**Applicant Success prediction for More Life**

Name: Vishwanath Reddy Aenugu

Student Id: 77249598

Submitted in partial fulfillment of the requirements for

M.Sc.

In Data Science

SCHOOL OF BUILT ENVIRONMENT, ENGINEERING & COMPUTING,

Leeds Beckett University.

Leeds Beckett University,

United Kingdom

September 2021

# Abstract

The project explores understanding and predicting attrition in weight management programs conducted at MoreLife, a community welfare organization. Supervised machine learning algorithms are utilized to predict attrition, and survival analysis was conducted to understand the participant's program survival chances during each week. The project development was carried in line with some components of the CRISP-DM project methodology by understanding MoreLife's requirements and data understanding, data cleaning, and model implementation. The study found that logistic regression, random forests, naïve Bayes, and multi-layer perceptron algorithms could predict attrition with reasonable accuracy, with random forests performing better than the other three. It is also found that dropout risks are relatively high in the initial and final program weeks. In addition, unemployed participants are relatively at higher risk of dropout. Programs held in January, February, November, and December are associated with higher dropouts, whereas programs held in June, August, September are associated with lower dropout risk.

# Candidate declaration

I confirm that this dissertation and the work presented in it are my own achievement. Where I have consulted the published work of others, this is always clearly attributed. Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work; I have acknowledged all primary sources of help; I have read and understand the penalties associated with Academic Misconduct.

Table of Contents

[Abstract 2](#_Toc82181050)

[Candidate declaration 2](#_Toc82181051)

[ABBREVIATIONS/ACRONYMS 5](#_Toc82181052)

[Obesity and Its health consequences 5](#_Toc82181053)

[The scale of Obesity and its financial burden to the UK 5](#_Toc82181054)

[Life-style Management services and the role of MoreLife to address the concern. 6](#_Toc82181055)

[An overview of MoreLife 7](#_Toc82181056)

[Early participant dropout a challenge to weight interventions and MoreLife 7](#_Toc82181057)

[Project Aim 10](#_Toc82181058)

[Projective Objectives 10](#_Toc82181059)

[Into the role of machine learning in weight management a Literature Review 10](#_Toc82181060)

[Selecting an appropriate project management methodology. 13](#_Toc82181061)

[A comparison of various project management methodologies: 13](#_Toc82181062)

[The project approach through CRISP-DM 14](#_Toc82181063)

[A brief introduction to predictive modeling through supervised machine learning approach, statistical analysis, and model validation 16](#_Toc82181064)

[Ensuring model reliability and resampling through k-fold Cross-validation: 16](#_Toc82181065)

[Methods to test and evaluate model performance: 17](#_Toc82181066)

[Some commonly used model evaluation techniques: 17](#_Toc82181067)

[Which feature has contributed more to a model’s prediction 18](#_Toc82181068)

[Model evaluation through ROC curve: 19](#_Toc82181069)

[Programing tools and libraries 19](#_Toc82181070)

[Data understanding through exploratory data analysis: 19](#_Toc82181071)

[EDA reveals data quality issues - A challenge to deal with: 21](#_Toc82181072)

[Addressing data quality and preparing data for further analysis: 22](#_Toc82181073)

[Deriving new features that could be potentially contributing to better prediction: 22](#_Toc82181074)

[Data Imputation and cleaning: 23](#_Toc82181075)

[Summarizing feature selection and elimination 26](#_Toc82181076)

[Data transformation for making the features compatible with the learning algorithms: 27](#_Toc82181077)

[Reflection on the exploratory data analysis and insights obtained from visualizations, findings, and discussion 28](#_Toc82181078)

[Setting a success criterion for classifier prediction and baseline accuracy 32](#_Toc82181079)

[Model building through supervised machine learning: 32](#_Toc82181080)

[Logistic Regression: 32](#_Toc82181081)

[Logistic regression model evaluation: 33](#_Toc82181082)

[Random Forest Classifier 35](#_Toc82181083)

[Confusion Matrix 36](#_Toc82181084)

[Performance metrics 36](#_Toc82181085)

[ROC-Curve and AUC value 37](#_Toc82181086)

[Feature importance 37](#_Toc82181087)

[Naïve Bayes 38](#_Toc82181088)

[Confusion matrix 38](#_Toc82181089)

[Performance metrics 38](#_Toc82181090)

[ROC-Curve and AUC value 38](#_Toc82181091)

[Feature importance 39](#_Toc82181092)

[MLP Classifier 39](#_Toc82181093)

[Comparison of Different models 41](#_Toc82181094)

[Survival analysis 41](#_Toc82181095)

[Redefining dropout: 41](#_Toc82181096)

[Survival models 42](#_Toc82181097)

[COX model for finding the influence of individual features on attrition: 45](#_Toc82181098)

[Hazard ratios 46](#_Toc82181099)

[Conclusion 47](#_Toc82181100)

[Study limitations 47](#_Toc82181101)

[Ethical approval: 48](#_Toc82181102)

[References: 48](#_Toc82181103)

[Appendices: 54](#_Toc82181104)

[Dataset dictionary 54](#_Toc82181105)

[Hazard ratios COX model 55](#_Toc82181106)

[python code 61](#_Toc82181107)

[Importing libraries 61](#_Toc82181108)

[Loading dataset 61](#_Toc82181109)

[Data Cleaning 61](#_Toc82181110)

[EDA 62](#_Toc82181111)

[Commissionaire 62](#_Toc82181112)

[Program Name 62](#_Toc82181113)

[*probability of Dropout or completer by program Size* 62](#_Toc82181114)

[Logistic Regression 67](#_Toc82181115)

[Random forest: 68](#_Toc82181116)

[Naïve bayes 70](#_Toc82181117)

[Multilayer perceptron: 72](#_Toc82181118)

[Code for Survival Analysis 74](#_Toc82181119)

[Supervisor meetings 76](#_Toc82181120)

[project approval and ethics 77](#_Toc82181121)

[Data Sharing agreement for this project 77](#_Toc82181122)

# ABBREVIATIONS/ACRONYMS

BMI - Body mass index

MLP – Multilayer perceptron

ANN – Artificial neural network

ROC – Receiver operating characteristic curve

AUC – Area Under Curve

EDA – Exploratory data analysis

CRISP-DM – Cross Industry standard for data mining

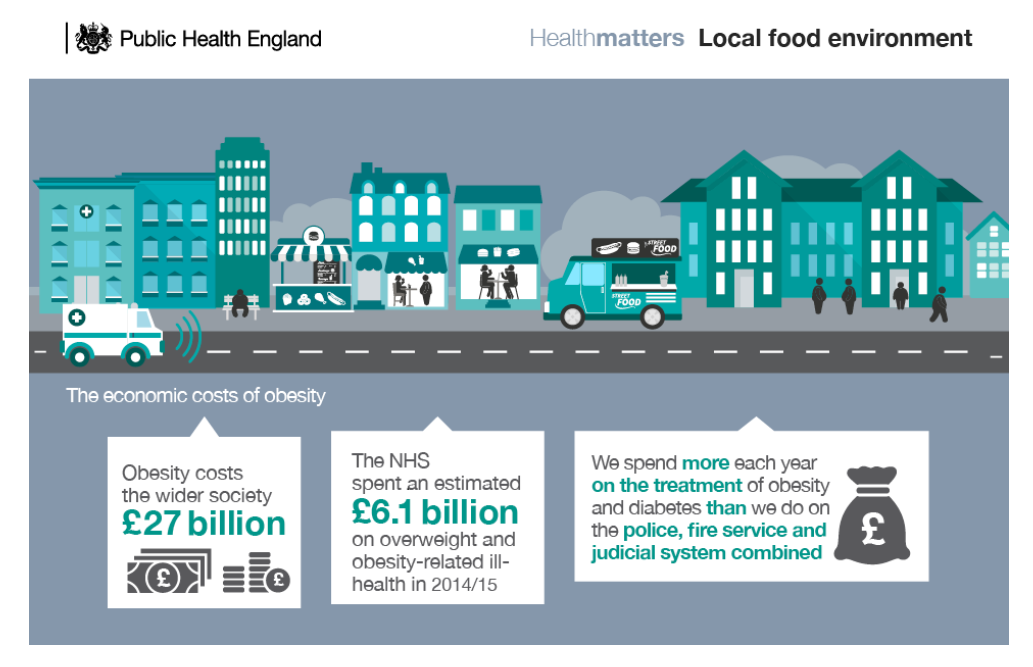
KDD – Knowledge discovery in databases

SEMMA - (Sample, Explore, Modify, Model, Assess)

# Obesity and Its health consequences

*Obesity* is a health condition that affects both children and adults. It occurs due to excessive body fat, which is 20% more than the ideal body weight, and can severely affect the quality of one’s life and mental health status. An obese person is characterized by a higher body mass index (BMI). According to the National Health Service (NHS), BMI is defined as the ratio between body weight in kilograms and the square of the height in meters. BMI in adults usually ranges from 18.5 to 24.9 (NHS, 2021), whereas it depends on age for children. According to the world health organization and NHS, obesity is classified into two categories based on one’s body mass index. For adults, a BMI more than 25Kg/m2 is considered overweight, and that above 30Kg/ m2 is considered obese (World Health Organization, 2021). Other measures of body fat include waist circumference, body fat ratio, and waist to hip ratio. Obesity increases the risk of type2 diabetes, cancer, and heart disease (Obesity, 2021). Other complications like osteoarthritis, high blood pressure, and cancers are also linked with Body mass index (BMI) levels above 25 (Thomas et al., 2011). Obesity harms adults and leads to other issues like lower employment chances, Discrimination, and social stigmatization decreased life expectancy and increased risk of hospitalization.

## The scale of Obesity and its financial burden to the UK

Obesity is a significant problem among individuals. About 30% of children in the UK below the age of 11 are overweight or obese (Baker, 2021). A report from the Organization for Economic Co-operation and Development (OECD) shows that 64.2% of the population in the UK with age above 15 are either overweight or obese (OECD, 2021). It is estimated that the NHS has spent 6.1 billion pounds in the treatment of obesity and overweight-related conditions in 2014-2015, and the project cost is about to reach above 9.7 billion pounds by 2050 (GOV.UK, 2021). Obesity not only affects an individual lifestyle but also has enormous financial costs in its treatment. Recently the government of the UK has taken response measures through an anti-obesity drive to place a ban on advertising sugary foods after 9 pm (**Parkinson J**, 2021). According to UK-GOV estimations, obesity is related to 30,000 deaths each year, reduces an individual’s lifespan by nine years on average (GOV.UK., 2021)

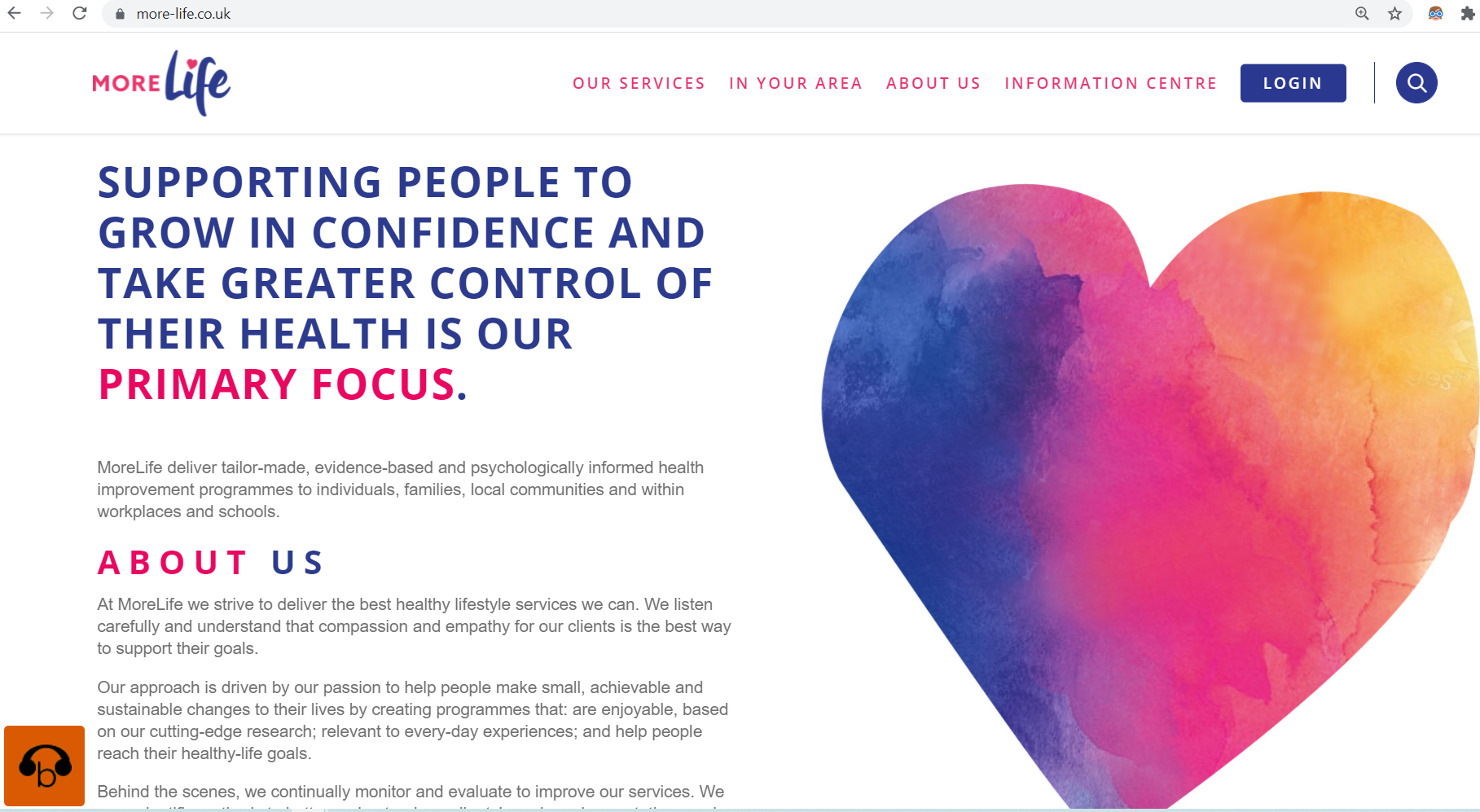
**Image source (gov.UK, 2017)**

[*https://www.gov.uk/government/publications/health-matters-obesity-and-the-food-environment/health-matters-obesity-and-the-food-environment--2*](https://www.gov.uk/government/publications/health-matters-obesity-and-the-food-environment/health-matters-obesity-and-the-food-environment--2)

## Life-style Management services and the role of MoreLife to address the concern.

Treatments for obesity include exercise, medication, surgery, and utilizing services provided by local support groups. A decrease in weight of around 5 to 10% of the bodyweight helps reduce the complications associated with obesity and its associated comorbidities (NHLBI, 2021). A frequently suggested treatment for obesity is to utilize the help provided by the local health groups and lifestyle intervention programs (Obesity- Treatment, 2021). Lifestyle management programs utilize diet and physical activity monitoring to help individuals overcome obesity.

## An overview of MoreLife



**Image source**: (MoreLife, 2021) [*https://www.more-life.co.uk/*](https://www.more-life.co.uk/)

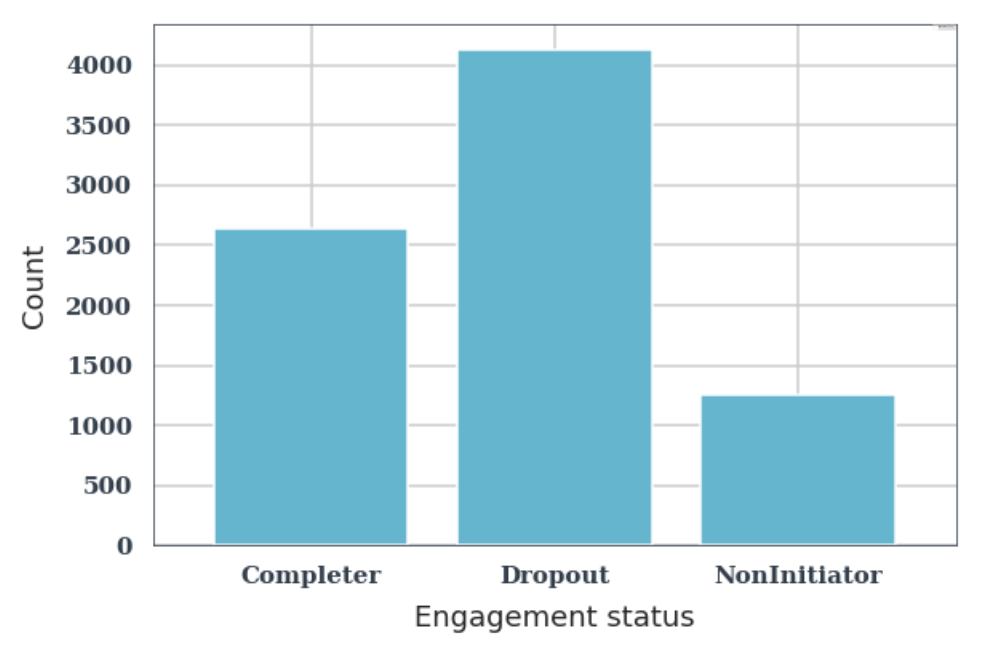
MoreLife is a community welfare organization with a mission to support people by building confidence and give them greater control of their health and lifestyle. MoreLife also offers weight management services to help overweight and obese adults and children. Weight management programs at MoreLife are usually held for 14 weeks at different locations in the UK. These divisions working at various geographical locations across the UK are called commissionaires. Participants utilizing MoreLife’s services include individuals from diverse gender, ethnicity and age groups.

## Early participant dropout a challenge to weight interventions and MoreLife

Lifestyle intervention programs were known to experience high dropout rates. About 45% of the participants are known to discontinue the treatment (Clark, M.M. et al., 1996). Causes for the treatment attrition are not well understood. Lack of participant's adherence to the program results in treatment failure and incurs financial losses for supporting organizations. Studies on attrition in eHealth interventions propose that unrealistic expectations due to miss information, intervention being an unpaid program, amount of work needed to notice benefits from an intervention can influence participant dropout (Eysenbach, G., 2005)

MoreLife’s weight management programs have a high dropout rate, with only one-third of the participants completing the program. Low participant retention is causing partial failure of their programs. Being a non-profit and community welfare organization exploring possible ways to understand attrition is paramount to MoreLife’s mission.

**Distribution of participants at MoreLife by their Program engagement status**



**Figure: Participant outcomes of the adult weight management programs conducted at MoreLife**

**between 2011 and 2018. Bar plot obtained through data visualization on adult dataset**

The ability to predict who is going to drop out from an intervention is an interesting problem. Predicting participant success or dropout status in advance helps MoreLife tailor its programs to suit individual participants and pay more attention to those at risk of dropping out of the program. Being relied on public funding sources and its reduction over time has put MoreLife under financial pressure leading to the discontinuation of its summer camp program for obese children (MoreLife, 2021). Thus, cost optimization solutions by cutting losses are also an important goal. This project aims to understand obesity treatment attrition by analyzing data through data visualization and utilize techniques in machine learning to build models to predict participant dropout.

By predicting the participant's outcomes (dropout or completer) in advance, customized strategies can be developed and suggested for individuals. Predicting participant's outcomes is a complex problem involving human behavior, socioeconomic status, and many other known and unknown factors (Skelton, J.A. et al., 2011), (Skelton, J.A. and Beech, B.M., 2011), (Burgess et al., 2017). One way to approach this problem is to draw insights from the available data on participant's behavioral, socioeconomic and other physiological information. Machine learning models can show predictions by combining all parameters as inputs in contrast to inferences made by understanding the influence of one variable at a time.

A previous study conducted on participant's engagement in MoreLife weight management with a participant sample size of 2948 (James Nobles et al., 2016) reported that features that include participant characteristics are weak predictors of attrition. In contrast, the program's features, including the participant group's size, the season of program delivery, are also known to affect participant attrition. The research also suggests that limiting a program to 20 participants may improve participant retention and successful engagement. The features corresponding to the program included in that study are participant group size, program year, program length, delivery period, and age. The study also shows that higher BMI SDS and IMD(index of multiple deprivations) Scores are associated with non-completion. In addition, nonwhite participants are at risk of not initiating the program.

Previous studies on predicting attrition are minimal. Most of the previous studies have focused on understanding attrition in lifestyle intervention through descriptive and inferential statistics (Germann, J.N., 2006) (Tollefson, D.R. et al., 2008) (Fabricatore, A.N., et al., 2009) Only a few predictive modeling studies through machine learning on attrition were conducted in eHealth and app-based interventions. This study uses supervised learning techniques to predict attrition using random forests, Naïve Bayes, MLP, and logistic regression.

With the digitalization and increase in data availability, artificial intelligence and techniques in machine learning have gained wide popularity in their applications to an extensive range of problems in almost all basic sciences, technology, finance, and other sectors. The health care industry has also seen a rise in machine learning and AI applications in medical image recognition, drug design, clinical judgment, decision support. (Lavecchia, A., 2015). Medical images from patients collected by NHS are being used to determine health risks with the help of google deep mind (Bali, Garg, and Bali, 2019). Machine learning has also has found in its applications in predicting weight loss in obesity management programs. (Babajide et al., 2019) (Kim, Eunjoo, et al., 2020). There is also progress in applying machine algorithms in predicting attrition in lifestyle intervention programs (Pedersen, D.H. et al., 2019). Thus, data collected from More Life’s participants may provide insights and help to reduce the dropout rates in obesity treatment.

## Project Aim

This project aims to build machine learning-based models to predict the dropout or success outcome for MoreLife weight intervention programs.

## Projective Objectives

1. To conduct a literature review to understand the role of machine learning in weight management programs.
2. To identify a suitable project management approach that helps analyze MoreLife’s adult participants dataset for predictive modeling.
3. To conduct an exploratory data analysis following the project management approach and appreciate the findings and insights obtained from data.

1. To preprocess the data addressing its quality issues.
2. Identify suitable machine learning techniques following the exploratory data analysis and implement the proposed machine learning models.
3. Evaluate the machine learning model performances using appropriate metrics.
4. To do survival analysis to understand participants program survival chances during each week of the program
5. Present and provide a critique on the results and findings

# Into the role of machine learning in weight management a Literature Review

Machine learning is a part of Artificial intelligence that involves studying algorithms that learn from data and improve their performance on a task with training (Machine learning - Wikipedia, 2021). Unlike the standard algorithms that are explicitly programmed to perform a task by following rules and instructions, machine learning algorithms perform a task after learning the required rules to perform the task. Learning methods were first used in the 1950s for playing checkers (Samuel, A.L., 1959) at IBM. The development of the perceptron model (Rosenblatt, F.,1958) at Cornell Aeronautical Laboratory and nearest neighbor algorithm (Cover, T. and Hart, P., 1967) has led to further advancements in machine learning. Today machine learning algorithms are often used in medical image diagnosis (Anwar et al., 2018), disease identification (Li, J.P. et al., 2020), analyzing sales data, and dynamic pricing.

Businesses have relied on data analytics for decision support taking advantage of statistics to understand the trends in data. This is generally achieved by creating reports and visualization through dashboards. Some of the intermediate stages in data analytics include calculating mean values, ratios, and percentage values of a performance indicator. While data analytics focuses on discovering trends in historical data and answering focused questions, machine learning algorithms focus on finding answers to questions that are yet to be asked (springboard, 2021). Machine learning has an extensive range of applications in almost all sectors, including health care, journalism, banking fraud detection, and drug discovery (Vamathevan, J. et al., 2019). For example, in health care, machine learning has been applied in tumor detection (Hodneland, E et al., 2021), lung cancer classification (Bergquist, S.L. et al., 2017), and classification of types of rheumatoid arthritis (Orange, D.E., et al., 2018).

Machine learning research in weight management can be classified into two classes of studies. One class of studies focuses on predicting participant's survival and their lack of adherence to weight intervention programs leading to early dropout. Another class of studies focuses on discovering the predictors that influence weight loss. As it is known that the response to weight loss treatments is not consistent among different individuals as they respond differently to weight loss treatments (Nielsen, R.L. et al., 2020), identifying them can help in designing personalized treatment and interventions. In this study, we only focus on studying attrition.

