014a). *Predicting User Replying Behavior on a Large Online Dating Site*. http://dblp.uni-trier.de

Xia, P., Jiang, H., Wang, X., Chen, C., & Liu, B. (2014b). Predicting User Replying Behavior on a Large Online Dating Site. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 545–554. https://ojs.aaai.org/index.php/ICWSM/article/view/14516

Yu, X., Liu, B., Pei, Y., & Xu, T. (2014). Evaluation of facial attractiveness for patients with malocclusion: A machine-learning technique employing Procrustes. *The Angle Orthodontist*, *84*(3), 410–416. https://doi.org/10.2319/071513-516.1

Zang, X., Yamasaki, T., Aizawa, K., Nakamoto, T., Kuwabara, E., Egami, S., & Fuchida, Y. (2017). You Will Succeed or Not Matching Prediction in a Marriage Consulting Service. *Proceedings - 2017 IEEE 3rd International Conference on Multimedia Big Data, BigMM 2017*, 109–116. https://doi.org/10.1109/BigMM.2017.11

Zhang, J., & Yasseri, T. (2016). *What Happens After You Both Swipe Right: A Statistical Description of Mobile Dating Communications*.

Zhang, M., & Chen, Y. (2018). *Link Prediction Based on Graph Neural Networks*.

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C-Emerging Technologies*, *58*, 308–324.

Zheng, N. (2019). Fundamentals and Learning of Artificial Neural Networks. In *Learning in Energy-Efficient Neuromorphic Computing* (pp. 11–60). Wiley. https://doi.org/10.1002/9781119507369.ch2

7 Appendix

7.1 Code instruction

The data collection from MongoDB can be accplished through Python and KNIME. The KNIME version is uploaded as “Import MongoDB1”

However, most of the data construction and data cleaning is done separately and interchangeably on both Microsoft Excel and Python, hence the code cannot be uploaded.

The main code for EDA, column filtering, and algorithm+features experiment is attached and can be directly run through terminal using python.

7.2 Code

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from xgboost import XGBClassifier

from sklearn import model_selection
from sklearn.utils import class_weight
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import matthews_corrcoef
from sklearn.preprocessing import MinMaxScaler

from scipy.stats import pointbiserialr
from scipy import stats

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

#read the data
df = pd.read_csv('result_pairwise_duplicate_completeness.csv')

#drop unnecessary columns
df.drop(['userID0', 'Activity','Activity.1', 'userID1', 'pairwise1',
'pairwise2', 'matchID', 'matchDate','matchDate.1'], axis = 1, inplace = True)

#filter the acrive users and pairs only
df.drop(df.loc[df['initialIteraction']<=1].index, inplace=True)

#drop unnecessary columns
```

```
df = df.drop('initialIteraction', axis = 1)

#get categorical columns
categorical_cols = ['gender0', 'lookingForGender0', 'lookingForType0',
'gender1', 'lookingForGender1', 'lookingForType1', 'matchType']

#one-hot-encoding for categorical columns
df_ohe = pd.get_dummies(df, columns = categorical_cols, drop_first = True)
X = df_ohe.drop('matchStatus', axis = 1)
y = df_ohe[['matchStatus']]

#get numerical data
numerical_features = [feature for feature in X.columns if X[feature].dtypes !=
'O']

#normalisation
trans = MinMaxScaler()
data = trans.fit_transform(X)
X_norm = pd.DataFrame(data, columns = numerical_features)

#rename column
X_norm.rename(columns={"'viewByUser0":"viewByUser0","'viewByUser0_beforeMatch"
:"viewByUser0_beforeMatch","'viewByUser1_beforeMatch":"viewByUser1_beforeMatch
"}, inplace = True)

#data description
desc = df.describe(include = 'all')
desc.to_csv('data_description.csv')

#calculate point-biserial correlation
df['target'] = df['matchStatus'].astype('category')

var_pbc_result = []
r_pbc_result = []
p_pbc_result = []
ass_pbc_result = []

for i in numerical_df:
    predictor = df[i]
    target = df['target']
    r, p = stats.pointbiserialr(predictor, target)
    print(i,"\n")
    print('r value=%.3f\np value=%.3f' % (r, p))
    alpha = 0.05
    if p <= alpha:
        ass = 'significant association (reject H0)'
        print(ass)
    else:
```