Supervised machine learning algorithms are known for learning from label data and are generally found to be help full in classification problems. Predicting the participant's engagement is a binary classification problem. A study conducted on an e-health lifestyle intervention program in Denmark focusing on weight loss or behavior change outcomes shows that dropout prediction can be made with high precision in eHealth interventions with the help of the random forest’s method (AUC 0.92) (Pedersen, D.H. et al., 2019). The input features include a measure of participant’s activity on the eHealth platform, weight loss, age, and gender. A decrease in user activity on the eHealth platform is found to be a good indicator for predicting dropout. This could be a reflection of a lack of interest or confidence. This suggests that maintaining regular interaction with the participants may help reduce dropout. It is also found that attrition is related to program features. This is consistent with previous studies on understanding MoreLife’s programs. Similarly, demographics are found to have less influence on dropout prediction. Also, the study finds that most of the program dropouts had discontinued in the first two weeks. Finally, the project has proven to predict dropouts with high accuracy. Few issues with this study are that it has a limited sample size, and it is primarily restricted to online platform-based intervention programs.

Weight interventions are designed for the well-being of overweight individuals. Even though this is in their best interests, dropping out of the program indicates that other behavioral components are responsible for this. A study focused on understanding participants engagement in an app-based weight intervention program through cognitive behavioral therapy has included factors related to behavioral, motivational, cognitive, and emotional information as inputs for the elastic-Net regression model (Kim, M. et al., 2021). The information about these factors is collected through a questionnaire. It is found that factors like lower self-esteem, confidence, higher interaction time with a mentor are good predictors of participant engagement. Thus, conducting a questionnaire to determine participants' motivational and emotional factors ahead of a program could be utilized to predict participant engagement with better accuracy. However, though this study shows the importance of individual psychological factors for programs adherence, it is limited to only 45 participants. Such a small group of participants makes the statistical approaches unreliable. Also, this study is conducted only on patients who received cognitive behavioral therapy, which implies that this study may not apply to the general participants in lifestyle intervention programs.

Studies on understanding attrition in weight interventions through machine learning techniques are limited as most of the research focuses on understanding weight loss compared to attrition. Few studies on factors affecting attrition were done in the past, which focused on understanding attrition through descriptive and inferential statistics. A literature review study (I. Moroshko et al., 2011) that examined 61 articles that were published before 2011 and focused on predictors of dropout from different weight loss intervention treatments reported that patient background characteristics were not strong predictors of attrition. In contrast, psychological and behavioral features are found to be commonly linked to treatment attrition. As attrition is linked to behavior, it is also essential for researchers to utilize relevant psychological and behavioral features to understand attrition through machine learning. Features like age, employment status, gender, ethnicity, and marital status were not consistent in predicting attrition. It is also found that higher age and educational qualifications are associated with lower attrition. Other factors identified are higher expectations of body mass reduction, lower initial weight loss, and treatment cost. Variables of demography like gender are found to be unreliable in predicting treatment attrition. This motivates to employ machine learning techniques to determine if demographic variables are essential in predicting attrition.

# Selecting an appropriate project management methodology.

This section explores various project management methodologies and relevant predictive modeling methods through supervised machine learning models, model performance evaluation, data quality exploration, and cleaning.

Finding a suitable project management approach has become significant in software project development for avoiding costly mistakes which could lead to project failure. A project methodology consists of tools, processes, techniques, standards, and a series of steps for accomplishing the project aims (Cockburn, A., 2000). The role of this project is to build multiple predictive models through supervised learning algorithms for predicting attrition in lifestyle management programs. A single predictive model cannot be assumed to be the best model before knowing the specific problem, available data, and quality issues. Different machine learning models should be implemented and compared to find a better-performing model. This idea is reflected in the no-free lunch theorem (Wolpert, D.H. and Macready, W.G., 1997).

## A comparison of various project management methodologies:

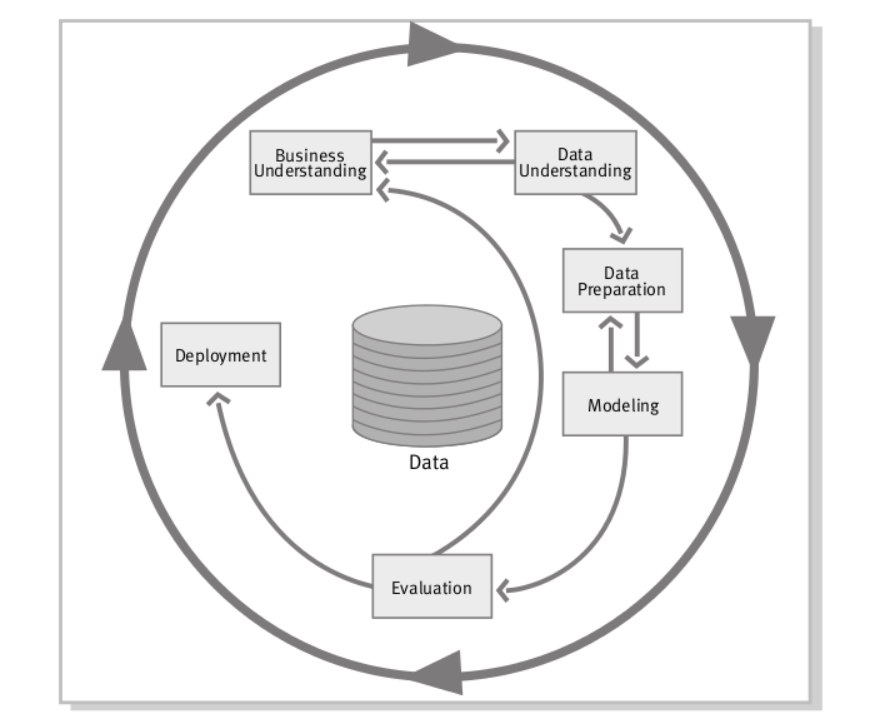
To achieve the project's aims, it is necessary first to examine the data, address its usability and quality issues, and transform it suitably and consistently with the learning models' requirements. This is a different phase that is not usually found in traditional software development life cycle and methodologies (Ruparelia, N.B., 2010). Traditional software development consists of phases like designing, testing, and deploying. The well-known agile scrum methodology is not very suitable for this project because it’s challenging to plan sprints without clearly understanding the client requirements (Saltz, J. and Crowston, K., 2017). client’s requirements may need to be assessed after data understanding, and data understanding requires data exploration. Therefore, an essential step a project management methodology should include is understanding data and then thinking of how it can address business requirements. This is followed by further narrowing the business requirements and set realistic goals after data understanding.

This project involves various phases like data gathering, data cleaning, data exploration, visualization and analysis, data modeling. Some of the successful and popular project management methodologies considered for this project are CRISP-DM (Cross Industry-standard process for data mining), KDD (knowledge discovery in databases), and SEMMA (Sample, Explore, Modify, Model, Assess). All three process models have phases that are similar and comparable to each other. SEMMA was developed with a primary intention to help the users of SAS software, focusing on business intelligence and statistics (SEMMA - Wikipedia, 2021). It is a step-by-step sequential process for data acquiring, transforming, and analyzing. Though SEMMA covers the essential phases of a data mining task, the main disadvantage of this process model is that it does not emphasize much on a business understanding of the project. CRISP-DM has a strong emphasis on data and business understanding, and it has become the most popular choice for data mining projects. CRISP-DM is the most frequently used project management methodology in data science projects by companies (Marbán, Ó, et al., 2009) and (Nadali, A et al., 2011). The KDD process model is very similar to crisp-dm. The individual stages of the KDD process can be mapped to different phases of CRISP-DM. (Shafique, U. and Qaiser, H., 2014). KDD has some resemblance to a waterfall model, while CRISP-DM is a better process model for this project due to its cyclic phases of project advancement, especially in the data understanding and business understanding phases.

### The project approach through CRISP-DM

CRISP-DM is the most widely used process model in data mining projects. The earliest version of the process model was presented in 1999 at the crisp-dm sig workshop (Chapman, P. et al., 1999). This methodology includes six phases in its project life cycle, namely

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. evaluation and deployment.



**CRISP-DM project Life cycle**

CRISP-DM project life cycle (CRISP-DM 1.0, Pete Chapman., et al.)

This project began with a rudimentary understanding of the data MoreLife is going to share for research. The requirements discussed for the project are the possibility of predicting attrition or understanding weight loss through machine learning models. The project has to set realistic objectives based on the available data—this required data understanding through exploratory data analysis. After EDA, the project focused on attrition rather than predicting weight loss as severe data quality issues were found with the weight measurements taken after each program week. More than 30% of the values are missing after the program's second week, and these missing values in measured weights kept increasing with successive weeks. Moreover, weight loss modeling has only shown better results after including features related to diet, genetics, and other biomarkers (Nielsen et al., 2020). With this understanding, the project objectives were refined to understanding attrition.

Business understanding is linked to every phase in CRISP (Sharma, S. and Osei-Bryson, K.M., 2009). This project found that the classification models performed better when missing values were replaced with a value like ‘missing’ or ‘not known’ compared to their performance after removing the data that contained missing elements as this reduced sample size. Data quality issues and the pre-processing methods have influenced the modeling technique's performances. This is the interplay between the data modeling and data preparation phases of CRISP-DM. The modeling and evaluation phases of the project have been explained in the later sections. The project is not entirely according to CRISP-DM as model deployment is not the focus of this project, yet it has gone through most of the phases of CRISP.

## A brief introduction to predictive modeling through supervised machine learning approach, statistical analysis, and model validation

Modeling and evaluation are essential phases of this project. The modeling phase involves selecting techniques, recording model assumptions, setting a success criterion for the model performance, designing procedures to build and test a model’s performance, further assessing the model, and improving its performance. The below section explains the methods designed for implementing these processes.

Predictive modeling is a set of techniques used to identify patterns in data (Kuhn, M. and Johnson, K., 2013) and make predictions based on them. Classifiers in predictive modeling are statistical techniques that classify data based on a target or a class variable. Commonly used classification models are decision trees, logistic regression, Naive Bayes, support vector machines (Soru, T. and Ngomo, A.C.N., 2014). These classifiers in machine learning can be trained on the historical data to make predictions on unseen data. This predictive capability on unseen data makes learning algorithms generalizable and valuable. This project uses logistic regression, Random Forests, Naïve bayes and mlp classifiers. These are some of the widely used classification algorithms and are found their applications in many classification projects.

The process of making a model to learn from historical data is called model training. Generally, a data set is first split into training and testing samples (Tan, J. et al., 2021) to train a supervised learning model and then measure its performance through evaluation metrics on the test sample. Sometimes a model learns well on training data while failing to predict on testing data accurately. This is called overfitting. Overfitting makes the models unreliable. For ensuring model reliability, other procedures are followed for training.

### Ensuring model reliability and resampling through k-fold Cross-validation:

Machine learning models sometimes fail to make accurate predictions when tested on unseen data. This happens if a model overfits the training data's patterns and noise and cannot generalize enough to predict unseen data. To avoid this, a data resampling technique called cross-validation is used. Training a model through cross-validation reduces overfitting (Moore, A.W., 2001).

In the k-fold cross-validation method, the training data is resampled into k disjoint subsets, with the model being tested on one of these subsets after being trained on the remaining k-1 subsets. This process is repeated k-times by selecting a unique subset each time for testing after training through remaining k-1 samples (Anguita, D. et al., 2012). The final performance is obtained by averaging the accuracies of the k models. To ensure a proportionate distribution of samples in the binary class, stratified k-fold cross-validation is used (Sklearn, 2021). The final model is tested by predicting the class of the samples belonging to the test sample. There are various metrics to test a model's performance.

### Methods to test and evaluate model performance:

There are five commonly used evaluation metrics in a classification problem with binary target variables for testing the model performance. They are accuracy, precision, sensitivity, F1- score, and AUC value, the area under the ROC curve.

### Some commonly used model evaluation techniques:

#### Confusion matrix:

A confusion matrix presents the predictions of the binary class as four numbers that represent the performance of a class, i.e., It classifies predictions as true positives, false positives, true negatives, and false negatives

***Actual***

|  |  |  |
| --- | --- | --- |
|  | Dropout | Completer |
| Dropout | TP (True dropout) | FP (Falsely predicted dropout) |
| Completer | FN (falsely predicted as completer) | TN (True completer) |

*Predicted*

#### Accuracy:

It is the total fraction of the correct predictions. It gives how an idea of how good is the model predicting unseen cases.

Accuracy = (TP+TN) / (Total number of predictions)

#### Sensitivity or recall and precision:

Sensitivity is the fraction of the correctly predicted dropouts from the total dropouts. Precision tells us that of all the dropout predictions, how many are actual dropouts. (Chicco, D. and Jurman, G., 2020)

Sensitivity = TP / (TP + FN)

Precision = TP / (TP + FP)

While these metrics help compare different models and evaluate their performance, the factors that influence a model’s prediction are generally not straightforwardly understood. The importance of a feature in its contribution to a model prediction is called feature importance.

Which feature has contributed more to a model’s prediction?

Feature importance measures are usually dependent on the model used for prediction. Therefore, every model has its own measure for feature importance. Moreover, these measures have their strengths and weaknesses. In this project, permutation feature importance is used for understanding feature importance.

Permutation feature importance is defined as the reduction in the model performance score when one of the column values in the tabular data corresponding to a feature are shuffled randomly (scikit-learn 0.24.2, 2021)

Importance = Model Score – Model Score after a feature shuffle

The more the reduction in a model score for a given feature, the higher is its contribution to the model’s prediction. Calculating feature importance can be applied to any model because it depends on tweaking the data and not the model itself. For example, tree-based models have other important scores like mean decrease impurity. However, these methods give the wrong importance to features that are not important when the model is subjected to overfitting. Moreover, they are biased towards numerical features over features with low cardinality (scikit-learn 0.24.2, 2021). Permutation feature importance avoids such problems. One drawback of this method is that it under-represents a feature regardless of its importance when a strongly correlated feature exists in the input feature set.

### Model evaluation through ROC curve:

The classification threshold value a model chooses to classify a sample into a binary class is not always its best. For example, the outcome can be classified in logistic regression based on setting the threshold outcome probability 0.5 for model decision. Nevertheless, this may not always give the best results when predicting unseen samples. To find the best classification threshold, a graph between the true and false positive rates is plotted by changing the classification cut-off. This curve is called Receiver operating characteristics (ROC) (Hoo, Z.H. et al., 2017). A model with an AUC value close to 1 is considered a better model, and a model with an AUC closer to 0.5 is considered to have no predictability (Johnson, 2021).

Before implementing and evaluating the supervised models, the data needs to be explored using suitable software packages and tools.

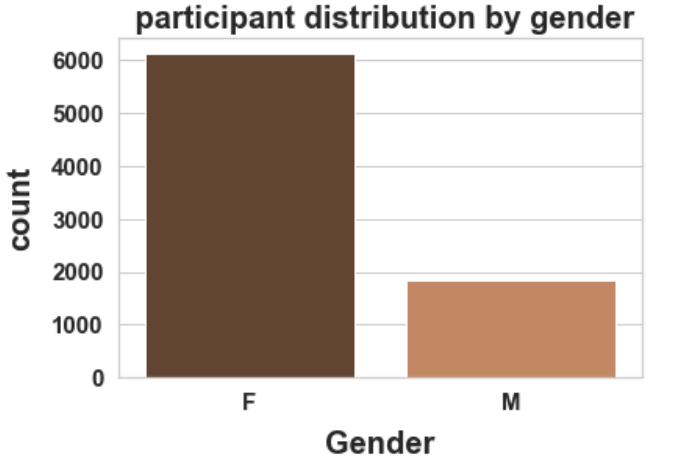
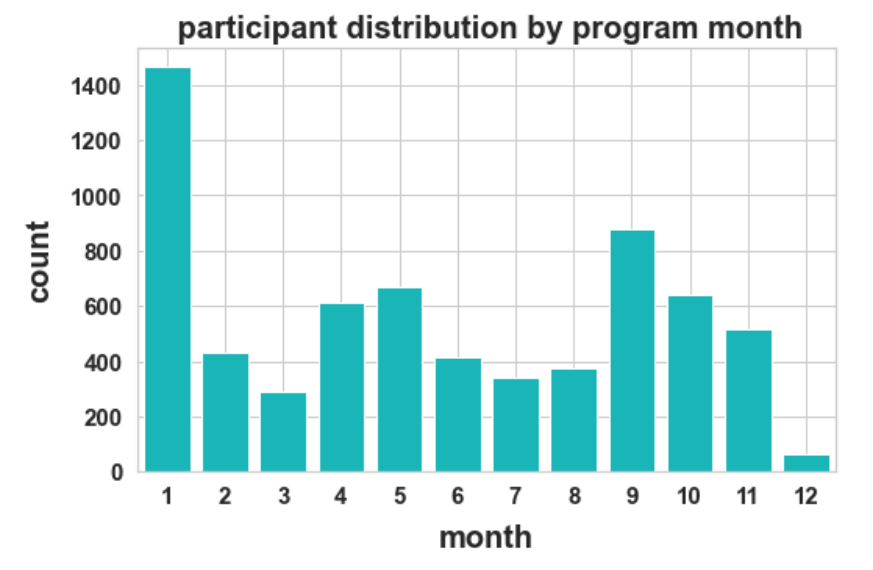
## Programing tools and libraries

In this project, the following libraries in the python-3.8.11 programing language are utilized for data processing, visualization, and building models.

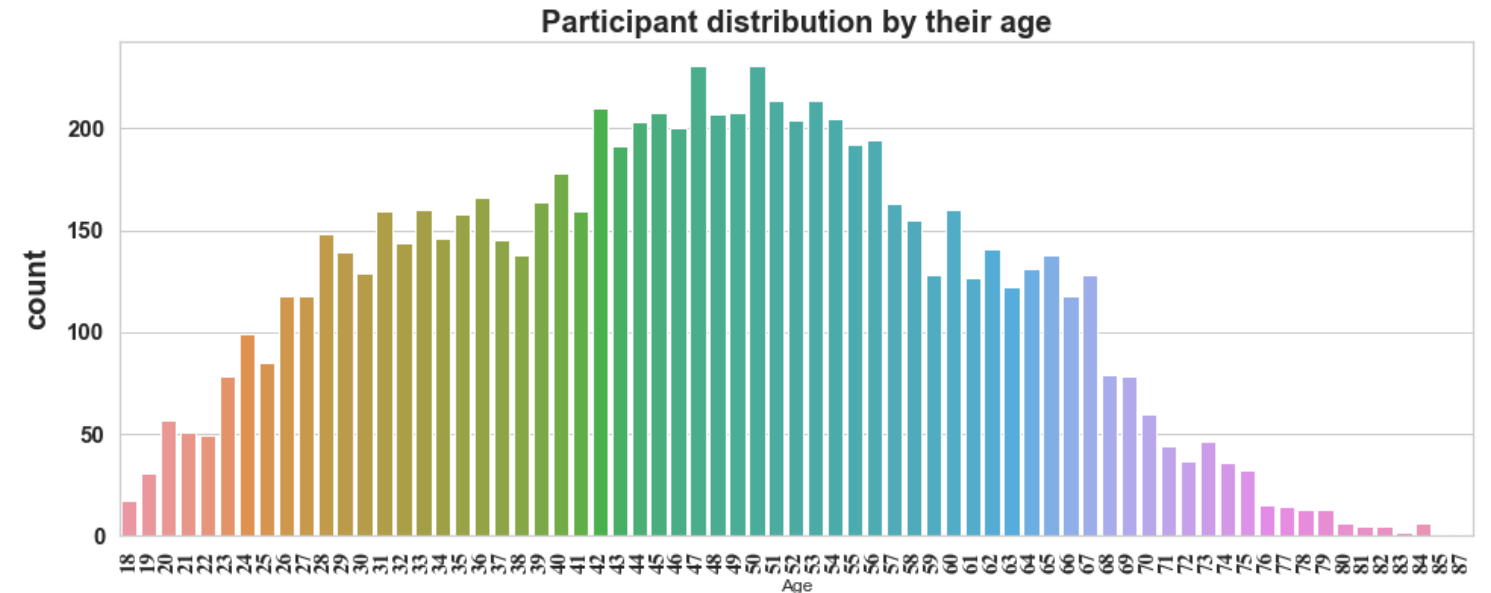
NumPy 1.19.1, Pandas 1.0.5, Graph viz 2.38, seaborn 0.11.2, SciPy 1.5.4, matplotlib 3.4.2, lifelines 0.26.0, scikit-learn 0.23.1

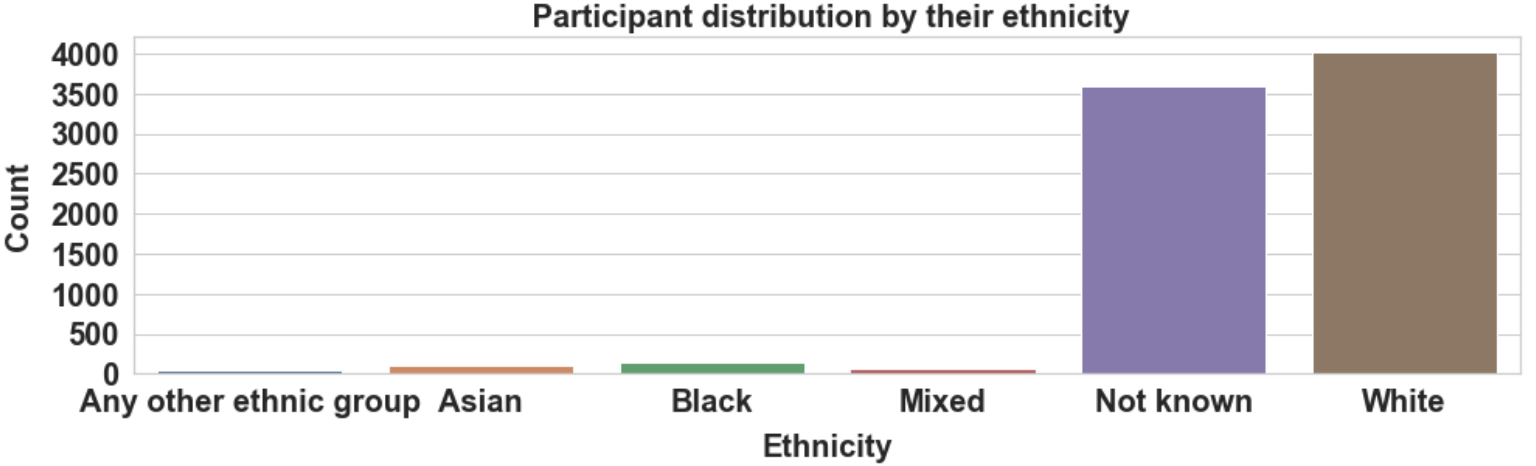
## Data understanding through exploratory data analysis:

Exploratory data analysis gives quicker insights into data through visualizations. This helps to understand data and refine project objectives and help in identifying suitable data mining methods and tools. Participant data is obtained from MoreLife after a data-sharing agreement between MoreLife and the Leeds Beckett University. The dataset is an EXCEL file which consists of information of the individuals who had participated in weight management programs conducted at MoreLife between the years 2011 and 2018 at various MoreLife commissionaires across the United Kingdom. There are a total of 7476 unique adult participants of various Ethnicity, gender, and age groups. The adult data set consists of 8029 rows and 50 columns, with each row consisting of information about the clients and the MoreLife program features. The female to male participation ratio is 6148: 1881. The number of participants in January is double that the monthly average number. This could be due to a new year resolution for losing weight.

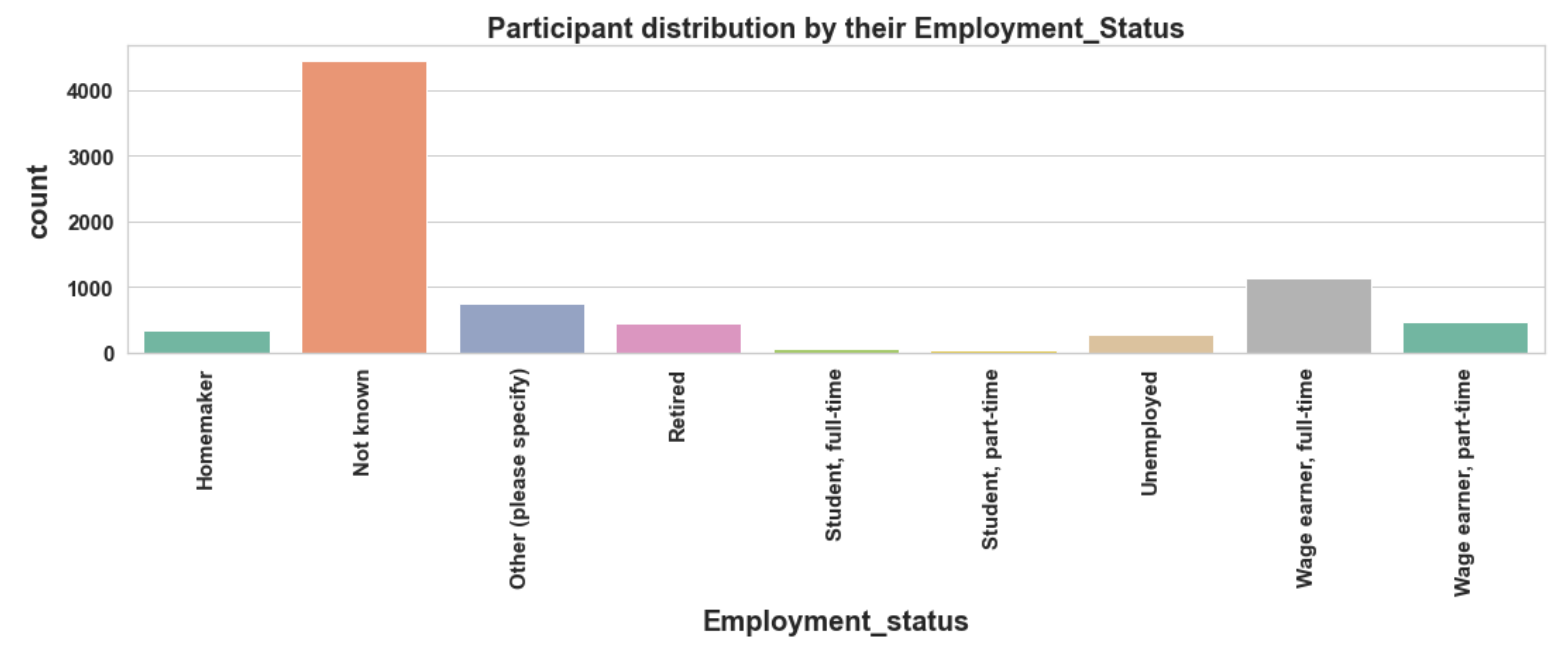


There are 4065 instances of white ethnic population, 141 Black, 104 Asian, 63 Mixed, and the rest are missing. Mean age 46.8 and standard deviation 13.6.





A typical program length is 12 to 14 weeks, with an average of 17 participants per program SD 13.6. Most of the participants are wage-earners 1618, homemakers 342, and retired individuals 460, 89 students. The rest are either unemployed or of unknown employment status.

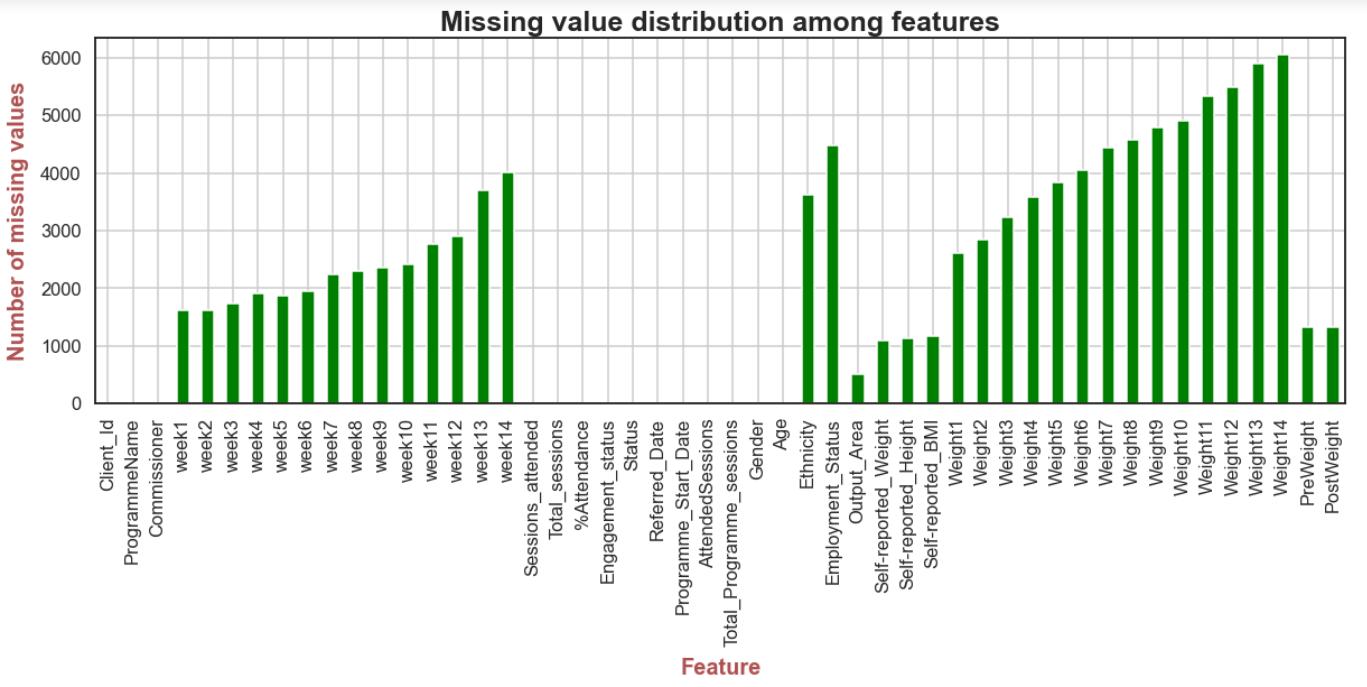


22 Commissioners across the UK hold the programs. Each participant is classified into completer, dropout, and non-initiator. Dropout is defined as a participant with an attendance of less than 75%. Only one-third of the participants have completed the program, while the rest are either dropouts or non-initiators.

The features in the dataset can be classified into three categories. First, there are categorical variables like age, gender, ethnicity, employment status, and location. These correspond to participant characteristics. The second category is the features related to a weight management program like program name, commissioner, Total program sessions, and program start date. The third category of features includes measurable variables like weight measurements and weekly attendances. Then a class variable engagement status that classifies attendees into dropouts, completers, or non-initiators category groups. More details about the definition of individual features and a data description report are given in appendix A.

## EDA reveals data quality issues - A challenge to deal with:

The dataset is tabular with rows and columns, with about 27.3 % of the cells having null values. The number of missing values per column is shown below.



There is a trend in the missing values in the feature's weekly attendances and weight measurements. The number of missing values is an increasing function of the week. These missing values do not look like they are missing at random. Their trend indicates that attendance and weight measurements are not recorded due to the program dropouts rather than the values missing at random due to human error in failing to record. This makes specific data imputation techniques inapplicable. There are several missing values in features ‘Ethnicity’ and ‘employment statuses’ with 45% and 56% of the data missing in their respective columns.

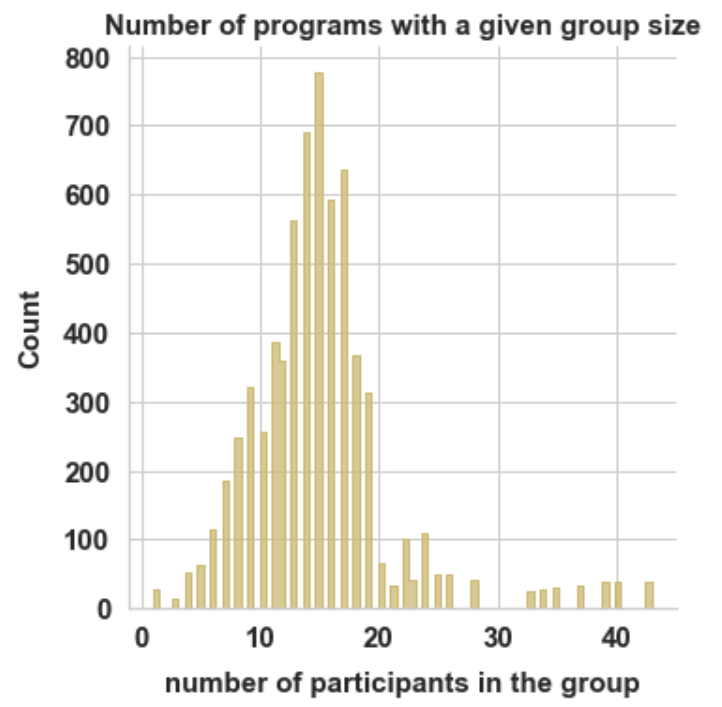
## Addressing data quality and preparing data for further analysis:

Following CRISP-DM, the data needs to be prepared before using it for building the models. Data quality is fundamental in data mining projects that impede knowledge discovery (Hand, D.J., 2007). Data quality issues arise due to Outliers, human error during data collection, and many other reasons. As the data set is smaller, deleting missing records will impact the statistical analysis and model performance. Data preparation also includes finding new features and removing unwanted features from the dataset.

### Deriving new features that could be potentially contributing to better prediction:

1. The program month is extracted from the column ‘program start date’ to understand if a program started in a given month effect participant engagement outcomes. This feature is included as data exploration shows a trend in participant engagement based on program month. Also, previous studies have shown that the MoreLife program’s engagement depends on the quarter of the year in which the program takes place.
2. A feature weight change in the first week is added, assuming participant's adherence to the program may be influenced by it. It could be possible that lower loss in weight could disappoint participants' expectations and result in the loss of confidence to continue the program further. This is also suggested by previous studies on understanding attrition (I. Moroshko et al., 2011). This feature is obtained by subtracting the weight after the second week ‘weight2’ from the weight recorded on the first week of the program ‘weight1’. Also, the average weight loss after the first week seems to be significantly different for dropouts and completers.

1. Previous studies suggest that limiting a program size to less than 20 participants may help reduce attrition (James Nobles et al., 2016). A new feature, “NumPerProgram,” is also derived from the existing features in the data set. Participants belonging to the same program name are identified and counted to obtain the number of participants per program group. The program name is found to be unique for a given program start date and a given commissioner. Program group size distribution is shown below.



### Data Imputation and cleaning:

As the data has many missing values, data cleaning is required. The missing values in the ethnicity are either replaced by ‘not known’ or by its value taken from another row if the client has attended multiple programs. The missing values in employment status are also replaced by ‘not known.’ This is done rather than removing the records containing the missing data to preserve the sample size. Initially, when data imputation is conducted by deleting missing records, it has reduced the accuracies of the predictive models. A trade-off of keeping the features with missing values reduces the reliability of the contribution of such features to the model. Maintaining a larger sample size is preferred as dropout prediction is the main objective of this project. The following steps are followed for reducing the number of features and removing unwanted samples from the data.

Sample size 8029

Features 50

1. Initial sample size and the initial number of features

1. All the applicants in the adult dataset are accepted for the program. The feature 'Status' is not helpful as it has zero variance. Therefore, this column is removed from the dataset. Also, it is assumed that the program referred date does not influence attrition and is removed. This is done after extracting the program month from it.

Sample size 8029

Features 48

1. The columns ‘sessions attended’ and 'Total\_sessions' are duplicates of the columns ‘AttendedSessions’ and ‘Total Program sessions’; hence they are removed. Similarly, the features weight1 and ‘preweight’ are also duplicates, and the column “preweight” will be dropped. However, before this, some of the missing values in weight1 are filled with the corresponding values from the ‘preweight’ column.

Sample size 8029

Features 45

1. The measured weights after the first week have more than 30% or more values missing per column. Thus, only weights measured in the first week are included. Weights from weight2 to weight14 are removed.

Sample size 8029

Features 32

1. The values in the numerical columns like ‘weight1’ and ‘height’ are considered as outliers if their distance from their mean is more than three times their standard deviation. Such rows are removed from the dataset.

Sample size 7952

Features 32

1. Self-reported weights have high variation and differ from measured weights by 3Lb on average. Therefore, they are generally unreliable (Rowland, 1990) and are removed from the dataset instead; measured weights are included. Although BMI is correlated to weight and height, it is removed because learning algorithms perform when input features are independent.

Sample size 7952

Features 30

1. Some participants have not initiated the program. Non-initiators are removed from the dataset. Furthermore, the Output area is removed as the samples for a given output area are sparse, and it has ID ness characteristic or huge variance. A lookup table to map output areas was not found to obtain other features. Therefore, the output area is removed. Similarly, the client ID is also removed. Finally, the program name is unique to the program date for a given commissioner and is removed.

Sample size 6698

Features 27

1. Weekly attendances, attended sessions, attendance percentage, post-weight is not known in prior and cannot be predictors, and hence they are removed.

Sample size 6698

Features 10

1. Engagement status is a predictor class variable and not a feature. The column program start date is replaced by program month.

Sample size 6698

Features 9

1. Features ‘program size’ and weight loss after the first week are derived from other features. Furthermore, rows containing null values in the selected 11 features are removed, reducing the final sample size to 4501.

Sample size 4501

Features 11

### Summarizing feature selection and elimination

The missing values in ethnicity and employment status are replaced by ‘not known’ rather than deleting the corresponding rows, and later these columns are removed to create multiple datasets. Models are studied in both cases. For clients who attended multiple programs, their ethnicity values are obtained from their ethnicity recorded before or later.

|  |  |  |
| --- | --- | --- |
| **Features derived** | **Features included** | **Features removed** |
| 1. Program month, 2. The number of participants per program. 3. Weight loss after first and second weeks. | 1. Commissioner 2. Initial weight 3. Gender 4. Age 5. Ethnicity 6. Employment status 7. Month 8. Total program sessions 9. The number of participants per program. 10. Weight loss after the first week. | 1. Client id (High variance) 2. Program name (high variance and correlation with program start date) 3. Weekly attendances (not known in advance and hence cant be predictors) 4. Sessions attended (same as above) 5. Total sessions (same as above) 6. Percentage attendance (same as above) 7. Status (Zero variance) 8. Referred date (high variance) 9. Program start date (month is extracted) 10. Attended sessions (not known in advance and hence cannot be predictors) 11. Output area (no lookup table found, very high variance) 12. Self-reported weight (errors and unreliability) 13. Self-reported BMI (errors and unreliability) 14. All weight measurements from the second week of a program (more than 30% values missing) |

Data transformation for making the features compatible with the learning algorithms:

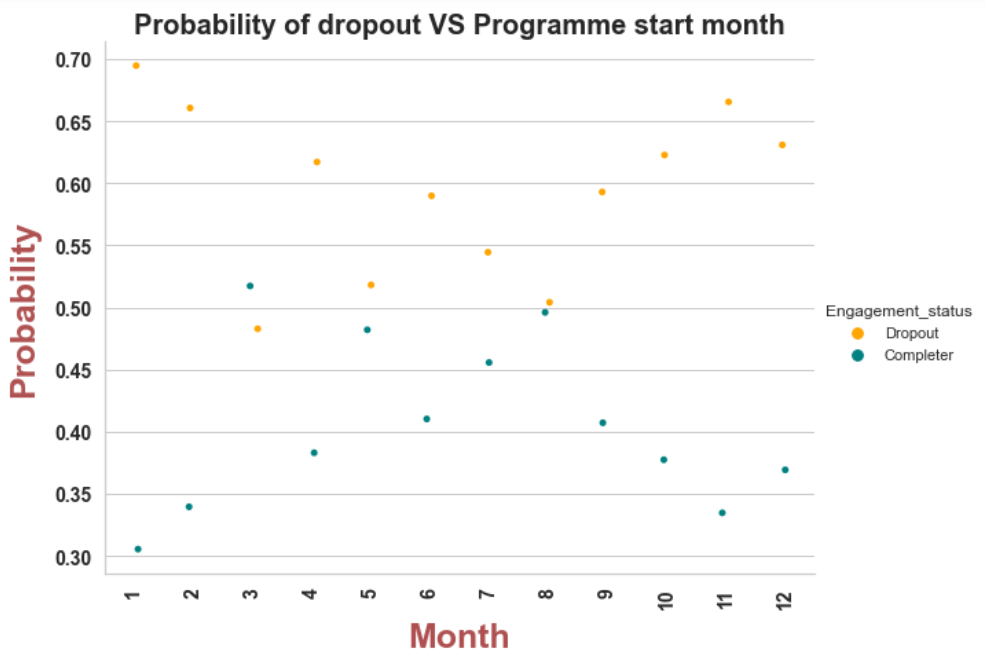
Many learning algorithms often cannot handle categorical variables in the training process and expect input variables to be numeric. For example, in tree-based models like random forests, features are compared with a numerical value when splitting a sample set into two disjoint subsets of samples at a node. Moreover, random forest estimators in the sci-kit learn library does not support categorical features (sklearn, 2021). In an artificial neural network, the output of a neuron is a linear combination of the input values, which also requires the features to be in numerical format (Kukreja et al., 2016). For this generally, encoders are used to convert categorical variables to numerical variables. One way to encode a feature, for example, ‘program month,’ is to replace January, February…. etc., with natural numbers like 1,2,3, etc. This is called label encoding. However, this process is simple but not always suitable. Natural numbers have ordering property with one number greater than, less than, or equal to the other number. However, categorical variables do not obey ordering property, and learning algorithms may capture such no-existent relations between the categorical variables and can give less accurate predictions when input features are label encoded (Brownlee, 2021). One way to deal with the problem is to use a one-hot encoding. In this scenario, a feature like a gender having values male and female is transformed to two new features with values (1,0) or (0,1). The numerical variables are scaled by multiplying them with a fraction obtained by dividing the difference between the maximum and minimum value of the feature by the maximum value.

# Reflection on the exploratory data analysis and insights obtained from visualizations, findings, and discussion

Data analysis is answering questions by analyzing data. From More Life’s perspective, the following questions are essential for program designers to understand factors affecting participant attrition.

1. Is program month related to participant engagement behavior?
2. Is the participant group size related to the participant engagement?
3. Who is more likely to complete the program old or young age groups?
4. Is lower weight loss in the initial weeks associated with participant dropout?
5. **Does program month influence participant engagement behavior?**

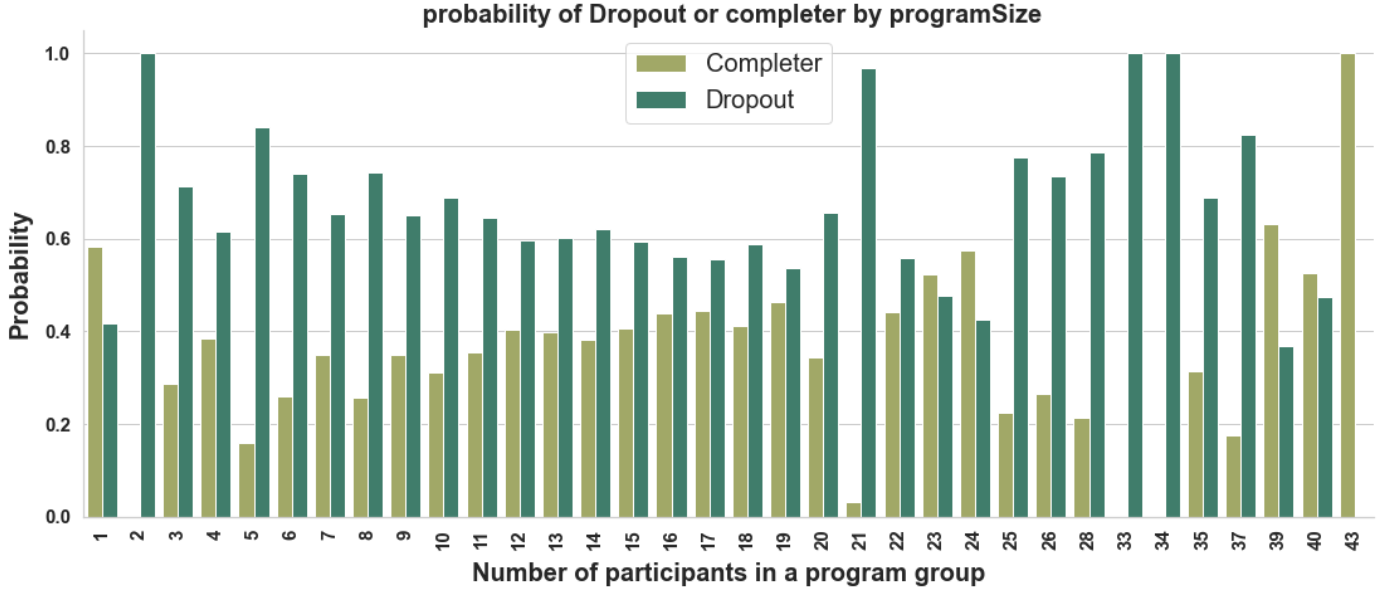
A distribution plot between program month and normalized values of the number of participants on the y axis is shown below.



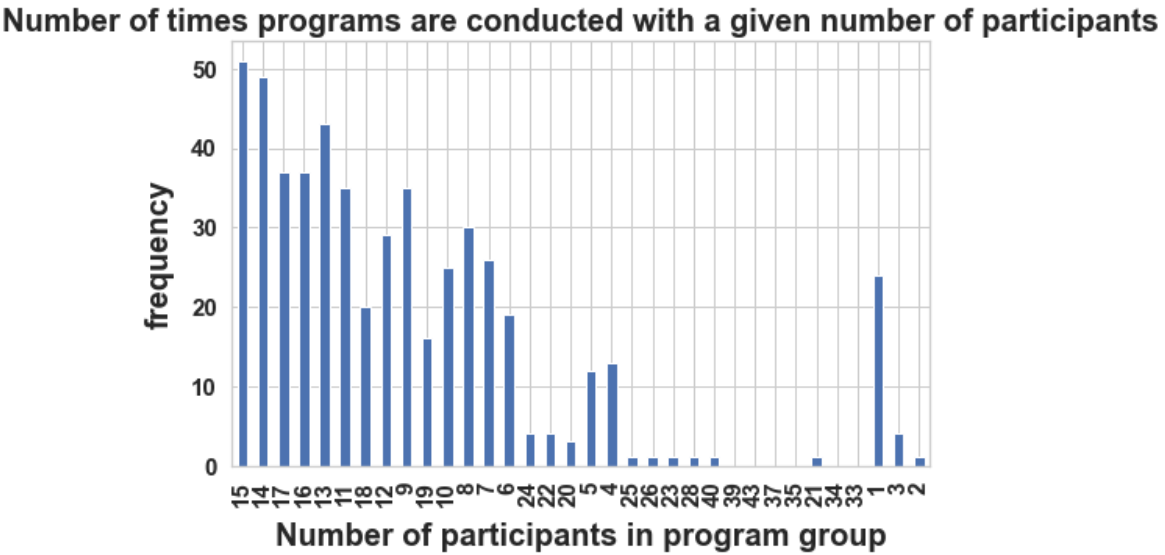
From the above plot, if we exclude the probabilities in March and may, we see that the participant's dropout probability keeps strictly decreasing with the month until August, where dropout probability becomes nearly 50% and then again strictly increasing with the month. This means a program that starts in May, June, July, and august has relatively lower dropout rates than the programs started in other months.

1. **Does the participant group size affect the participant engagement?**

Below is a normalized distribution bar plot of the probability of dropout or completion based on the number of participants for a program group.