```
        ass = 'no significant association (accept H0)'
        print(ass)
    print('*'*70)
    var_pbc_result.append(i)
    r_pbc_result.append(r)
    p_pbc_result.append(p)
    ass_pbc_result.append(ass)

num_eda_pbc = pd.DataFrame(
    {'feature': var_pbc_result,
     'r_score': r_pbc_result,
     'p_value': p_pbc_result,
     'conclusion': ass_pbc_result
    })

#save the result
num_eda_pbc.to_csv('stat_df.csv')

# train-test split
X_train, X_test, y_train, y_test = train_test_split(X_norm, y, test_size =
0.2, stratify = y, random_state = 1)

#all features
def run_exps(X_train: pd.DataFrame , y_train: pd.DataFrame, X_test:
pd.DataFrame, y_test: pd.DataFrame) -> pd.DataFrame:
    '''
    Lightweight script to test many models and find winners
    :param X_train: training split
    :param y_train: training target vector
    :param X_test: test split
    :param y_test: test target vector
    :return: DataFrame of predictions
    '''

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']
```

```
for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)



that_df = pd.DataFrame(names, columns = ['Algorithm'])

#IP Features
X_IP = X_norm[["viewByUser0_beforeMatch",
               'firstLikeUser0',
               "viewByUser1_beforeMatch",
               'firstLikeUser1',
               'matchType_man-woman',
               'matchType_woman-genderNeutral',
               'matchType_woman-woman']]

X_train, X_test, y_train, y_test = train_test_split(X_IP, y, test_size = 0.2,
stratify = y, random_state = 1)
dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
```

```
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)
that_df['interactionPattern'] = mcc_scores

#PB Features
X_PB = X_norm[['gender0_man', 'gender0_woman',
               'lookingForGender0_man',
               'lookingForGender0_woman',
               'lookingForType0_friendship',
               'lookingForType0_letsSeeHowItGoes',
               'lookingForType0_longTermRelationships',
               'lookingForType0_marriage',
               'lookingForType0_nothingSerious',
               'gender1_man',
               'gender1_woman',
               'lookingForGender1_man',
               'lookingForGender1_woman',
               'lookingForType1_friendship',
               'lookingForType1_letsSeeHowItGoes',
               'lookingForType1_longTermRelationships',
               'lookingForType1_marriage',
               'lookingForType1_nothingSerious',
               'bioCompleteness0',
               'demographyCompleteness0',
               'physicalCompleteness0',
               'interestCompleteness0',
               'bioCompleteness1',
               'demographyCompleteness1',
               'physicalCompleteness1',
               'interestCompleteness1',]]

X_train, X_test, y_train, y_test = train_test_split(X_PB, y, test_size = 0.2,
stratify = y, random_state = 1)

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
```

```
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)

that_df['profileBased'] = mcc_scores

#TP Features
X_TF = X_norm[['reactionReceived0',
               'reactionGiven0',
               'mediaCount0',
               'viewReceived0',
               'viewGiven0',
               'matchAttemptGiven0',
               'matchAttemptReceived0',
               'matchAccepted0',
               'matchRejected0',
               'matchRejects0',
               'reactionReceived1',
               'reactionGiven1',
               'mediaCount1',
               'viewReceived1',
               'viewGiven1',
               'matchAttemptGiven1',
               'matchAttemptReceived1',
               'matchAccepted1',
               'matchRejected1',
               'matchRejects1'
              ]]
```

```
X_train, X_test, y_train, y_test = train_test_split(X_TF, y, test_size = 0.2,
stratify = y, random_state = 1)

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)

that_df['topological'] = mcc_scores

#TPIP Features
X_IPTF = X_norm[['reactionReceived0',
                 'reactionGiven0',
                 'mediaCount0',
                 'viewReceived0',
                 'viewGiven0',
                 'matchAttemptGiven0',
                 'matchAttemptReceived0',
                 'matchAccepted0',
                 'matchRejected0',
                 'matchRejects0',
                 'reactionReceived1',
                 'reactionGiven1',
                 'mediaCount1',
```

```
                'viewReceived1',
                'viewGiven1',
                'matchAttemptGiven1',
                'matchAttemptReceived1',
                'matchAccepted1',
                'matchRejected1',
                'matchRejects1',
                  'viewByUser0_beforeMatch',
                  'firstLikeUser0',
                  "viewByUser1_beforeMatch",
                  'firstLikeUser1',
                  'matchType_man-woman',
                  'matchType_woman-genderNeutral',
                  'matchType_woman-woman'
               ]]

X_train, X_test, y_train, y_test = train_test_split(X_TF, y, test_size = 0.2,
stratify = y, random_state = 1)