****

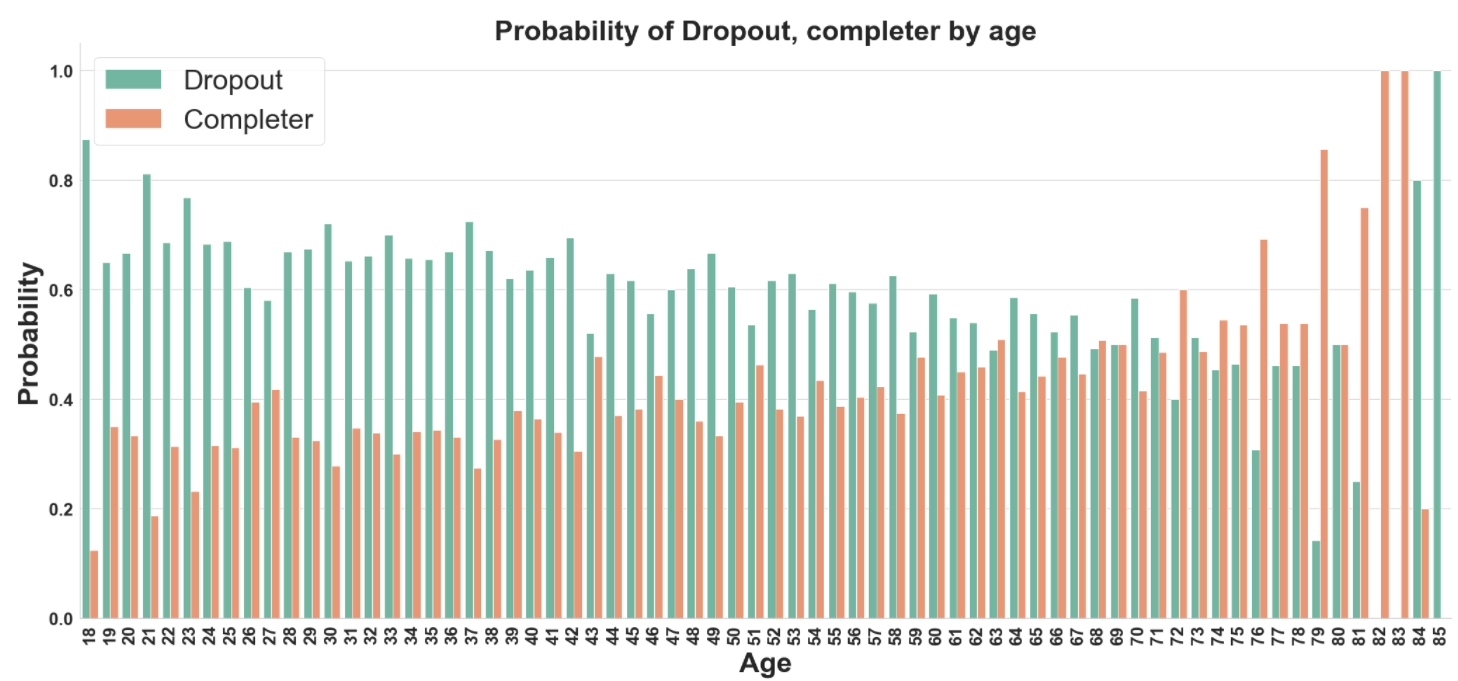
From the above bar plot, we see no consistent trend between the program size and attrition. However, a reasonable inference cannot be made as the frequency of some programs with a given number of participants is very low. For example, programs with participant sizes between 20 and 43 are scarce. A frequency plot shown below shows the frequency with which a program with a given program size occurred.



As most of the programs conducted belong to program sizes in between 4 and 19 participants per group, we can find from the above bar chart that participant attrition probability decreases with increases with the program size in this range.

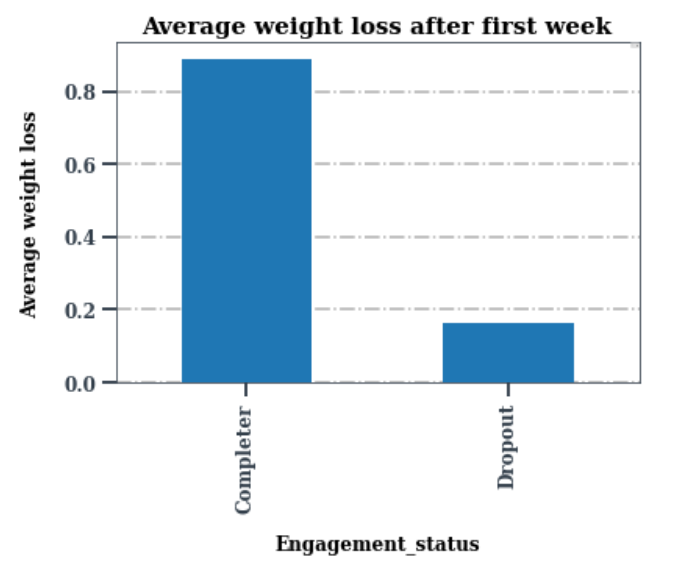
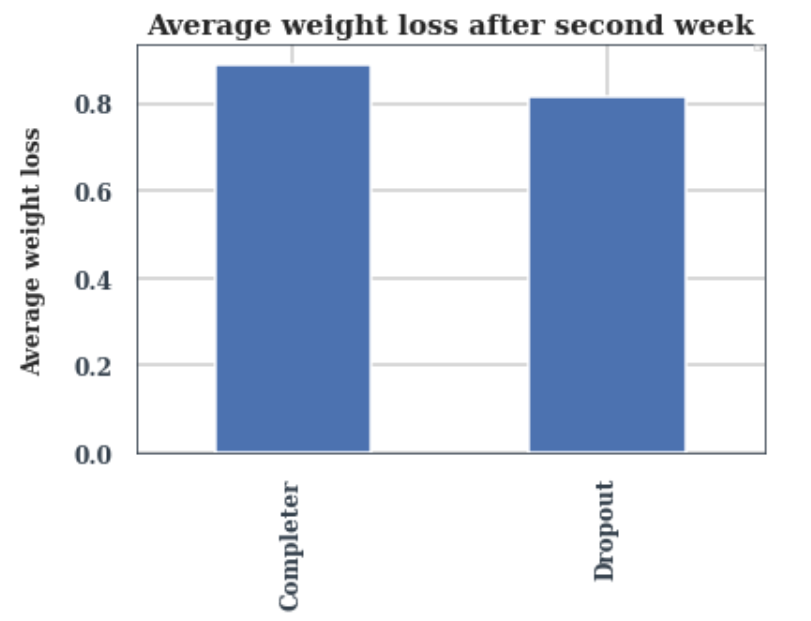
1. **Who is more likely to complete the program old or young age groups?**

The following probability graph shows that younger age groups are more likely to drop out than older age groups. The anomaly in the trend above the age of 80 should be ignored as a few participants represent that age group.



1. **Is lower weight loss in the initial weeks associated with participant dropout**?

The average weight loss of completers and dropouts is shown below. From the plots, we can see that dropouts tend to lose less weight after the first and second week compared to the completers

Exploratory data analysis showed the trends in data and answered some of the questions about whether a given feature influences attrition. However, for MoreLife program designers predicting the chances of a participant dropping out early is more important. For this, machine learning techniques are implemented.

While exploratory data analysis gives some insights into the trends in participant’s program engagement, it is not enough to predict because EDA is limited by visualization with a few features taken together. Therefore, a model should incorporate influence obtained from many input features taken together to predict participant engagement. This is where classification algorithms come in.

## Setting a success criterion for classifier prediction and baseline accuracy

The final sample after data preparation has 4501 rows, with 2331 instances corresponding to participant dropout and 2170 instances corresponding to successful program completion. This means 52 percent are dropouts, and 48 percent are completers. This implies that a blind prediction that all participants discontinue the program predicts dropouts correctly with 52 percent success and the rest of the other 48 percent completers are miss classified as dropouts. Thus, a blind prediction that everyone drops out makes the model 52 percent accurate. For a machine-learning algorithm to be helpful, it should show at least 52 percent accuracy when tested on unseen data.

Base line accuracy = 52%

# Model building through supervised machine learning:

Supervised learning models like logistic regression, random forest, naïve Bayes, and artificial neural network models are trained on the data to predict the chances of a participant dropout. The dataset is initially split into train and test samples in the ratio of 7:2. Then, cross-validation is used on the training dataset to reduce model overfitting. Finally, the sklearn library is used (Pedregosa, F. et al., 2011) for implementing the model

### Logistic Regression:

Logistic regression is a well-suited model when the outcome is a binary categorical variable like dropout or completer, as in this project. Logistic regression links the probability with which a sample belongs to a class with multiple explanatory variables (Hoffman, I.E., 2019). The model can also be extended to multinomial dependent class variables. Logistic regression assumes that the logarithm of the odds ratio can be set equal to a linear combination of input features and a constant term. The odds ratio is the ratio of the probability that a sample is in one of the binary classes and is not in that binary class. The equation used in logistic regression is

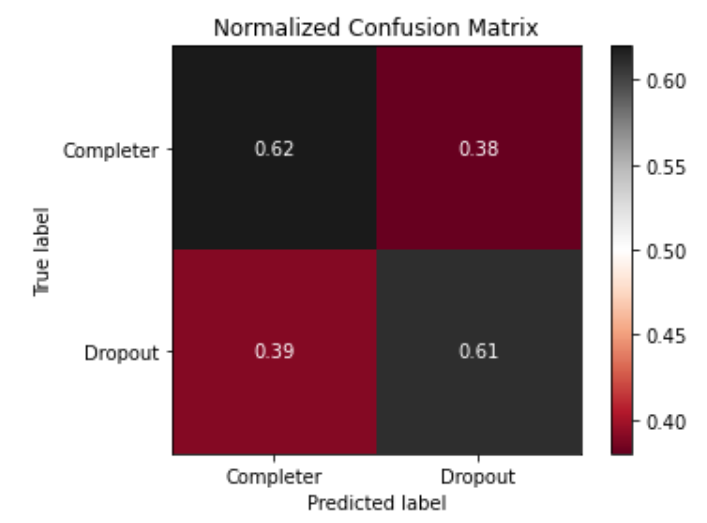
Log(P/1-P) = C0 + C1X1 + C2X2 + …. CnXn

Where Xi is, an input feature and the constants are coefficients. And P is the probability of Completer or Dropout in this case. The values of the coefficients are obtained by ­training the model with a set of input features whose outcome is already known. The model is trained to estimate the constants to minimize the classification error. This is achieved by minimizing a loss function of coefficients. The exponential of the coefficients can be interpreted as the factor by which odds (P/1-P) increase for a unit change in the corresponding input feature. This is known as the feature importance score.

The dataset is initially split in the ratio of 8:2 for training and testing. The logistic model hyperparameters are obtained after evaluating the different combinations for accuracy. The final model is implemented with l2 regularization penalty term, C parameter set to 10000, and ‘newton-cg’ solver. The model gave an accuracy of 64% on train data and 62% on test data which is slightly above the baseline accuracy of 52%. The obtained AUC (Area under the receiver operator characteristics curve) value is 0.68.

### Logistic regression model evaluation:

Confusion matrix:



**Logistic regression Testing performance**

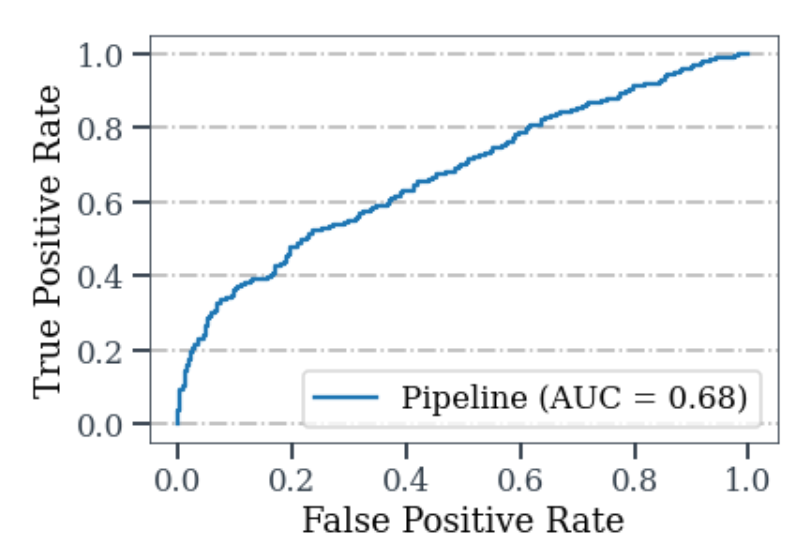
Performance metrics:

Model accuracy = 0.62

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | Recall | f1-score |
| Completer | 0.59 | 0.62 | 0.61 |
| Dropout | 0.64 | 0.61 | 0.63 |

#### Receiver operator characteristics:

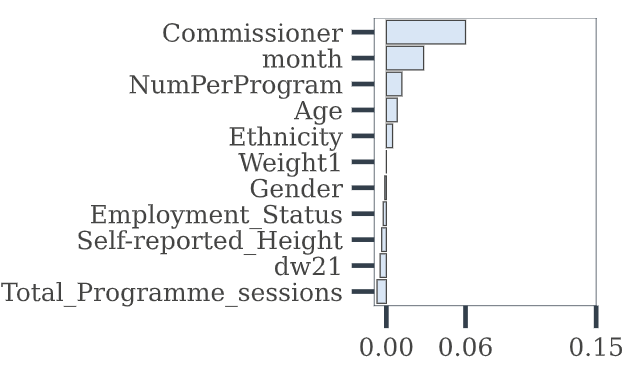
**ROC-CURVE and AUC value, Logistic regression**



#### Feature importance

For understanding the contribution of each feature to the logistic model, permutation feature importance is evaluated. Commissioner, age, total program sessions, and program month are strong predictors of participant engagement.

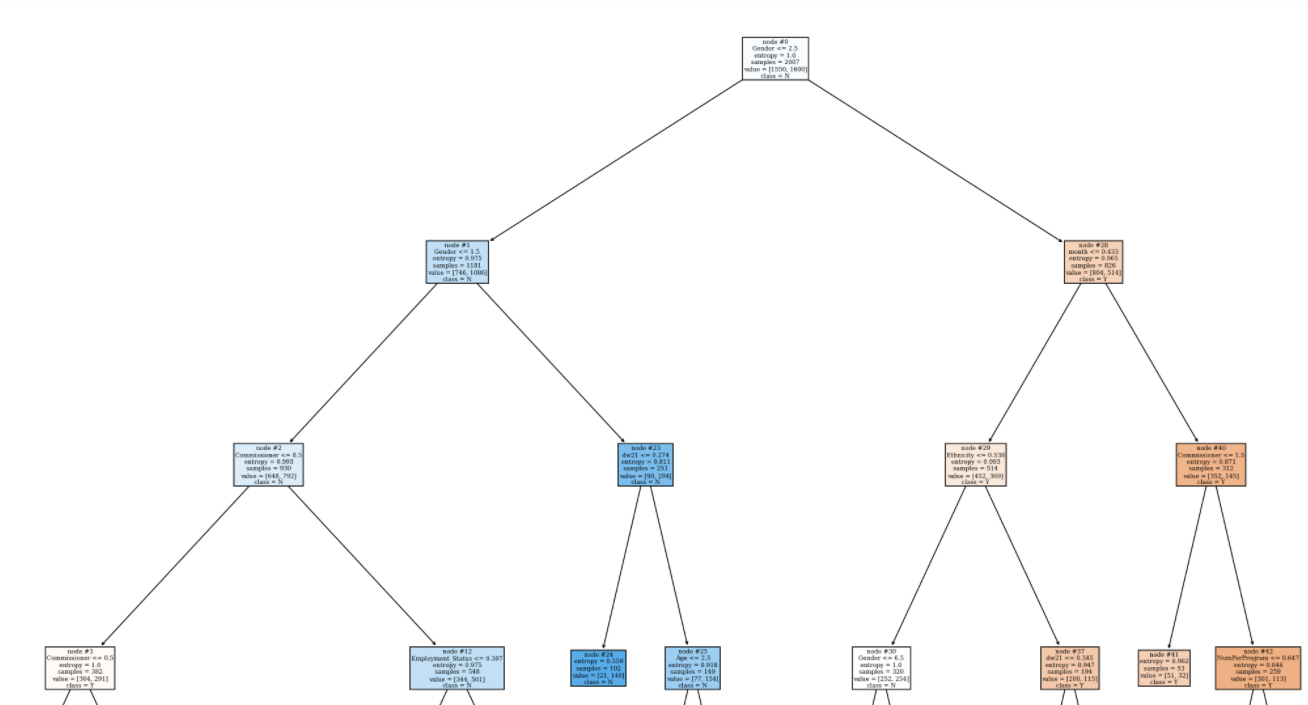
**Permutation feature importance score**

****

### Random Forest Classifier

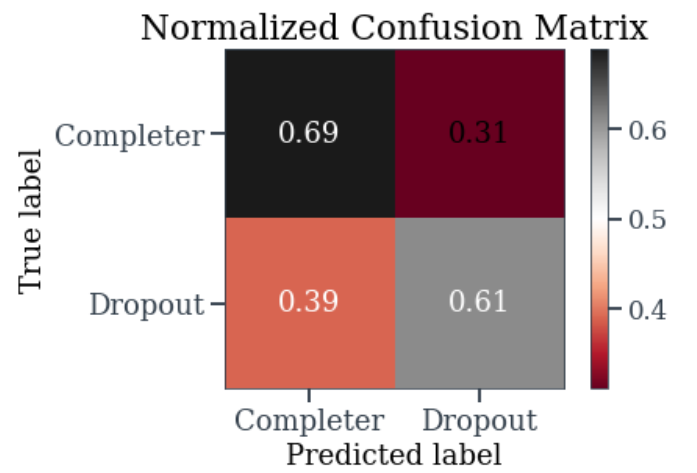
Random forest is a supervised machine learning technique that performs classification based on the decision aggregated from an ensemble of trees. The collective decision obtained from the ensemble minimizes learning through overfitting on the training data that generally happens with a decision tree. In random forest models, a random subset of input features and a random subset of the sample are used for training the individual decision trees. The sample subsets are extracted by replacing back the selected records from the original population. This is called bootstrapping. When each subset is trained with the selected features, individual decision trees that are very different from each other are obtained. When a new testing record is given as an input to the random forest model, each decision tree classifies the testing record, and the aggregation of all the results gives the final classification output. This whole process is called bagging. Bootstrapping makes the data less sensitive to the training data set, and random feature selection makes the trees split in different ways

The random forest model is trained with the training data. First, the models are trained by encoding categorical variables using ordinal and one-hot encoding. Both gave similar accuracies, but the ordinal encoding process time is quicker. After evaluating the random forests' accuracy with different parameter combinations and 10-fold stratified cross-validation, the best model parameters are obtained. The final model has 180 trees, with a maximum tree depth of 100 nodes, 130 samples per node, minimum of 40 samples per leaf node. Entropy is used for splitting a node, and 10-fold cross-validation is also included to avoid overfitting. These parameters are obtained by evaluating model performances on different possible parameters. The final model is tested on the test dataset giving an accuracy of 0.64. AUC value is 0.7.



A decision tree in the forest viewed to a depth of 4 levels

Confusion Matrix:

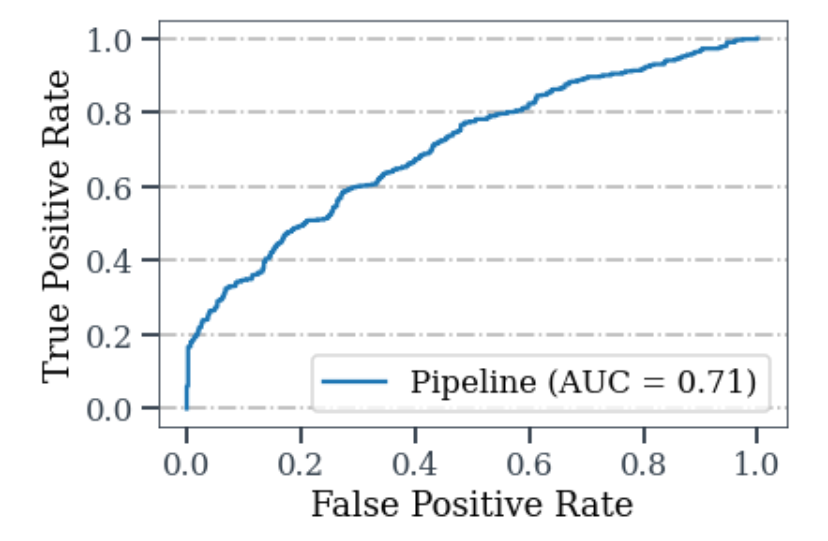


Performance metrics:

Model accuracy 0.65

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| Completer | 0.61 | 0.69 | 0.64 |
| Dropout | 0.68 | 0.60 | 0.64 |

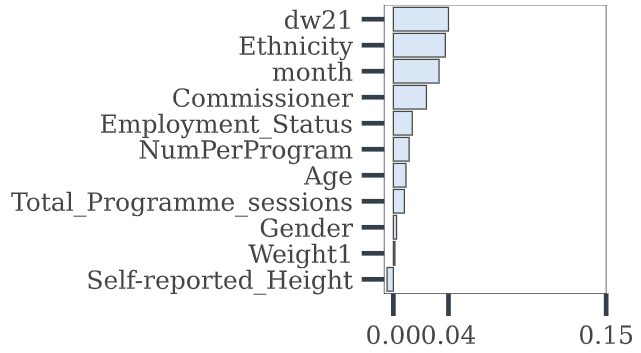
### ROC-Curve and AUC value



### Feature importance

Permutation feature importance of the random forest model shows that the weight change after the first week, ethnicity, program month, and commissionaire are strong predictors for classifying participants into completers and dropouts. While gender, initial weight, number of participants per program are week predictors. As Employment status has lots of missing values replaced with ‘Not known’ this feature may be included as a strong indicator for classification.

**Permutation feature importance Random Forest**

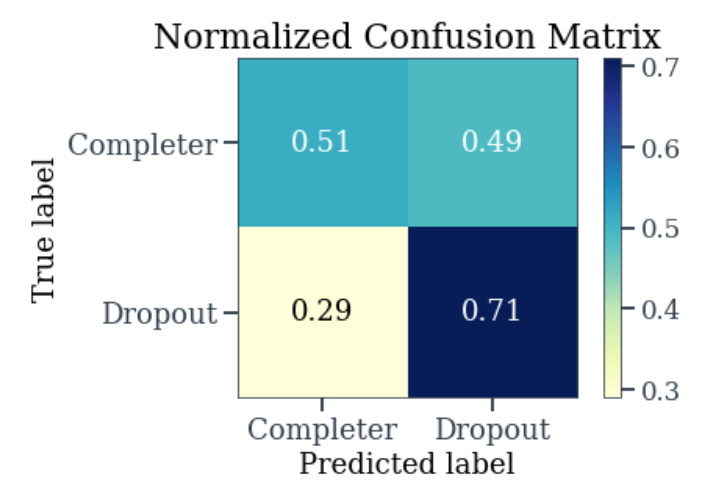


Commissionaire, program month, and weight change after the first week are better predictors of participant engagement.

Naïve Bayes:

Naïve Bayes is a probabilistic approach based on the conditional probability of Bayes theorem. In this model, it is assumed that the input features are independent of each other. The dataset has both continuous and categorical variables. The categorical variables are label encoded, and the gaussian naive bays model is trained as the continuous variables like age, initial weight, weight change after the first week, number of participants per program, height resemble a normal distribution. The model accuracy is 61 percent and has an AUC value of 0.64.

### Confusion matrix

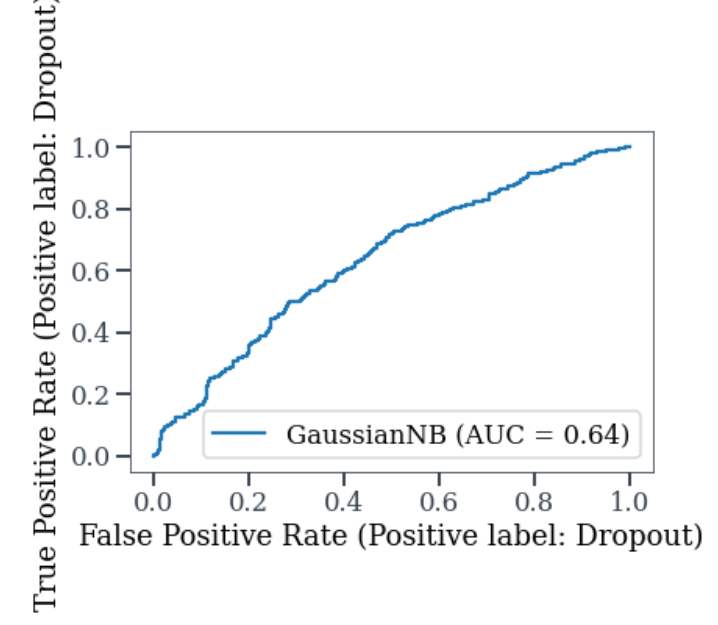


Performance metrics:

Accuracy 0.61

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| Completer | 0.61 | 0.51 | 0.56 |
| Dropout | 0.61 | 0.71 | 0.66 |

### ROC-Curve and AUC value



Feature importance:

Permutation feature importance shows that employment status, commissioner, ethnicity, and age are important factors contributing to classification.

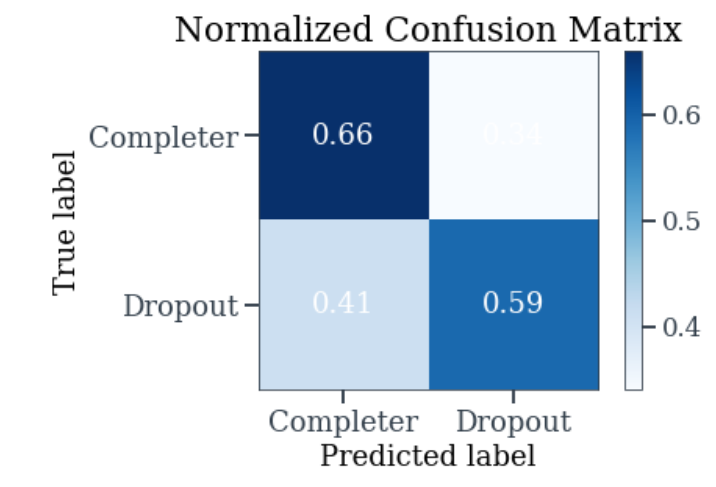


### MLP Classifier

The multilayer perceptron is a deep neural network classifier that approximates a function when learning through training data. The model consists of layers of artificial neurons with weights and biases that are obtained after training the network. An artificial neuron is characterized by an output obtained after a linear combination of the inputs is passed through an activation function. Learning involves adjusting the weights of individual neurons to minimize the error in the output.

The best hyper parameters are obtained after a grid search. They are alpha = 0.0001, hidden layers = 5, learning rate = constant, max iterations = 10000, solver = ‘adam’. The model accuracy is 62%

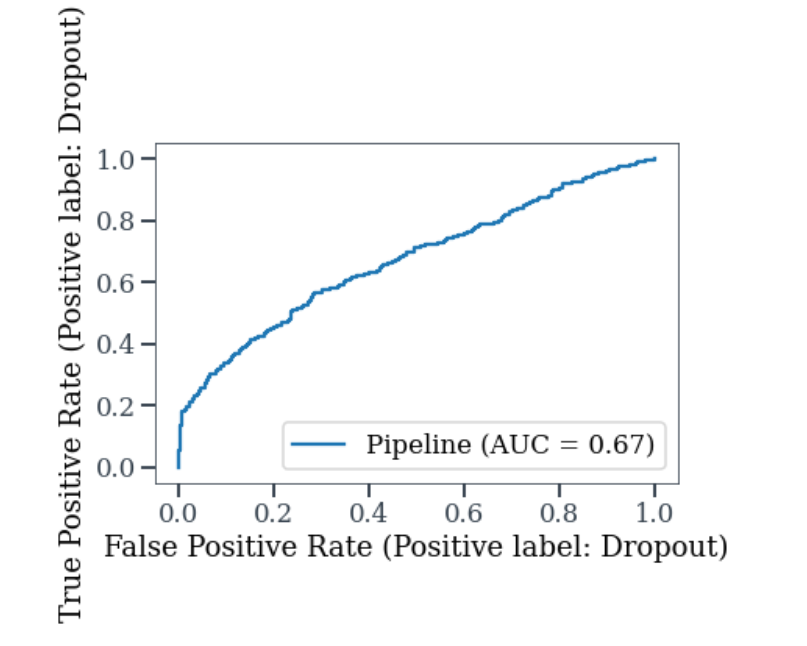
#### Confusion matrix



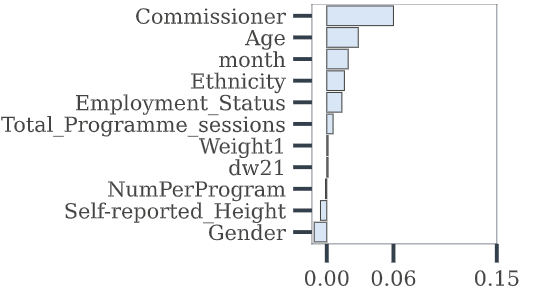
#### Performance metrics

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| Completer | 0.59 | 0.66 | 0.62 |
| Dropout | 0.65 | 0.59 | 0.62 |

#### ROC Curve



#### Permutation importance



### Comparison of Different models

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** |  | **Precision** | **Recall** | **F1- score** | **Accuracy** | **AUC** |
| Logistic Regression |  |  |  |  | 0.62 | 0.68 |
|  | Completer | 0.59 | 0.62 | 0.61 |  |  |
|  | Dropout | 0.64 | 0.61 | 0.63 |  |  |
| Random Forest |  |  |  |  | 0.64 | 0.70 |
|  | Completer | 0.61 | 0.69 | 0.64 |  |  |
|  | Dropout | 0.68 | 0.60 | 0.64 |  |  |
| Naïve Bayes |  |  |  |  | 0.61 | 0.64 |
|  | Completer | 0.61 | 0.51 | 0.56 |  |  |
|  | Dropout | 0.61 | 0.71 | 0.66 |  |  |
| MLP |  |  |  |  | 0.62 | 0.67 |
|  | Completer | 0.59 | 0.66 | 0.62 |  |  |
|  | Dropout | 0.65 | 0.59 | 0.62 |  |  |

# Survival analysis

Survival analysis is a collection of methods used for finding the time duration after which an event of interest occurs. An example of an event could be the death of a cancer patient after treatment, and the duration could be the number of days or months after which the event occurred. The period before which an event occurs is called survival, and the event is generally called hazard. Survival analysis helps find the duration after which participants could discontinue the program and the factors that influence the survival probability and duration.

## Redefining dropout:

The original dataset defines dropout with an attendance percentage less than a cutoff of 75%, but in survival analysis, a dropout is an event when a participant is last seen irrespective of his attendance percentage. The survival duration of such participants is the number of weeks between the program start date and the occurrence of the dropout event.

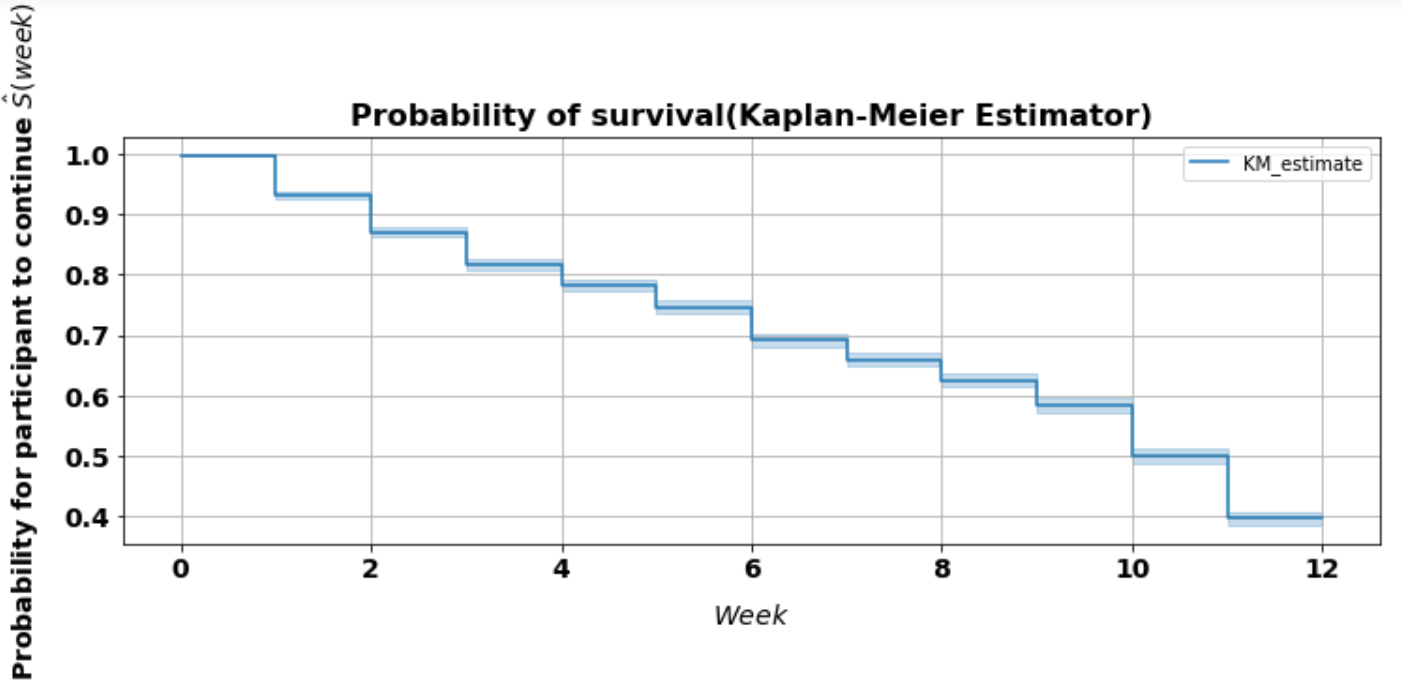
## Survival models

There are two quantities of interest the survival probability and the hazard function. Survival probability gives the probability of survival at a time t beyond a given program week. The hazard function is the probability that an event occurs at a time t in its neighborhood interval ‘𝛥t.’ The time integral of the hazard function is the cumulative hazard function. Kaplan–Meier estimator S(t) is a survival probability function defined as

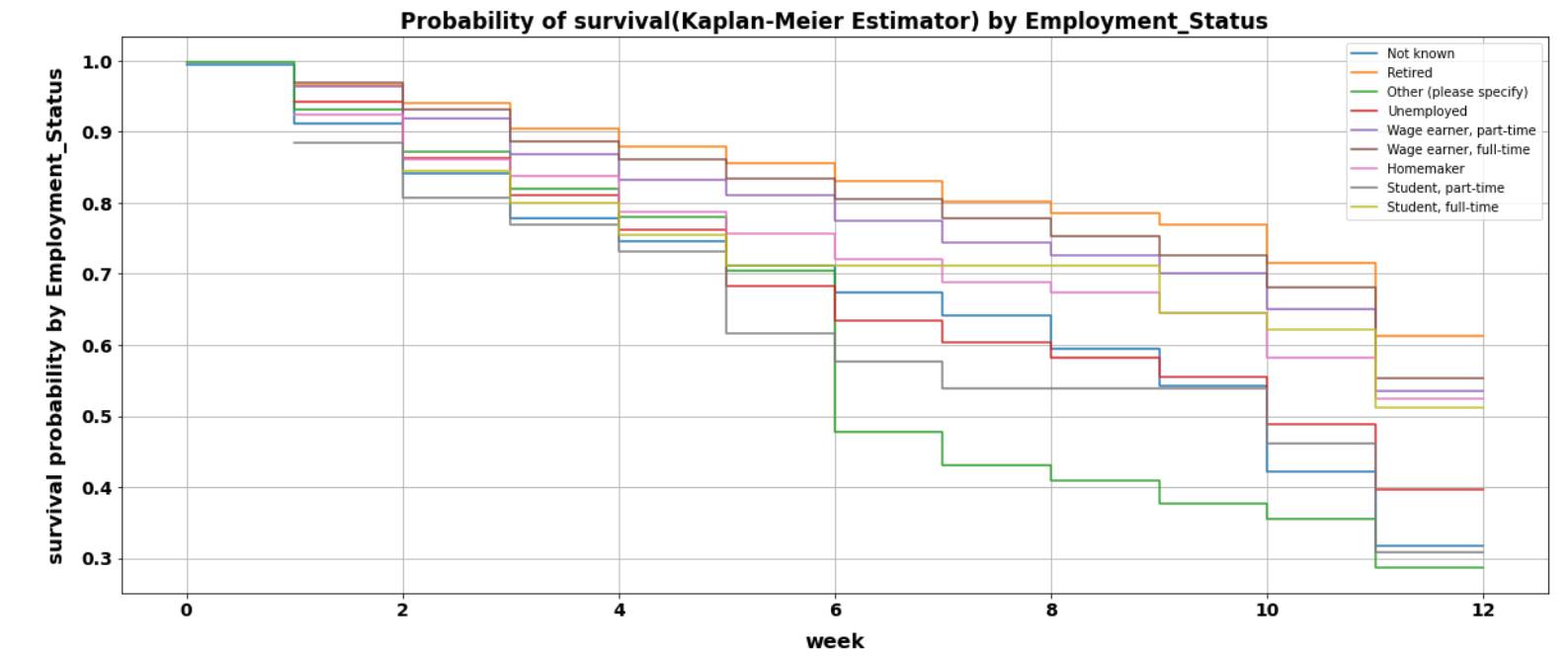
Where is the number of dropouts at week and is the number of survivors before the week

A new feature called survival time is added to the dataset. It indicates the week after which a participant has discontinued from the program or the week last attended. Some programs have more than 12-week sessions. Some have even more. For survival analysis, only 12 weeks are considered. The weekly attendance has three unique values 'Y', 'N,' 'nan.' In the dataset, the percentage of attendance is calculated by assuming that null values indicate absence.

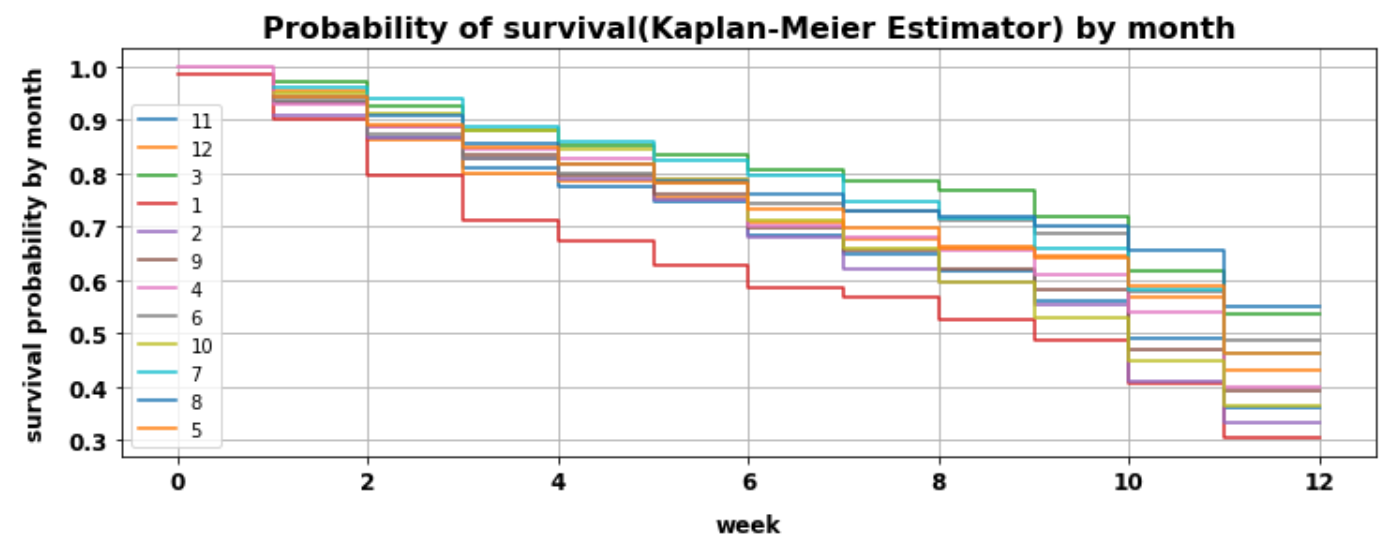
The probability of survival at the end of each program week is shown below.



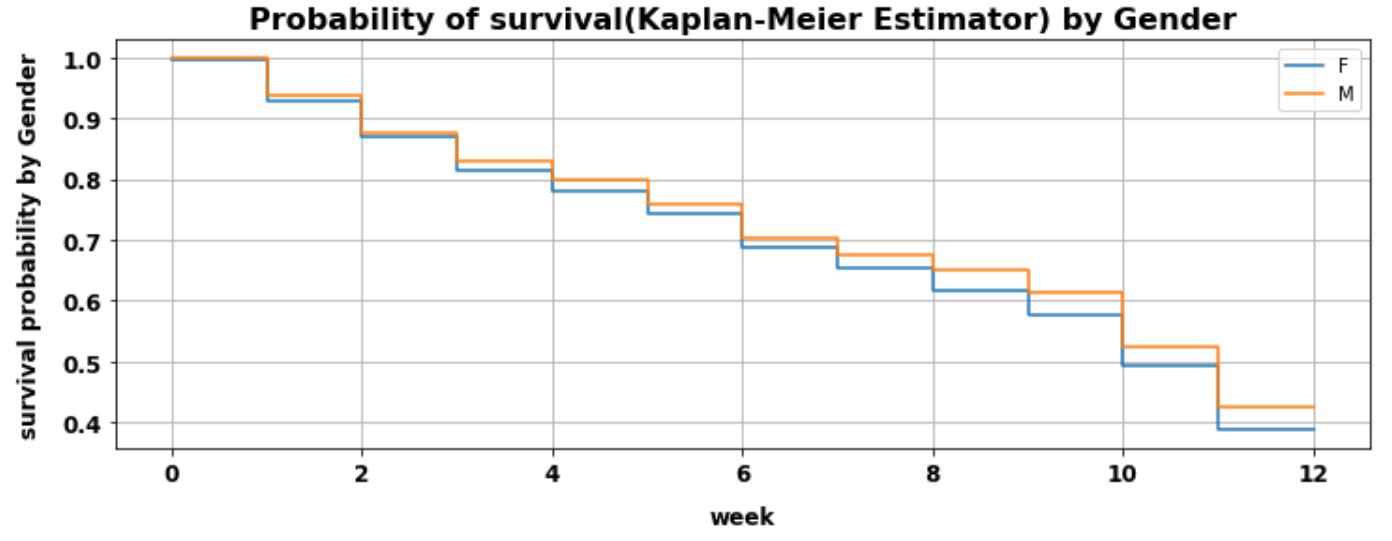
The participant survival chances drop rapidly in the beginning and ending weeks of the program. The drop in survival between weeks 3 and 9 is relatively low. Survival functions are also obtained for each category of the values in a feature.



The survival curves show that retired, full-time, and part-time wage earners, Home-makers, and full-time students have lower survival rates than unemployed, Part time students, other employment categories, and unknown employment status. The drop-in survival rates are relatively steady between week6 to week 10. Retired participants have a relatively higher survival probability at every program week. This also supports that age has an impact, as previously seen.



The probability of completing(surviving) every week is low in January, February and march compared to other months. The survival chances drop significantly in first 4 weeks in the month of January with nearly 30% of participants dropping with in first 3 weeks. Participants have higher survival rates in the month of November and march.



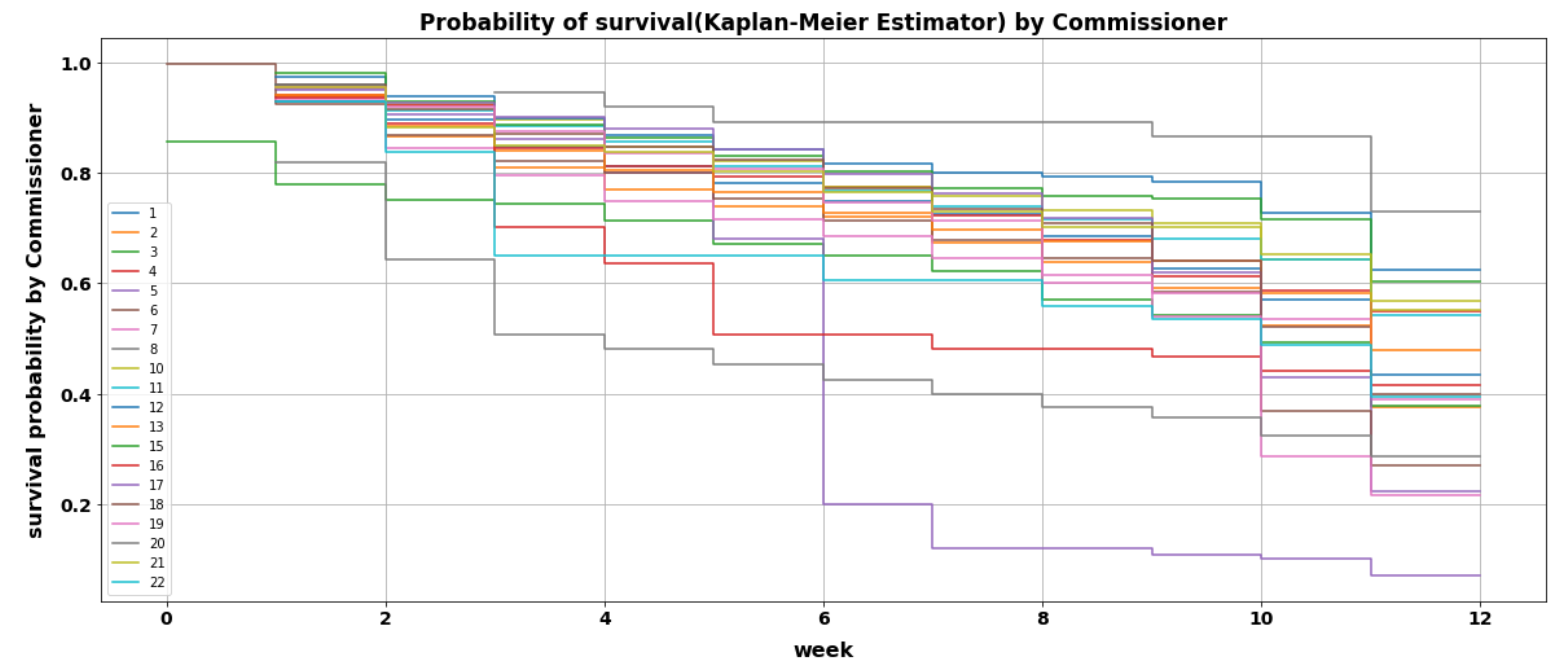
The female completion chances reduce faster than males during the ending program sessions

Is this difference significant?