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)
```

```
that_df['Topological_interactioPattern'] = mcc_scores

#TPPB Features
X_TFPB = X_norm[['reactionReceived0',
                 'reactionGiven0',
                 'mediaCount0',
                 'viewReceived0',
                 'viewGiven0',
                 'matchAttemptGiven0',
                 'matchAttemptReceived0',
                 'matchAccepted0',
                 'matchRejected0',
                 'matchRejects0',
                 'reactionReceived1',
                 'reactionGiven1',
                 'mediaCount1',
                 'viewReceived1',
                 'viewGiven1',
                 'matchAttemptGiven1',
                 'matchAttemptReceived1',
                 'matchAccepted1',
                 'matchRejected1',
                 'matchRejects1',
                 'gender0_man', 'gender0_woman',
                 'lookingForGender0_man',
                 'lookingForGender0_woman',
                 'lookingForType0_friendship',
                 'lookingForType0_letsSeeHowItGoes',
                 'lookingForType0_longTermRelationships',
                 'lookingForType0_marriage',
                 'lookingForType0_nothingSerious',
                 'gender1_man',
                 'gender1_woman',
                 'lookingForGender1_man',
                 'lookingForGender1_woman',
                 'lookingForType1_friendship',
                 'lookingForType1_letsSeeHowItGoes',
                 'lookingForType1_longTermRelationships',
                 'lookingForType1_marriage',
                 'lookingForType1_nothingSerious',
                 'bioCompleteness0',
                 'demographyCompleteness0',
                 'physicalCompleteness0',
                 'interestCompleteness0',
                 'bioCompleteness1',
                 'demographyCompleteness1',
                 'physicalCompleteness1',
                 'interestCompleteness1'
```

```
              ]]

X_train, X_test, y_train, y_test = train_test_split(X_TFPB, y, test_size =
0.2, stratify = y, random_state = 1)

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)

that_df['Topological_profileBased'] = mcc_scores

#IPPB Features
X_IPPB = X_norm[[
               'gender0_man', 'gender0_woman',
               'lookingForGender0_man',
               'lookingForGender0_woman',
               'lookingForType0_friendship',
               'lookingForType0_letsSeeHowItGoes',
               'lookingForType0_longTermRelationships',
               'lookingForType0_marriage',
               'lookingForType0_nothingSerious',
               'gender1_man',
               'gender1_woman',
```

```
                'lookingForGender1_man',
                'lookingForGender1_woman',
                'lookingForType1_friendship',
                'lookingForType1_letsSeeHowItGoes',
                'lookingForType1_longTermRelationships',
                'lookingForType1_marriage',
                'lookingForType1_nothingSerious',
                'bioCompleteness0',
                'demographyCompleteness0',
                'physicalCompleteness0',
                'interestCompleteness0',
                'bioCompleteness1',
                'demographyCompleteness1',
                'physicalCompleteness1',
                'interestCompleteness1',
                 'viewByUser0_beforeMatch',
                 'firstLikeUser0',
                 "viewByUser1_beforeMatch",
                 'firstLikeUser1',
                 'matchType_man-woman',
                 'matchType_woman-genderNeutral',
                 'matchType_woman-woman'
               ]]

X_train, X_test, y_train, y_test = train_test_split(X_IPPB, y, test_size =
0.2, stratify = y, random_state = 1)

dfs = []
models = [('LogReg', LogisticRegression()),
          ('RF', RandomForestClassifier()),
          ('KNN', KNeighborsClassifier()),
          ('SVM', SVC()),
          ('GNB', GaussianNB()),
          ('XGB', XGBClassifier(verbosity = 0))]
results = []
names = []
mcc_scores = []
scoring = ['accuracy', 'precision_weighted', 'recall_weighted', 'f1_weighted',
'roc_auc']
target_names = ['ACCEPTED', 'NO MATCH']

for name, model in models:
    kfold = model_selection.KFold(n_splits=5, shuffle=True,
random_state=90210)
    cv_results = model_selection.cross_validate(model, X_train, y_train,
cv=kfold, scoring=scoring)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
```

```
    print(name)
#     print(classification_report(y_test, y_pred, target_names=target_names))
    mcc = matthews_corrcoef(y_test, y_pred)
    print('MCC Score: ', mcc)
    results.append(cv_results)
    names.append(name)
    mcc_scores.append(mcc)

that_df['profileBased_interactionPattern'] = mcc_scores

that_df.to_csv('final_result.csv')
```