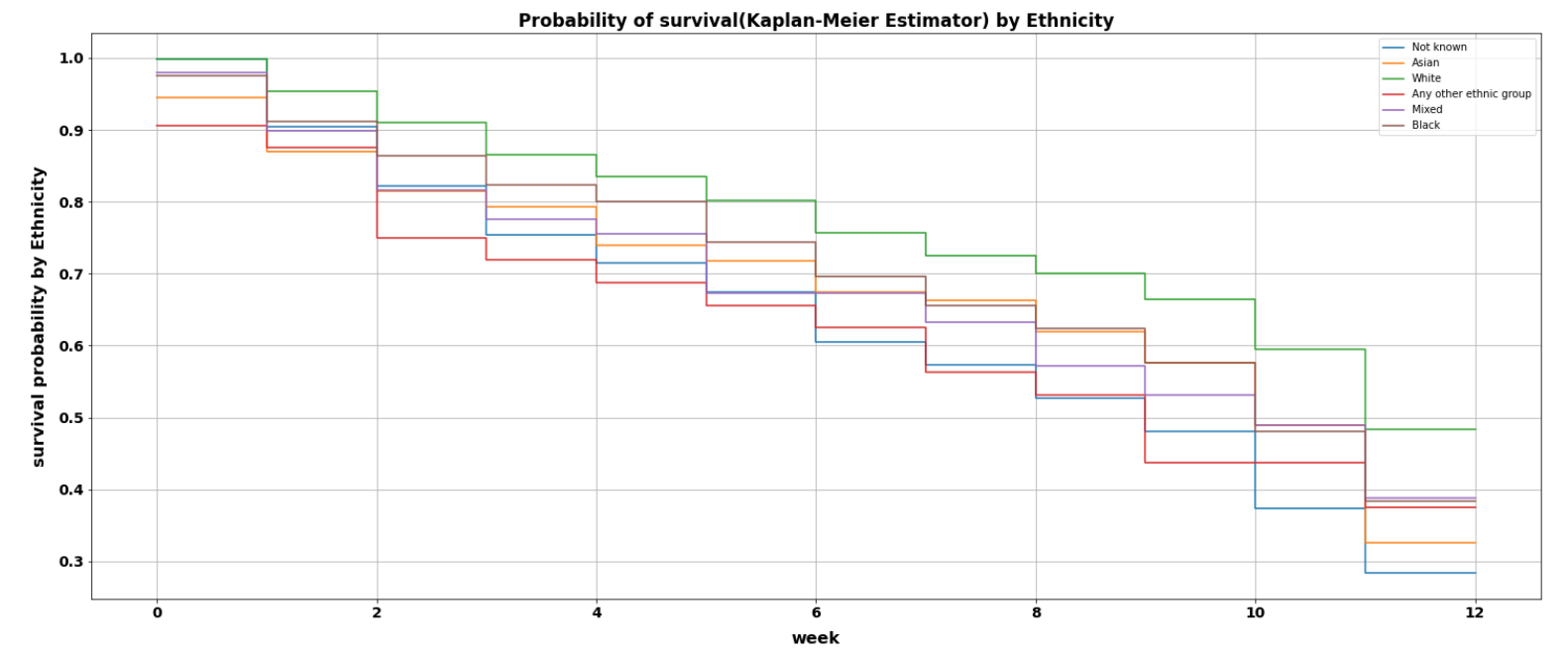
Null hypothesis there is no significant difference between the survival of both gender groups

Log-rank test for statistical significance gave a p-value = 0.01 and test statistic = 6.36

p < 0.05 rejects the null hypothesis. Thus, there is a significant difference in the program survival of male and female populations.



Commissionaires also show differences in survival rates at each week of the program sessions. Commissionaire 5 has very high drop in survival after 6th week of the program. For commissioner 20 survival rate drops rapidly in first 3 weeks.



The survival of Asian ethnic group drops highly after the 9th program week.

## COX model for finding the influence of individual features on attrition:

The Kaplan curves take only one variable group. For analyzing the hazard function with multiple variables, we use the Cox proportional hazards model. This model identifies how different factors like age, commissioner, employment status affect a participant's survival. COX model also can handle both categorical and numerical variables.

In the proportional hazard model, the hazard function is modeled in terms of a bassline hazard function which is independent of the variables like gender, ethnicity, etc. The baseline hazard function is then multiplied by an exponential of a linear sum of the covariates, and Covariates are the terms like gender, commissioner, age, etc. The hazard function is defined as

where

is called the hazard ratio and is the baseline hazard function.

> 1 indicates higher participants dropout chances

< 1 indicates lower participant dropout chances.

### Hazard ratios

The complete hazard ratios and their statistical significance are tabulated in the appendix section: -Hazard ratios.

Here are some of the significant results.

|  |  |  |
| --- | --- | --- |
|  | Hazard ratio | Statistical significance(p-value) |
| Ethnicity=White | 0.7 | 0.1 |
| Employment\_Status=Other (please specify) | 1.19 | 0.13 |
| Employment\_Status=Retired | 0.82 | 0.1 |
| Employment\_Status=Student, part-time | 1.47 | 0.14 |
| Employment\_Status=Unemployed | 1.38 | 0.01 |
| month=6 | 0.47 | 0.005 |
| month=8 | 0.53 | 0.005 |
| month=9 | 0.47 | 0.005 |
| GroupSize=2 | 5.79 | 0.02 |
| GroupSize = 8 | 2.3 | 0.01 |
| GroupSize = 14 | 2.23 | 0.01 |
| GroupSize = 21, 28, 33, 35 | > 3 and <4 | 0.005 |
| GroupSize = 37 | 7.14 | 0.005 |
| Commissionaire = 3 | 0.61 | 0.005 |

* Commissionaire = 3 reduces the dropout risk
* Commissionaire = 5, 7, 16, 17, 19, 20, 22 increase the dropout risk (p < 0.005)
* White ethnicity is associated with reduction in dropout (p < 0.1)
* Retired individuals have lower risk associated with dropout (p < 0.1)
* For the unemployed category, p<0.1, which means unemployment related to dropout. Being unemployed increases the hazard by a factor of 1.41
* The hazard ratio is high for programs with larger participant sizes. Conversely, the hazard ratios are relatively small for programs below size 21.
* Program months of June, August, September reduces the hazard risk by a factor of 0.47, 0.53, 0.57, respectively

# Conclusion

The application of machine learning and statistical techniques can help predict participants at risk of dropout in lifestyle management programs. Logistic regression, random forest, naïve Bayes, and multilayer perceptron models could predict participant engagement with accuracies of 62%, 64%, 61%, and 62%, respectively. The baseline prediction accuracy was 52%. The random forest method performed relatively better than the other three algorithms, with an AUC value of 0.7. Exploratory data analysis shows that lower age is associated with participant dropout. Survival analysis shows that a decrease in participant’s program survival is higher in the first and the last three weeks of the program. It also identifies how participant's program survival depends on their employment status, age, program month, and gender. Cox’s proportional hazards model identifies a statistically significant risk associated with unemployed participants, larger program group size, certain commissionaires, and program start months.

# Study limitations

1. The study is has considered very features for predictive modeling. However, as attrition has behavioral and socio-economic components, further studies should include more features by conducting a questionnaire to estimate participant’s behavioral and psychological factors.
2. The data has many missing values making it difficult for reliable and accurate studies. In addition, in this study, the sample size is preserved by data imputation rather than removing many rows that contained missing data. This has made it challenging to understand feature contributions in identifying the participants at risk of dropping. Feature importance scores were not consistent and reproducible for some features like ethnicity and employment status. The cox hazard model gives the hazard ratios which indicates the importance of a feature in its contribution for increasing ore decreasing the survival risk. But the results are not statistically significant due to missing values. Further studies should be based on data with higher quality.
3. This study did not explore any interaction terms in the input features to check if they could improve the model performances. This study also has not attempted to identify any noisy features that contribute to low model performance.
4. This research Could not link up the deprivation indices with output area. Further studies should aim to obtain relevant datasets that obtaining more features.

# Ethical approval:

The data obtained from MoreLife is pseudonymized to maintain the anonymity of the participants. The ethics approval for this project was given after a two-stage risk evaluation by the ethics board, Leeds Beckett University.

# References:

Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., and Khan, M.K., 2018. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, *42*(11), pp.1-13.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L. and Ridella, S., 2012. The ‘K’in K-fold cross validation. In *20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (pp. 441-446). i6doc. com publ.

*Adults*. [online] Ncbi.nlm.nih.gov. Available at: <https://www.ncbi.nlm.nih.gov/books/NBK2003/> [Accessed 28 June 2021].

Baker, C., 2021. *Obesity Statistics*. [online] House of Commons Library. Available at: <https://commonslibrary.parliament.uk/research-briefings/sn03336/> [Accessed 31 August 2021].

‌Bali, J., Garg, R. and Bali, R., 2019. Artificial intelligence (AI) in healthcare and biomedical research: Why a strong computational/AI bioethics framework is required?. *Indian Journal of Ophthalmology*, 67(1), p.3.

Babajide, O., Tawfik, H., Palczewska, A., Gorbenko, A., Astrup, A., Martinez, J., Oppert, J. and Sorensen, T., 2019. Application of Unsupervised Learning in Weight-Loss Categorisation for Weight Management Programs. *2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*

Burgess, E., Hassmén, P. and Pumpa, K.L., 2017. Determinants of adherence to lifestyle intervention in adults with obesity: a systematic review. *Clinical obesity*, *7*(3), pp.123-135.

BBC News. 2021. *Anti-obesity drive: Junk food TV adverts to be banned before 9pm*. [online] Available at: <https://www.bbc.co.uk/news/uk-politics-57593599> [Accessed 6 September 2021].

Brownlee, J., 2021. *Why One-Hot Encode Data in Machine Learning?*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/> [Accessed 22 August 2021].

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R., 1999, March. The CRISP-DM user guide. In *4th CRISP-DM SIG Workshop in Brussels in March* (Vol. 1999). sn.

Chicco, D. and Jurman, G., 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), pp.1-13.

Cockburn, A., 2000. Selecting a project's methodology. *IEEE software*, *17*(4), pp.64-71.

En.wikipedia.org. 2021. *SEMMA - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/SEMMA> [Accessed 2 September 2021].

En.wikipedia.org. 2021. *Machine learning - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Machine\_learning> [Accessed 14 August 2021].

Eysenbach, G., 2005. The law of attrition. *Journal of medical Internet research*, *7*(1), p.e402.

Finley, C.E., Barlow, C.E., Greenway, F.L., Rock, C.L., Rolls, B.J. and Blair, S.N., 2007. Retention rates and weight loss in a commercial weight loss program. *International journal of obesity*, *31*(2), pp.292-298.

Germann, J.N., Kirschenbaum, D.S. and Rich, B.H., 2006. Use of an orientation session may help decrease attrition in a pediatric weight management program for low-income minority adolescents. *Journal of Clinical Psychology in Medical Settings*, *13*(2), pp.169-179.

GOV.UK. 2021. *Health matters: obesity and the food environment*. [online] Available at: <https://www.gov.uk/government/publications/health-matters-obesity-and-the-food-environment/health-matters-obesity-and-the-food-environment--2> [Accessed 12 August 2021].

Hand, D.J., 2007. Principles of data mining. *Drug safety*, *30*(7), pp.621-622.

Hodneland, E., Dybvik, J.A., Wagner-Larsen, K.S., Šoltészová, V., Munthe-Kaas, A.Z., Fasmer, K.E., Krakstad, C., Lundervold, A., Lundervold, A.S., Salvesen, Ø. and Erickson, B.J., 2021. Automated segmentation of endometrial cancer on MR images using deep learning. *Scientific reports*, *11*(1), pp.1-8.

Hoffman, J.I.E., Baic Biostatistics for Medical and Biomedical Practitioners.

Johnson, M., 2021. *1.1 A Simple Example | Feature Engineering and Selection: A Practical Approach for Predictive Models*. [online] Bookdown.org. Available at: <https://bookdown.org/max/FES/a-simple-example.html> [Accessed 4 September 2021].

Kim, E., Park, Y.B., Choi, K., Lim, Y.W., Ok, J.M., Noh, E.Y., Song, T.M., Kang, J., Lee, H. and Kim, S.Y., 2020. Application of Machine Learning to Predict Weight Loss in Overweight, and Obese Patients on Korean Medicine Weight Management Program. *The Journal of Korean Medicine*, *41*(2), pp.58-79.

Kukreja, H., Bharath, N., Siddesh, C.S. and Kuldeep, S., 2016. An introduction to artificial neural network. *Int J Adv Res Innov Ideas Educ*, *1*, pp.27-30.

Kuhn, M. and Johnson, K., 2013. *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.

Kim, M., Yang, J., Ahn, W.Y. and Choi, H.J., 2021. Machine Learning Analysis to Identify Digital Behavioral Phenotypes for Engagement and Health Outcome Efficacy of an mHealth Intervention for Obesity: Randomized Controlled Trial. *Journal of medical Internet research*, *23*(6), p.e27218.

Medium. 2021.Spring board. *Data Science vs Data Analytics — How to decide which one is right for you?*. [online] Available at: <https://medium.com/@springboard\_ind/data-science-vs-data-analytics-how-to-decide-which-one-is-right-for-you-41e7bdec080e> [Accessed 14 August 2021].

Marbán, Ó., Mariscal, G. and Segovia, J., 2009. *A data mining & knowledge discovery process model*. IntechOpen.

Moore, A.W., 2001. Cross-validation for detecting and preventing overfitting. *School of Computer Science Carneigie Mellon University*.

Nadali, A., Kakhky, E.N. and Nosratabadi, H.E., 2011, April. Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system. In *2011 3rd International Conference on Electronics Computer Technology* (Vol. 6, pp. 161-165). IEEE.

NHLBI Obesity Education Initiative Expert Panel on the Identification, a., 2021. *Clinical Guidelines on the Identification, Evaluation, and Treatment of Overweight and Obesity*

nhs.uk. 2021. *What is the body mass index (BMI)?*. [online] Available at: <https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/> [Accessed 12 August 2021]. Available at: https://www.nhs.uk/common-health-questions/lifestyle/what-is-the-body-mass-index-bmi/ (Accessed: 1 August 2021)

Nielsen, R.L., Helenius, M., Garcia, S.L., Roager, H.M., Aytan-Aktug, D., Hansen, L.B.S., Lind, M.V., Vogt, J.K., Dalgaard, M.D., Bahl, M.I. and Jensen, C.B., 2020. Data integration for prediction of weight loss in randomized controlled dietary trials. *Scientific reports*, *10*(1), pp.1-15.

Nobles, J., Griffiths, C., Pringle, A. and Gately, P., 2016. Design programmes to maximise participant engagement: a predictive study of programme and participant characteristics associated with engagement in paediatric weight management. *International Journal of Behavioral Nutrition and Physical Activity*, *13*(1), pp.1-10.

Oecd-ilibrary.org. 2021. *OECD iLibrary | Overweight or obese population*. [online] Available at: <https://www.oecd-ilibrary.org/social-issues-migration-health/overweight-or-obese-population/indicator/english\_86583552-en> [Accessed 12 August 2021].

Pedersen, D.H., Mansourvar, M., Sortsø, C. and Schmidt, T., 2019. Predicting dropouts from an electronic health platform for lifestyle interventions: analysis of methods and predictors. *Journal of medical Internet research*, *21*(9), p.e13617.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, pp.2825-2830.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), p.386.

Rowland, M., 1990. Self-reported weight and height. *The American Journal of Clinical Nutrition*, 52(6), pp.1125-1133.

Ruparelia, N.B., 2010. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), pp.8-13.

Skelton, J.A. and Beech, B.M., 2011. Attrition in paediatric weight management: a review of the literature and new directions. *Obesity reviews*, *12*(5), pp.e273-e281.

Soru, T. and Ngomo, A.C.N., 2014, September. A comparison of supervised learning classifiers for link discovery. In *Proceedings of the 10th international conference on semantic systems* (pp. 41-44).

Saltz, J. and Crowston, K., 2017. Comparing data science project management methodologies via a controlled experiment.

Scikit-learn.org. 2021. *4.2. Permutation feature importance — scikit-learn 0.24.2 documentation*. [online] Available at: <https://scikit-learn.org/stable/modules/permutation\_importance.html> [Accessed 4 September 2021].

Samuel, A.L., 1959. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*(3), pp.210-229.

Sharma, S. and Osei-Bryson, K.M., 2009. Framework for formal implementation of the business understanding phase of data mining projects. *Expert Systems with Applications*, *36*(2), pp.4114-4124.

Shafique, U. and Qaiser, H., 2014. A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, *12*(1), pp.217-222.

Sisense. 2021. *Data Science vs. Data Analytics - Defenitions & Differences Explained*. [online] Available at: <https://www.sisense.com/blog/data-science-vs-data-analytics/> [Accessed 14 August 2021].

Skelton, J.A., Goff Jr, D.C., Ip, E. and Beech, B.M., 2011. Attrition in a multidisciplinary pediatric weight management clinic. *Childhood Obesity*, *7*(3), pp.185-193.

Thomas, D.M., Martin, C.K., Heymsfield, S., Redman, L.M., Schoeller, D.A. and Levine, J.A. (2011). A simple model predicting individual weight change in humans. *Journal of Biological Dynamics*, [online] 5(6), pp.579–599. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3975626/>.

Tollefson, D.R., Gross, E. and Lundahl, B., 2008. Factors that predict attrition from a state-sponsored rural batterer treatment program. *Journal of Aggression, Maltreatment & Trauma*, *17*(4), pp.453-477.

Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M. and Zhao, S., 2019. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, *18*(6), pp.463-477.

World Health Organization (2021). *Obesity*. [online] www.who.int. Available at: <https://www.who.int/health-topics/obesity#tab=tab_1>.

Wolpert, D.H. and Macready, W.G., 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), pp.67-82.

# Appendices:

## Dataset dictionary

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Data type** | **Examples** |
| Client Id | string | C58504, C58937 . . . |
| Programme Name | string | P1, P2, p3, . . . P642 |
| Commissioner | integer | 1, 2, 3, . . . . 22 |
| week1 | string | ‘Y’, ‘N’, nan or null, Unique values |
| week2 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week3 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week4 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week5 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week6 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week7 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week8 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week9 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week10 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week11 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week12 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week13 | String | ‘Y’, ‘N’, nan or null, Unique values |
| week14 | String | ‘Y’, ‘N’, nan or null, Unique values |
| Sessions  Attended | Integer | 0,1,2,3, . . . . 11, 12 Total sessions attended |
| Total  Sessions | Integer | 12, 13, 14 Unique values |
| %Attendance | Float | 0% to 100% |
| Engagement  Status | string | ‘Dropout’, ’NonInitiator’,  ‘Completer’ |
| Status | string | ‘Accepted’ |
| Referred Date | datetime | Values from 2011-05-19 To 2018-02-21 |
| Programme Start Date | datetime | Values from 2011-09-14 To 2018-01-23 |
| Attended Sessions | Integer | 0,1,2,3, . . . . 11, 12 |
| Total Programme sessions | Integer | 12, 13, 14 |
| Gender | String | ‘F’, ‘M’ |
| Age | Integer | 25 – 87 |
| Ethnicity | String | nan, 'Asian', 'White', 'Any other ethnic group', 'Mixed', 'Black' |
| Employment  Status | String | Null, “Retired”, “Other (please specify)”, “Unemployed”, “Wage earner, part-time”, “Wage earner, full-time”, “Homemaker”, “Student, part-time”, “Student, full-time” |
| Output Area | String | '13UBGL0002', '13UBGG0019',... |
| Self-reported Weight | Float | Weight in lb |
| Self-reported Height | Float | Height in centimetre |
| Self-reported BMI | Float | Values between 0.33 and 488(need to remove outliers) |
| Weight1 | Float | Weight in lb |
| Weight2 | Float | Weight in lb |
| Weight3 | Float | Weight in lb |
| Weight4 | Float | Weight in lb |
| Weight5 | Float | Weight in lb |
| Weight6 | Float | Weight in lb |
| Weight7 | Float | Weight in lb |
| Weight8 | Float | Weight in lb |
| Weight9 | Float | Weight in lb |
| Weight10 | Float | Weight in lb |
| Weight11 | Float | Weight in lb |
| Weight12 | Float | Weight in lb |
| Weight13 | Float | Weight in lb |
| Weight14 | Float | Weight in lb |
| PreWeight | Float | Weight in lb |
| PostWeight | Float | Weight in lb |

## Hazard ratios COX model

is the hazard ratios

P is the statistical significance of a feature in influencing participant dropout.

| **coef** | **coef)** | **se(coef)** |  | **coef lower 95%** | **coef upper 95%** | **exp(coef) lower 95%** | **exp(coef) upper 95%** | **z** | **p** | **log2(p)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Commissioner=2** | 0.23 | 1.25 |  | 0.08 | 0.06 | 0.39 | 1.06 | 1.48 | 2.70 | 0.01 | 7.18 |
| **Commissioner=3** | -0.50 | 0.60 |  | 0.16 | -0.81 | -0.20 | 0.45 | 0.82 | -3.25 | <0.005 | 9.76 |
| **Commissioner=4** | -0.09 | 0.91 |  | 0.16 | -0.40 | 0.22 | 0.67 | 1.24 | -0.58 | 0.56 | 0.84 |
| **Commissioner=5** | 1.19 | 3.29 |  | 0.15 | 0.90 | 1.48 | 2.47 | 4.39 | 8.12 | <0.005 | 50.88 |
| **Commissioner=6** | 0.21 | 1.24 |  | 0.12 | -0.02 | 0.45 | 0.98 | 1.57 | 1.76 | 0.08 | 3.69 |
| **Commissioner=7** | 0.31 | 1.37 |  | 0.12 | 0.09 | 0.54 | 1.09 | 1.72 | 2.71 | 0.01 | 7.22 |
| **Commissioner=8** | -0.62 | 0.54 |  | 0.33 | -1.27 | 0.04 | 0.28 | 1.04 | -1.85 | 0.06 | 3.95 |
| **Commissioner=10** | -0.15 | 0.86 |  | 0.12 | -0.39 | 0.09 | 0.68 | 1.09 | -1.25 | 0.21 | 2.24 |
| **Commissioner=11** | -0.19 | 0.83 |  | 0.11 | -0.41 | 0.03 | 0.66 | 1.03 | -1.68 | 0.09 | 3.44 |
| **Commissioner=12** | -0.31 | 0.73 |  | 0.13 | -0.57 | -0.06 | 0.56 | 0.94 | -2.40 | 0.02 | 5.91 |
| **Commissioner=13** | 0.13 | 1.14 |  | 0.13 | -0.12 | 0.38 | 0.89 | 1.47 | 1.02 | 0.31 | 1.69 |
| **Commissioner=15** | -0.23 | 0.79 |  | 0.15 | -0.53 | 0.06 | 0.59 | 1.06 | -1.57 | 0.12 | 3.11 |
| **Commissioner=16** | 0.62 | 1.86 |  | 0.18 | 0.26 | 0.98 | 1.30 | 2.66 | 3.39 | <0.005 | 10.47 |
| **Commissioner=17** | 0.30 | 1.35 |  | 0.09 | 0.12 | 0.48 | 1.13 | 1.62 | 3.31 | <0.005 | 10.08 |
| **Commissioner=18** | 0.22 | 1.25 |  | 0.11 | 0.01 | 0.43 | 1.01 | 1.54 | 2.03 | 0.04 | 4.57 |
| **Commissioner=19** | 0.28 | 1.33 |  | 0.09 | 0.10 | 0.47 | 1.10 | 1.60 | 2.99 | <0.005 | 8.48 |
| **Commissioner=20** | 0.90 | 2.45 |  | 0.08 | 0.74 | 1.06 | 2.09 | 2.89 | 10.84 | <0.005 | 88.49 |
| **Commissioner=21** | -0.09 | 0.92 |  | 0.15 | -0.38 | 0.21 | 0.68 | 1.23 | -0.57 | 0.57 | 0.82 |
| **Commissioner=22** | 0.63 | 1.89 |  | 0.22 | 0.19 | 1.07 | 1.21 | 2.93 | 2.83 | <0.005 | 7.74 |
| **Gender=M** | -0.05 | 0.95 |  | 0.04 | -0.13 | 0.04 | 0.88 | 1.04 | -1.13 | 0.26 | 1.95 |
| **Age** | -0.01 | 0.99 |  | 0.00 | -0.01 | -0.01 | 0.99 | 0.99 | -6.13 | <0.005 | 30.09 |
| **Ethnicity=Asian** | -0.09 | 0.91 |  | 0.26 | -0.60 | 0.41 | 0.55 | 1.51 | -0.37 | 0.71 | 0.49 |
| **Ethnicity=Black** | -0.26 | 0.77 |  | 0.26 | -0.76 | 0.24 | 0.47 | 1.27 | -1.03 | 0.30 | 1.73 |
| **Ethnicity=Mixed** | -0.04 | 0.96 |  | 0.29 | -0.61 | 0.54 | 0.54 | 1.71 | -0.13 | 0.90 | 0.16 |
| **Ethnicity=Not known** | -0.26 | 0.77 |  | 0.23 | -0.71 | 0.18 | 0.49 | 1.20 | -1.15 | 0.25 | 2.01 |
| **Ethnicity=White** | -0.35 | 0.70 |  | 0.23 | -0.80 | 0.09 | 0.45 | 1.10 | -1.56 | 0.12 | 3.06 |
| **Employment\_Status=Not known** | 0.28 | 1.32 |  | 0.12 | 0.04 | 0.52 | 1.04 | 1.68 | 2.26 | 0.02 | 5.38 |
| **Employment\_Status=Other (please specify)** | 0.18 | 1.19 |  | 0.12 | -0.05 | 0.40 | 0.95 | 1.49 | 1.53 | 0.13 | 2.99 |
| **Employment\_Status=Retired** | -0.20 | 0.82 |  | 0.12 | -0.43 | 0.04 | 0.65 | 1.04 | -1.64 | 0.10 | 3.30 |
| **Employment\_Status=Student, full-time** | -0.14 | 0.87 |  | 0.23 | -0.59 | 0.31 | 0.55 | 1.37 | -0.61 | 0.54 | 0.88 |
| **Employment\_Status=Student, part-time** | 0.37 | 1.45 |  | 0.26 | -0.13 | 0.88 | 0.87 | 2.42 | 1.44 | 0.15 | 2.75 |
| **Employment\_Status=Unemployed** | 0.32 | 1.38 |  | 0.12 | 0.08 | 0.56 | 1.08 | 1.75 | 2.59 | 0.01 | 6.71 |
| **Employment\_Status=Wage earner, full-time** | -0.09 | 0.92 |  | 0.10 | -0.28 | 0.11 | 0.76 | 1.11 | -0.87 | 0.38 | 1.38 |
| **Employment\_Status=Wage earner, part-time** | -0.06 | 0.94 |  | 0.11 | -0.28 | 0.16 | 0.76 | 1.17 | -0.54 | 0.59 | 0.77 |
| **Weight1** | -0.00 | 1.00 |  | 0.00 | -0.00 | 0.00 | 1.00 | 1.00 | -0.02 | 0.98 | 0.03 |
| **month=2** | -0.22 | 0.80 |  | 0.08 | -0.37 | -0.07 | 0.69 | 0.93 | -2.90 | <0.005 | 8.05 |
| **month=3** | -0.36 | 0.70 |  | 0.10 | -0.55 | -0.17 | 0.58 | 0.85 | -3.65 | <0.005 | 11.91 |
| **month=4** | -0.23 | 0.79 |  | 0.07 | -0.37 | -0.10 | 0.69 | 0.91 | -3.30 | <0.005 | 10.03 |
| **month=5** | -0.51 | 0.60 |  | 0.07 | -0.64 | -0.38 | 0.53 | 0.69 | -7.56 | <0.005 | 44.45 |
| **month=6** | -0.76 | 0.47 |  | 0.09 | -0.93 | -0.59 | 0.39 | 0.56 | -8.64 | <0.005 | 57.36 |
| **month=7** | -0.42 | 0.66 |  | 0.09 | -0.59 | -0.25 | 0.56 | 0.78 | -4.85 | <0.005 | 19.66 |
| **month=8** | -0.62 | 0.54 |  | 0.09 | -0.80 | -0.45 | 0.45 | 0.64 | -6.85 | <0.005 | 36.95 |
| **month=9** | -0.75 | 0.47 |  | 0.06 | -0.87 | -0.62 | 0.42 | 0.54 | -11.86 | <0.005 | 105.45 |
| **month=10** | -0.19 | 0.83 |  | 0.07 | -0.32 | -0.06 | 0.72 | 0.94 | -2.87 | <0.005 | 7.95 |
| **month=11** | -0.26 | 0.77 |  | 0.07 | -0.40 | -0.13 | 0.67 | 0.88 | -3.74 | <0.005 | 12.42 |
| **month=12** | -0.46 | 0.63 |  | 0.18 | -0.82 | -0.11 | 0.44 | 0.90 | -2.55 | 0.01 | 6.54 |
| **NumPerProgram=2** | 1.76 | 5.84 |  | 0.78 | 0.23 | 3.29 | 1.26 | 26.97 | 2.26 | 0.02 | 5.39 |
| **NumPerProgram=3** | 0.84 | 2.32 |  | 0.43 | -0.00 | 1.69 | 1.00 | 5.41 | 1.95 | 0.05 | 4.29 |
| **NumPerProgram=4** | 0.43 | 1.54 |  | 0.37 | -0.29 | 1.15 | 0.75 | 3.16 | 1.17 | 0.24 | 2.05 |
| **NumPerProgram=5** | 0.58 | 1.78 |  | 0.35 | -0.12 | 1.27 | 0.89 | 3.57 | 1.63 | 0.10 | 3.28 |
| **NumPerProgram=6** | 0.83 | 2.29 |  | 0.34 | 0.17 | 1.49 | 1.18 | 4.44 | 2.45 | 0.01 | 6.13 |
| **NumPerProgram=7** | 0.59 | 1.81 |  | 0.33 | -0.06 | 1.24 | 0.94 | 3.46 | 1.78 | 0.07 | 3.75 |
| **NumPerProgram=8** | 0.84 | 2.31 |  | 0.33 | 0.20 | 1.48 | 1.22 | 4.38 | 2.56 | 0.01 | 6.59 |
| **NumPerProgram=9** | 0.51 | 1.67 |  | 0.33 | -0.13 | 1.15 | 0.88 | 3.17 | 1.57 | 0.12 | 3.11 |
| **NumPerProgram=10** | 0.67 | 1.95 |  | 0.33 | 0.02 | 1.31 | 1.02 | 3.70 | 2.03 | 0.04 | 4.57 |
| **NumPerProgram=11** | 0.65 | 1.92 |  | 0.33 | 0.02 | 1.29 | 1.02 | 3.64 | 2.01 | 0.04 | 4.48 |
| **NumPerProgram=12** | 0.51 | 1.67 |  | 0.33 | -0.13 | 1.15 | 0.88 | 3.16 | 1.57 | 0.12 | 3.11 |
| **NumPerProgram=13** | 0.62 | 1.86 |  | 0.32 | -0.02 | 1.25 | 0.98 | 3.50 | 1.91 | 0.06 | 4.16 |
| **NumPerProgram=14** | 0.81 | 2.25 |  | 0.32 | 0.18 | 1.44 | 1.20 | 4.23 | 2.51 | 0.01 | 6.38 |
| **NumPerProgram=15** | 0.58 | 1.79 |  | 0.32 | -0.05 | 1.21 | 0.95 | 3.37 | 1.81 | 0.07 | 3.82 |
| **NumPerProgram=16** | 0.67 | 1.96 |  | 0.32 | 0.04 | 1.31 | 1.04 | 3.69 | 2.07 | 0.04 | 4.70 |
| **NumPerProgram=17** | 0.50 | 1.65 |  | 0.32 | -0.14 | 1.13 | 0.87 | 3.11 | 1.54 | 0.12 | 3.02 |
| **NumPerProgram=18** | 0.67 | 1.96 |  | 0.33 | 0.03 | 1.31 | 1.03 | 3.71 | 2.06 | 0.04 | 4.67 |
| **NumPerProgram=19** | 0.62 | 1.86 |  | 0.33 | -0.02 | 1.27 | 0.98 | 3.55 | 1.89 | 0.06 | 4.10 |
| **NumPerProgram=20** | 0.16 | 1.17 |  | 0.36 | -0.55 | 0.87 | 0.58 | 2.38 | 0.43 | 0.66 | 0.59 |
| **NumPerProgram=21** | 1.15 | 3.15 |  | 0.37 | 0.42 | 1.87 | 1.53 | 6.50 | 3.11 | <0.005 | 9.06 |
| **NumPerProgram=22** | 0.81 | 2.24 |  | 0.35 | 0.13 | 1.49 | 1.13 | 4.43 | 2.32 | 0.02 | 5.63 |
| **NumPerProgram=23** | 0.61 | 1.84 |  | 0.38 | -0.15 | 1.36 | 0.86 | 3.90 | 1.58 | 0.11 | 3.13 |
| **NumPerProgram=24** | -0.02 | 0.98 |  | 0.36 | -0.74 | 0.69 | 0.48 | 1.99 | -0.06 | 0.95 | 0.08 |
| **NumPerProgram=25** | 0.10 | 1.11 |  | 0.37 | -0.62 | 0.83 | 0.54 | 2.30 | 0.28 | 0.78 | 0.36 |
| **NumPerProgram=26** | 1.05 | 2.86 |  | 0.36 | 0.34 | 1.76 | 1.41 | 5.82 | 2.90 | <0.005 | 8.06 |
| **NumPerProgram=28** | 1.12 | 3.08 |  | 0.37 | 0.40 | 1.85 | 1.49 | 6.37 | 3.03 | <0.005 | 8.67 |
| **NumPerProgram=33** | 1.36 | 3.88 |  | 0.39 | 0.59 | 2.12 | 1.81 | 8.32 | 3.49 | <0.005 | 11.00 |
| **NumPerProgram=34** | 1.09 | 2.98 |  | 0.38 | 0.35 | 1.83 | 1.43 | 6.21 | 2.90 | <0.005 | 8.09 |
| **NumPerProgram=35** | 1.26 | 3.53 |  | 0.38 | 0.52 | 2.00 | 1.69 | 7.39 | 3.34 | <0.005 | 10.25 |
| **NumPerProgram=37** | 1.98 | 7.24 |  | 0.40 | 1.20 | 2.75 | 3.34 | 15.72 | 5.01 | <0.005 | 20.78 |
| **NumPerProgram=39** | -0.68 | 0.51 |  | 0.43 | -1.53 | 0.17 | 0.22 | 1.18 | -1.57 | 0.12 | 3.11 |
| **NumPerProgram=40** | 0.12 | 1.13 |  | 0.39 | -0.65 | 0.89 | 0.52 | 2.43 | 0.30 | 0.76 | 0.39 |
| **NumPerProgram=43** | -14.57 | 0.00 |  | 291.87 | -586.62 | 557.48 | 0.00 | 1.29e+242 | -0.05 | 0.96 | 0.06 |

## python code

### Importing libraries

import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

import matplotlib

from datetime import datetime

import seaborn as sns

from scipy import stats

from sklearn.preprocessing import MinMaxScaler,OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.pipeline import Pipeline

from sklearn import set\_config

from sklearn.model\_selection import KFold, StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import GridSearchCV

import scikitplot as skplt

from sklearn import metrics

from sklearn.metrics import roc\_curve

from sklearn.inspection import permutation\_importance

from rfpimp import \*

set\_config(display='diagram')

%matplotlib inline

### Loading dataset

df\_adult = pd.read\_excel("ML\_Adults\_data.xlsx")

pd.set\_option('display.max\_rows', None

pd.set\_option('display.max\_columns',df\_adult.shape[1]+1

print('The number of rows and columns in the adult dataset are', df\_adult.shape)

### Data Cleaning

#### Deriving new features

#1 month

df\_adult['month'] = pd.DatetimeIndex(df\_adult['Programme\_Start\_Date']).month

#2 adding number of participants per group in a program session

df = df\_adult[['ProgrammeName','Programme\_Start\_Date']]

grp = df.groupby(['ProgrammeName']).count()#.sort\_values(['Programme\_Start\_Date'])

grp.reset\_index(inplace = True)

grp = grp.rename(columns = {'Programme\_Start\_Date' : 'NumPerProgram'})

grp.head(2)

df\_adult = pd.merge(df\_adult, grp, how="left", on="ProgrammeName")

## EDA

### Commissionaire

There are 22 Commissionaires. The programs p1 to p52 are conducted at Commissionair-1, p53 to p87 at Commissionair-2 and so on. Program names are serially ordered for a given commissioner and they are unique to a commissioner.

### Program Name

There are 642 unique program sessions conducted across different commissioners and the programs are named as p1, p2, p3, .... p642. Exploration of the data shows that program names at a given commissionaire correspond to program start date.

There are 52 programs in commissionair-1 with 52 unique start dates. Thus, program names correspond to program start date. The name given to a program no matter what cannot influence the participants outcome. Hence this feature can be removed. The month part of the program start date is selected as a feature. This will group together different programs which occurred in month which could be a characteristic feature for determining participants engagement.

#### Probability of dropout VS Programme start month

df = df\_adult[['month','Engagement\_status']]

df =df[~(df['Engagement\_status'] =='NonInitiator')]

(df.groupby('month')['Engagement\_status'] #group rows by month and for each month count the rows by engagement\_status

.value\_counts(normalize=True) # values\_count ignores null rows

.rename('Probability') # renaming the column obtained by value\_counts to probability

.reset\_index() # If the index has multiple levels, we can reset a subset of them:(remove index month)

.pipe((sns.catplot,'data'), x='month',y='Probability',hue='Engagement\_status', height=6, aspect=8/6, palette=('orange','teal')))

plt.title('Probability of dropout VS Programme start month', fontsize=20, weight = 'bold')

plt.xlabel('Month', color='#AF5050', labelpad=10, fontsize=25, weight = 'bold')

plt.ylabel('Probability', color='#af5050', labelpad=10, fontsize=25, weight = 'bold')

plt.xticks(fontsize=14, weight = 'bold')

plt.yticks(fontsize=14, weight = 'bold')

#### participant distribution by program group size

df = df\_adult[~(df\_adult['Engagement\_status'] == 'NonInitiator')]

sns.displot(df, x="NumPerProgram",color="y",element="step")

plt.title('Number of programs with a given group size', fontsize=15, weight = 'bold')

plt.xlabel("number of participants in the group", labelpad=10, fontsize=15, weight = 'bold')

plt.ylabel("Count", labelpad=10, fontsize=15, weight = 'bold')

plt.xticks(fontsize=15, weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

#### probability of Dropout or completer by program Size

p = 'gist\_earth\_r'

df = df\_adult[~(df\_adult['Engagement\_status'] == 'NonInitiator')]

group, column = 'NumPerProgram', 'Engagement\_status'

grp\_obj = df.groupby(group)[column]

grp\_obj.value\_counts(normalize=True).rename('Probability').reset\_index().pipe((sns.catplot,'data'),palette =p ,height=6, aspect=15/6, x=group, y='Probability',hue=column,kind='bar')

plt.title(' probability of Dropout or completer by programSize', fontsize=20, weight = 'bold')

plt.xlabel("Number of participants in a program group", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Probability", labelpad=10, fontsize=20, weight = 'bold')

plt.xticks(fontsize=15, weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

plt.legend(fontsize=20)

#### Participant distribution by program engagement status

Total = df\_adult['Engagement\_status'].value\_counts().sum()

Dropout = df\_adult['Engagement\_status'].value\_counts()[0]\*100/Total

Completer = df\_adult['Engagement\_status'].value\_counts()[1]\*100/Total

NonInitiator =df\_adult['Engagement\_status'].value\_counts()[2]\*100/Total

print('\n Dropout% = ',Dropout,'\n Completer% = ', Completer, '\n NonInitiator% = ',NonInitiator)

plt.figure(figsize=(9,6))

colors\_list = ['c','c','c']

x = df\_adult['Engagement\_status'].value\_counts().sort\_index()

ax = x.plot(kind='bar',edgecolor=None,legend=True,width = 0.8,rot=0,color = colors\_list)

plt.legend(loc='upper right',fontsize=13)

plt.title('Distritibution of participants at MoreLife by their Programe engagement status \n', fontsize=20, weight = 'bold')

plt.xlabel("Engagement status", labelpad=10, fontsize=20,font ='bold')

plt.ylabel("Count", labelpad=10, fontsize=20,font ='bold')

plt.xticks(fontsize=15, weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

plt.legend(fontsize=0)

#### Participant distribution by gender

plt.figure(figsize=(6,4))

plt.title(' participant distribution by gender', fontsize=20, weight = 'bold')

plt.xlabel("Gender", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Count", labelpad=10, fontsize=20, weight = 'bold')

#plt.style.use('fivethirtyeight')

ax = sns.countplot(x="Gender", data=df\_adult, palette="copper")

plt.xticks(fontsize=15, weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

#### Age distribution

plt.figure(figsize=(16, 6))

plt.title('Participant distribution by their age', fontsize=20, weight = 'bold')

plt.xlabel("Age", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Count", labelpad=10, fontsize=20, weight = 'bold')

plt.xticks(fontsize=15, rotation=90,fontname = "Times New Roman",weight = 'bold')

sns.countplot(data=df\_adult, x='Age', order=df\_adult.Age.value\_counts().sort\_index().index)

plt.yticks(fontsize=15, weight = 'bold')

#### Probability of Dropout, completer by age

group, column = 'Age', 'Engagement\_status'

grp\_obj = df.groupby(group)[column]

grp\_obj.value\_counts(normalize=True).rename('Probability').reset\_index().pipe((sns.catplot,'data'), x=group, y='Probability',hue=column,palette="Set2",kind='bar',height=12, aspect=28/12)

plt.title('Probability of Dropout, completer by age', fontsize=40, weight = 'bold')

plt.xlabel("Age", labelpad=10, fontsize=40, weight = 'bold')

plt.ylabel("Probability", labelpad=10, fontsize=40, weight = 'bold')

plt.xticks(fontsize=25, rotation=90,weight = 'bold')

plt.yticks(fontsize=25, weight = 'bold')

plt.legend(fontsize=40)

#### participant Engagement status by ethnicity

plt.figure(figsize=(16, 4))

plt.title(' Participant distribution by their ethnicity', fontsize=20, weight = 'bold')

sns.countplot(data=df\_adult, x='Ethnicity', order=df\_adult.Ethnicity.value\_counts().sort\_index().index)

plt.xlabel("Ethnicity", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Count", labelpad=10, fontsize=20, weight = 'bold')

plt.xticks(fontsize=20,weight = 'bold')

plt.yticks(fontsize=20, weight = 'bold')

#### Probability of Dropout, completer by Ethnicity

df = df\_adult[~(df\_adult['Engagement\_status'] == 'NonInitiator')]

group, column = 'Ethnicity', 'Engagement\_status'

grp\_obj = df.groupby(group)[column]

grp\_obj.value\_counts(normalize=True).rename('Probability').reset\_index().pipe((sns.catplot,'data'), x=group, y='Probability',palette="Set2",hue=column,kind='bar',height=4, aspect=12/4)

plt.title('Probability of Dropout, completer by Ethnicity', fontsize=20, weight = 'bold')

plt.xlabel("Ethnicity", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Probability", labelpad=10, fontsize=20, weight = 'bold')

#### Participant distribution by their Employment Status

plt.figure(figsize=(18, 4))

sns.countplot(data=df\_adult, x='Employment\_Status',palette="Set2" ,order=df\_adult.Employment\_Status.value\_counts().sort\_index().index)

plt.title('Participant distribution by their Employment\_Status', fontsize=20, weight = 'bold')

plt.xlabel("Employment\_status", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("count", labelpad=10, fontsize=20, weight = 'bold')

plt.xticks(fontsize=15, rotation=90, weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

#### probability of Dropout, completer by Employment Status

df = df\_adult[~(df\_adult['Engagement\_status'] == 'NonInitiator')]

group, column = 'Employment\_Status', 'Engagement\_status'

grp\_obj = df.groupby(group)[column]

grp\_obj.value\_counts(normalize=True).rename('Probability').reset\_index().pipe((sns.catplot,'data'),palette="Set3" ,x=group, y='Probability',hue=column,kind='bar',height=4, aspect=13/4)

plt.title('probability of Dropout, completer by Employment\_Status', fontsize=20, weight = 'bold')

plt.xlabel("Employment\_status", labelpad=10, fontsize=20, weight = 'bold')

plt.ylabel("Probability", labelpad=10, fontsize=20, weight = 'bold')

plt.xticks(fontsize=15, rotation=90,weight = 'bold')

plt.yticks(fontsize=15, weight = 'bold')

#### Average weight loss after first week

df = df\_adult[~(df\_adult['Engagement\_status'] == 'NonInitiator')]

df = df[['dw21','Engagement\_status']]

df = df.dropna()

df.groupby('Engagement\_status').mean().plot(kind = 'bar')

plt.title('Average weight loss after first week', fontsize=15, weight = 'bold')

plt.rcParams['axes.axisbelow'] = True

plt.xlabel("Engagement\_status", labelpad=14, fontsize=12, weight = 'bold')

plt.ylabel("Average weight loss", labelpad=14, fontsize=12, weight = 'bold')

plt.xticks(fontsize=12,weight = 'bold')

plt.yticks(fontsize=12, weight = 'bold')

plt.legend("weightloss",fontsize=0)

matplotlib.pyplot.grid(axis = 'y', linestyle='-.') # b=None, linestyle='-', linewidth=3

#### Data cleaning

def fillnull(column):

for c\_id in repeating\_clients: # for a given client with client\_id = c\_id and c\_id is repeating

filt = (~df\_adult[column].isnull()) & (df\_adult["Client\_Id"]==c\_id) #filter rows with ethnicity notnull and for client\_id = c\_id

data = df\_adult[filt][column] # series with ethnicity in rows not null for a client with client\_id = c\_id

data\_2 = df\_adult[filt][[column,'Referred\_Date']]

if(data.unique().shape[0] == 1): # to make sure that ethnicity series has only 1 unique value(one row) which replaces nulls in other rows

fillby = data.iloc[0] # as only there is one row in data\_select

df\_adult.loc[df\_adult['Client\_Id'] == c\_id, [column]] = fillby # gives a warning but works

elif(data.unique().shape[0] > 1): # if many values of ethnicity for the same client

#print(data\_2,'\n') # df with ethnicity in rows not null for a client with client\_id = c\_id

data\_2 = data\_2[data\_2['Referred\_Date']==max(data\_2['Referred\_Date'])]

fillby = data\_2[column].iloc[0] # using the latest recorded value for ethnicity

#print("fill nulls by = ",fillby )

df\_adult.loc[(df\_adult['Client\_Id'] == c\_id) & (df\_adult[column].isnull()) , [column]] = fillby # this will allow to select the value that can be used replace nulls. I am filling nulls by the value at most recent refered date

#print('\n')

fillnull(column='Ethnicity')

#### filling nulls in weight1 with pre-weight values and vice-versa

column1 = 'Weight1'

column2 = 'PreWeight'

for i,r in df\_adult[[column1,column2]].iterrows():

if(pd.isna(df\_adult.loc[i,column1])):

df\_adult.loc[i,column1] = r[column2]

elif(pd.isna(df\_adult.loc[i,column2])):

df\_adult.loc[i,column2] = r[column1]

#### dropping outliers

def dropoutliers(column,df):

mean\_value = df[column].mean()

std\_dev = df[column].std()

filt = ~(((df[column] - mean\_value) / std\_dev).abs() > 3)

df = df[filt]

return df

# droping outliers in Weight1

print(df\_adult.shape[0])

df\_adult = dropoutliers(column="Weight1",df =df\_adult)

print(df\_adult.shape[0])

## Logistic Regression

def runLogREG(df,columns,cat,num,split\_ratio,cmap):

df = df # dataFrame

columns = columns # features

df = df[columns]

df = df[columns].dropna()

print('\n shape of the dataset is',df.shape)

print("\n Data distribution \n")

print(df['Engagement\_status'].value\_counts())

print("\n features included")

catageorical\_variables = cat # categorical features

numerical\_variables = num # numerical features

split\_ratio = split\_ratio # test train split ratio

cmap = cmap # color of confusion matrix

print(catageorical\_variables)

print(numerical\_variables)

X = df.drop(['Engagement\_status'],axis=1).astype(str) #features

y = df['Engagement\_status'] #label

# spliting data into train and test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=split\_ratio, random\_state=0)

#Transforming categorical features to numerical values using oneHotEncoder and sacling numerical features

transform\_cat = OneHotEncoder(handle\_unknown='ignore')

transform\_num = MinMaxScaler()

#Logistic Regression

model\_lr = LogisticRegression(max\_iter=5000)

preprocess\_lr = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

pipe\_lr = Pipeline(steps = [('preprocess',preprocess\_lr),('LogisticClassifier',model\_lr)])

#Searching for best parameters

param\_grid = {

'LogisticClassifier\_\_penalty': [ 'l1','l2'],

'LogisticClassifier\_\_C': [ 10000,5000 ,1000,500,100,10,1,0.1],

'LogisticClassifier\_\_solver' : ['newton-cg', 'lbfgs', 'liblinear']

}

cv = StratifiedKFold(n\_splits=10, random\_state=3, shuffle=True)

model\_lr = LogisticRegression(max\_iter=5000)

search = GridSearchCV(pipe\_lr, param\_grid, n\_jobs=-1, cv=cv, scoring='accuracy', error\_score=0)

search.fit(X\_train, y\_train)

bestParams = search.best\_params\_

print('\n',"Logistic Regression parameters: \n",bestParams)

c = bestParams["LogisticClassifier\_\_C"]

p = bestParams["LogisticClassifier\_\_penalty"]

s = bestParams["LogisticClassifier\_\_solver"]

finaleModel = LogisticRegression(max\_iter=5000, C= c, penalty = p, solver = s)

pipe\_lr = Pipeline(steps = [('preprocess',preprocess\_lr),('LogisticClassifier',finaleModel)])

pipe\_lr.fit(X\_train, y\_train)

cv = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

scores = cross\_val\_score(pipe\_lr, X\_train, y\_train, cv=cv, scoring='accuracy')

print('\n','Cross-Val Accuracy\_train = %.2f' % scores.mean())

#Model Evaluation on test data

print("\n Cross-Val Accuracy\_test: %.2f" % pipe\_lr.score(X\_test, y\_test))

print('\n')

#Confusion Matrix Testing evaluation

y\_pred = pipe\_lr.predict(X\_test)

print(metrics.classification\_report(y\_test,y\_pred),'\n')

skplt.metrics.plot\_confusion\_matrix(y\_test, y\_pred, normalize=True, cmap = cmap)

#ROC-Curve and AUC value# test

print('\n')

metrics.plot\_roc\_curve(pipe\_lr, X\_test, y\_test)

matplotlib.pyplot.grid(axis = 'y', linestyle='-')

plt.show()

#Permutation feature Importance

result = permutation\_importance(pipe\_lr, X, y, n\_repeats=100,random\_state=0)

# plot feature importance

imp = importances(pipe\_lr, X\_test, y\_test) # permutation

fi = plot\_importances(imp)

return fi

df = pd.read\_csv('data.csv')

df =df.drop(axis = 1,columns = 'Unnamed: 0')

columns = ['Commissioner','Weight1','Self-reported\_Height','Gender','Age','Ethnicity','Employment\_Status','Engagement\_status','month','dw21','NumPerProgram','Total\_Programme\_sessions']

cat = ['Commissioner', 'Gender', 'Ethnicity', 'Employment\_Status', 'month']

num = ['Age','Weight1','dw21','NumPerProgram','Self-reported\_Height','Total\_Programme\_sessions']

fi = runLogREG(df,columns,cat,num,split\_ratio = 0.2,cmap = 'RdGy')

fi

## Random forest:

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import OrdinalEncoder

df = pd.read\_csv('data.csv')

df =df.drop(axis = 1,columns = 'Unnamed: 0')

columns = ['Commissioner','Weight1','Self-reported\_Height','Gender','Age','Ethnicity','Employment\_Status','Engagement\_status','month','dw21','NumPerProgram','Total\_Programme\_sessions']

df = df[columns]

df = df[columns].dropna()

X = df.drop(['Engagement\_status'],axis=1).astype(str) #features

y = df['Engagement\_status'] #label

catageorical\_variables = ['Commissioner', 'Gender', 'Ethnicity', 'Employment\_Status', 'month']

numerical\_variables = ['Age','Weight1','dw21','NumPerProgram','Self-reported\_Height','Total\_Programme\_sessions']

split\_ratio = 0.7

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=split\_ratio, random\_state=0)

cv = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

transform\_num = MinMaxScaler()

splitting\_criteria = 'gini'

num\_trees = 1000

model\_rf = RandomForestClassifier()

preprocess\_rf = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

pipe\_rf = Pipeline(steps = [('preprocess',preprocess\_rf),('RF',model\_rf)])

max\_features = [0.6,0.7,0.8] # percent values

param\_grid = {

'RF\_\_bootstrap' : [True],

'RF\_\_criterion' : ['gini','entropy'],

'RF\_\_max\_depth' : [95,100,110], # maximum depth of a tree

'RF\_\_max\_features' : max\_features, # max features(subset) to include per tree

'RF\_\_min\_samples\_leaf' : [30, 40,50], # min samples required per leaf node

'RF\_\_min\_samples\_split' : [120, 130, 140], # min samples required per node before split

'RF\_\_n\_estimators' : [170, 180, 200] # num of trees

}

cv = StratifiedKFold(n\_splits=5, random\_state=1, shuffle=True)

search = GridSearchCV(pipe\_rf, param\_grid, n\_jobs=-1, cv=cv, scoring='accuracy')

search.fit(X\_train, y\_train)

prm = search.best\_params\_

n\_estimators = prm['RF\_\_n\_estimators']

criterion = prm['RF\_\_criterion']

max\_depth = prm['RF\_\_max\_depth']

min\_samples\_split = prm['RF\_\_min\_samples\_split']

min\_samples\_leaf = prm['RF\_\_min\_samples\_leaf']

max\_features = prm['RF\_\_max\_features']

bootstrap = prm['RF\_\_bootstrap']

finaleModel = RandomForestClassifier(n\_estimators=n\_estimators, criterion=criterion, max\_depth=max\_depth, min\_samples\_split=min\_samples\_split, min\_samples\_leaf=min\_samples\_leaf, max\_features=max\_features, bootstrap=bootstrap)

preprocess\_rf = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

pipe\_rf = Pipeline(steps = [('preprocess',preprocess\_rf),('RF',finaleModel)])

pipe\_rf.fit(X\_train, y\_train)

cv = StratifiedKFold(n\_splits=10, random\_state=1, shuffle=True)

scores = cross\_val\_score(pipe\_rf, X\_train, y\_train, cv=cv, scoring='accuracy')

print('Accuracy =',scores.mean())

print("model score or (Accuracy): %.4f" % pipe\_rf.score(X\_test, y\_test))

import scikitplot as skplt

y\_pred = pipe\_rf.predict(X\_test)

skplt.metrics.plot\_confusion\_matrix(y\_test, y\_pred, normalize=True, cmap = 'RdGy')

from sklearn import metrics

print(metrics.classification\_report(y\_test,y\_pred))

print("\n")

print("model score: %.3f" % pipe\_rf.score(X\_test, y\_test))

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print("Precision score for dropout prediction:",metrics.precision\_score(y\_test, y\_pred, average="binary", pos\_label='Dropout'))

print("Recall score for dropout prediction:",metrics.recall\_score(y\_test, y\_pred, average="binary", pos\_label='Dropout'))

from sklearn.metrics import roc\_curve # test

metrics.plot\_roc\_curve(pipe\_rf, X\_test, y\_test)

matplotlib.pyplot.grid(axis = 'y', linestyle='-.')

plt.show()

## Naïve bayes

df = pd.read\_csv('data.csv')

pd.set\_option('display.max\_columns',df.shape[1]+1)

columns = ['Commissioner','Weight1','Gender','Age','Ethnicity','Employment\_Status','Engagement\_status','month','dw21','NumPerProgram','Self-reported\_Height','Total\_Programme\_sessions']

catageorical\_variables = ['Commissioner', 'Gender', 'Ethnicity', 'Employment\_Status', 'month']

numerical\_variables = ['Age','Weight1','dw21','NumPerProgram','Self-reported\_Height','Total\_Programme\_sessions']

df = df[columns].dropna()

df[catageorical\_variables] = df[catageorical\_variables].astype(str)

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

for column in df[catageorical\_variables]:

df[column] = le.fit\_transform(df[column])

X = df.drop(['Engagement\_status'],axis=1).astype(str)

y = df['Engagement\_status']

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 0)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

from sklearn.naive\_bayes import GaussianNB

classifier = GaussianNB()

classifier.fit(X\_train, y\_train)

score = cross\_val\_score(classifier, X\_train, y\_train, cv=cv, n\_jobs=1)

score.mean()

y\_pred = classifier.predict(X\_test)

print("model score or (Accuracy): %.2f" % classifier.score(X\_test, y\_test))

import scikitplot as skplt

skplt.metrics.plot\_confusion\_matrix(y\_test, y\_pred, normalize=True, cmap = 'YlGnBu')

metrics.plot\_roc\_curve(classifier, X\_test, y\_test)

plt.show()

## Multilayer perceptron:

df = pd.read\_csv('data.csv')

df =df.drop(axis = 1,columns = 'Unnamed: 0')

columns = ['Commissioner','Weight1','Self-reported\_Height','Gender','Age','Ethnicity','Employment\_Status','Engagement\_status','month','dw21','NumPerProgram','Total\_Programme\_sessions']

df = df[columns].dropna()

X = df.drop(['Engagement\_status'],axis=1).astype(str) #features

y = df['Engagement\_status'] #label

catageorical\_variables = ['Commissioner', 'Gender', 'Ethnicity', 'Employment\_Status', 'month']

numerical\_variables = ['Age','Weight1','dw21','NumPerProgram','Self-reported\_Height','Total\_Programme\_sessions']

from sklearn.neural\_network import MLPClassifier

transform\_cat = OneHotEncoder(handle\_unknown='ignore')

preprocess\_ann = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

model = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden\_layer\_sizes=(5, 2), random\_state=1, max\_iter=5000)

pipe = Pipeline(steps = [('preprocess\_ann',preprocess\_ann),('ANN',model)])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

score = cross\_val\_score(pipe, X\_train, y\_train, cv=10, n\_jobs=1)

score.mean()

model\_ann = MLPClassifier(solver='lbfgs', alpha=1e-5, hidden\_layer\_sizes=(5, 2), random\_state=1, max\_iter=5000)

preprocess\_ann = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

pipe = Pipeline(steps = [('preprocess',preprocess\_ann),('ann',model\_ann)])

param\_grid = {

'ann\_\_solver' : ['lbfgs','sgd','adam'],

'ann\_\_max\_iter' : [10000 ],

'ann\_\_alpha' : [0.0001, 100],

'ann\_\_hidden\_layer\_sizes' : [5,80],

'ann\_\_random\_state' : [0],

'ann\_\_learning\_rate' : ['constant', 'invscaling', 'adaptive']

}

cv = StratifiedKFold(n\_splits=5, random\_state=1, shuffle=True)

search = GridSearchCV(pipe, param\_grid, n\_jobs=-1, cv=cv, scoring='accuracy')

search.fit(X\_train, y\_train)

print("Accuracy = %0.3f:" % search.best\_score\_)

print(search.best\_params\_)

prm = search.best\_params\_

solver = prm['ann\_\_solver']

alpha = prm['ann\_\_alpha']

hidden\_layer\_sizes = prm['ann\_\_hidden\_layer\_sizes']

max\_iter = prm['ann\_\_max\_iter']

learning\_rate = prm['ann\_\_learning\_rate']

model\_ann = MLPClassifier(solver=solver, learning\_rate=learning\_rate, alpha= alpha, hidden\_layer\_sizes=hidden\_layer\_sizes, random\_state=1, max\_iter=max\_iter)

preprocess\_ann = ColumnTransformer(transformers =[('catToNum', transform\_cat, catageorical\_variables),('scalingNumerical', transform\_num, numerical\_variables)])

pipe = Pipeline(steps = [('preprocess',preprocess\_ann),('ann',model\_ann)])

pipe.fit(X\_train, y\_train)

y\_pred = pipe.predict(X\_test)

print("model score or (Accuracy): %.4f" % pipe.score(X\_test, y\_test))

import scikitplot as skplt

skplt.metrics.plot\_confusion\_matrix(y\_test, y\_pred, normalize=True)

print("\n")

print("model score: %.3f" % pipe.score(X\_test, y\_test))

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

print("Precision score for dropout prediction:",metrics.precision\_score(y\_test, y\_pred, average="binary", pos\_label='Dropout'))

print("Recall score for dropout prediction:",metrics.recall\_score(y\_test, y\_pred, average="binary", pos\_label='Dropout'))

metrics.plot\_roc\_curve(pipe, X\_test, y\_test)

plt.show()

## Code for Survival Analysis

df = pd.read\_csv('data.csv') # load new data here and do new data cleaning

df =df.drop(axis = 1,columns = 'Unnamed: 0')

columns = ['ProgrammeName', 'Commissioner', 'week1', 'week2', 'week3', 'week4', 'week5', 'week6', 'week7', 'week8', 'week9', 'week10','week11','week12','Engagement\_status','AttendedSessions','Total\_Programme\_sessions','Gender','Age','Ethnicity','Employment\_Status','Self-reported\_Height','Weight1','month','NumPerProgram','dw21']

df = df[columns]

# Adding new column 'survival\_time'

df['survival\_time'] = 0 # week after which a participant has discontinued from the program or week last attended

df['status'] = 1 # we have to redefine dropout. for now let all participants dropout, i.e status = 1

df['Total\_Programme\_sessions'].unique()

weeks = ['week1','week2','week3','week4','week5','week6','week7','week8' ,'week9','week10','week11','week12']

for i, r in df.iterrows():

for week in df[weeks]:

if(pd.isna(df.loc[i,week])):

df.loc[i,week] = 'N'

ReversedWeeks = []

for week in reversed(weeks):

ReversedWeeks.append(week)

for i, r in df.iterrows():

survivalTime = 12

for week in df[ReversedWeeks]:

if(df.loc[i,week] == 'N' ):

survivalTime = survivalTime - 1

else:

break

df.loc[i,'survival\_time'] = survivalTime

df[weeks + ['survival\_time', 'AttendedSessions', 'status']].head(2)

df['status'] = df['status'].astype('bool')

for i,r in df.iterrows():

if(r['survival\_time'] > 11):

df.loc[i,'status'] = False

from sksurv.nonparametric import kaplan\_meier\_estimator

from lifelines import KaplanMeierFitter, NelsonAalenFitter

plt.figure(figsize=(12,4))

kmf = KaplanMeierFitter()

kmf.fit(durations = df["survival\_time"],event\_observed = df["status"])

kmf.event\_table

kmf.survival\_function\_

kmf.plot()

plt.title("Probability of survival(Kaplan-Meier Estimator)", fontsize=16,weight = 'bold')

plt.ylabel("Probability for participant to continue $\hat{S}(week)$", labelpad=10, fontsize=14, weight = 'bold')

plt.xlabel("$Week$", labelpad=10, fontsize=14, weight = 'bold')

plt.xticks(fontsize=14,weight = 'bold')

plt.yticks(fontsize=14, weight = 'bold')

matplotlib.pyplot.grid(axis = 'both', linestyle='-')

# survival functions by group

def kaplanPlot(column):

plt.figure(figsize=(12,4))

for val in df[column].unique():

df2 = df[df[column]==val]

time, survival\_prob = kaplan\_meier\_estimator(df2["status"], df2["survival\_time"])

plt.step(time, survival\_prob, where="post", label="%s" % val)

c = "survival probability by " + column

plt.ylabel(c, labelpad=10, fontsize=12, weight = 'bold')

plt.xlabel("week", labelpad=10, fontsize=12, weight = 'bold')

plt.xticks(fontsize=12,weight = 'bold')

plt.yticks(fontsize=12, weight = 'bold')

matplotlib.pyplot.grid(axis = 'both', linestyle='-')

plt.legend(loc="best")

c = "Probability of survival(Kaplan-Meier Estimator) by " + column

plt.title(c, fontsize=16,weight = 'bold')

column = 'Employment\_Status'

kaplanPlot(column)

column = 'month'

kaplanPlot(column)

column = 'Gender'

kaplanPlot(column)

group1\_status = df[df['Gender'] == 'M']['status']

group1\_survival\_time = df[df['Gender'] == 'M']['survival\_time']

group2\_status = df[df['Gender'] == 'F']['status']

group2\_survival\_time = df[df['Gender'] == 'F']['survival\_time']

from lifelines.statistics import logrank\_test

test = logrank\_test(group1\_survival\_time,group2\_survival\_time,event\_observed\_A = group1\_survival\_time, event\_observed\_B =group2\_survival\_time)

test.print\_summary()

# COX model

from sksurv.preprocessing import OneHotEncoder

from sksurv.linear\_model import CoxPHSurvivalAnalysis

num\_df2 = OneHotEncoder().fit\_transform(df2)

from lifelines import CoxPHFitter

cox = CoxPHFitter()

cox.fit(num\_df2,'survival\_time',event\_col='status')

cox.print\_summary()

df2 =df.drop(axis = 1,columns = ['dw21','Self-reported\_Height'])

columns =['Commissioner','Gender','Age','Ethnicity','Employment\_Status','Weight1','month','NumPerProgram','survival\_time','status']

df2 = df2[columns]

df2 = df2.dropna()

kmf.fit(event\_observed = df2["status"], durations = df2["survival\_time"])

# Supervisor meetings

June 16th

Discussion on refining the project objectives:

* Key factors that affect the weight management
* Role of machine learning in weight management

July 1st

* Discussion about using Crisp-Dm methodology.
* Discussed dealing with data quality issues.
* Discussed about the main project objective, dropout probability calculation suggested Naive base, Random Forest algorithms.
* Suggestion to think about modeling and write background and rational

Supervisor meeting 28-July

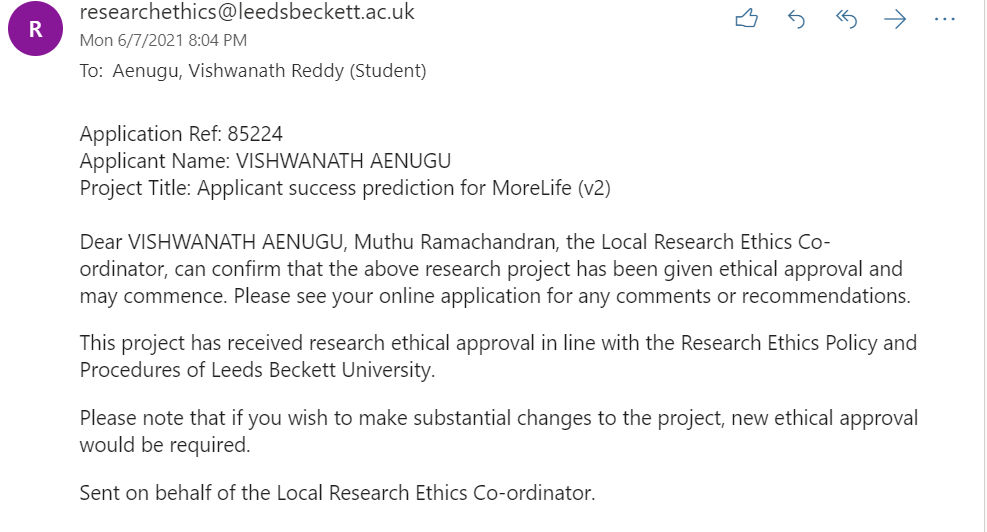
* Discussed handling data quality issues, CRISP-DM, literature review

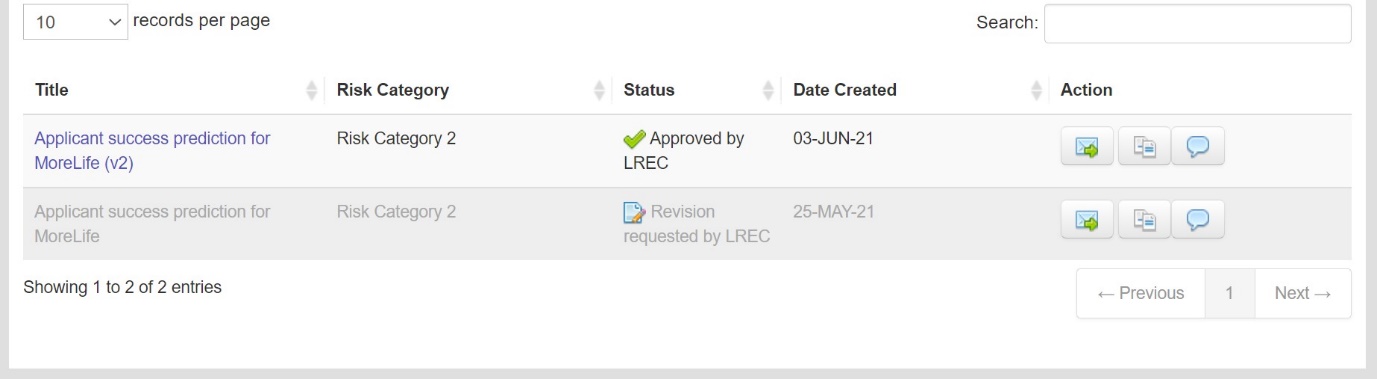
Email correspondences on 12 and 30th August:

* Suggestions on improving dissertation writing.

# project approval and ethics

Date: 7-June-2021





# Data Sharing agreement for this project



**Data Sharing Agreement**

PROJECT: Computing Science Projects

This Data Sharing Agreement ("Agreement") is between:

1. MoreLife (UK) Ltd, Churchwood Hall, Leeds Beckett University, Headingley Campus, Leeds, LS6 3QJ.

and

1. Leeds Beckett University (LBU), Headingley Campus, Leeds, LS6 3QJ.

This Agreement is to ensure that there are sufficient security controls/guarantees in place for the secure sharing of information in accordance with the General Data Protection Regulation (GDPR), the Data Protection Act 2018 and all other applicable Data Protection Laws.

Purpose of agreement

The Agreement defines how information or data may be shared between the Parties and the methods used by the Parties for the secure handling storage and processing of the data. The objectives of this Agreement are to:

* set out the operational arrangements for the sharing of personal data between the Parties; and
* set out the principles and commitments of the Parties when they process, store disclose, retain and dispose of the shared data.

Data Controllers

The Parties agree that they are Data Controllers and that as Controllers they are obliged to ensure that shared personal data is protected in line with the GDPR and other relevant Data Protection laws.

Processing of Shared Data

Each party remains responsible for the data they hold and process as Data Controller of that information. In relation to the sharing of information, each of the Parties shall take all measures necessary to ensure their respective compliance with all relevant legislation, including, but not limited to, regulations or restrictions regarding disclosure of information to third parties.

Each Party will comply fully with the common law duty of confidentiality, the General Data Protection Regulations and other applicable legislation. Particular attention must be paid to principle 6 and ensuring the security of information and systems. Each party will protect their information from unauthorised or unlawful processing, accidental loss, destruction or damage, and acknowledges they have implemented the required technical and organisational measures.

Each party will ensure that all staff with access to the information have received appropriate information governance and data protection training and are aware of the duties placed on those processing person-identifiable information. This includes ensuring that they have appropriate monitoring, policies and procedures in place for all staff, particularly regarding storage, security, retention and disposal of shared data.

Failure to meet the data processing standards within this Agreement will result in information not being shared which could result in the termination of the service agreement.

Data Subjects’ Rights

Under the GDPR, data subjects can ask to see the information that is held on computer and in paper records about them. If data subjects wish to know what information is held about them, requests must be put in writing to the Party processing the information following their official Subject Access Request process.

The Parties agree that the responsibility for complying with a request from a data subject falls on the Party receiving the request in respect of the Personal Data held by that Party.

The Parties agree to provide reasonable and prompt assistance (within 5 working days of such a request for assistance) as is necessary to each other to enable them to comply with a Data Subject Request and to respond to any other queries or complaints from Data Subjects.

Parties to this Agreement shall maintain a record of requests from data subjects, the decisions made and any information that was provided under the request. Records must include copies of the request for information, details of the data provided and where relevant, notes of any meeting, correspondence relating to the request.

Security of Shared Data

Regardless of how shared information is being accessed, processed or stored, security is considered of paramount importance. All information processed by both Parties must be held on secure servers, with access restricted to internal use by appropriately authorised members of staff.

As Data Controller for the information they process, each Party is expected to treat all information in accordance with the requirements of the GDPR and ensure that appropriate security controls are in place to protect the information from unauthorised access. This includes physical security, such as adhering to organisational clear desk policies and adequate protection for premises when unattended, to IT related security such as passwords, secure IDs and secure servers.

The Parties agree to implement appropriate technical and organisational measures to protect the Shared Personal Data in their possession against unauthorised or unlawful processing and against accidental loss, destruction, damage, alteration or disclosure, including but not limited to:

* Ensuring IT equipment, including portable equipment, is kept in lockable areas when unattended;
* ensuring that staff use appropriate secure passwords for logging into systems or databases containing the Personal Data;
* ensuring that all IT equipment is protected by antivirus software, firewalls, passwords and suitable encryption devices;
* limiting access to relevant databases and systems to officers, staff agents and contractors who need to have access to the Personal Data, and ensuring that passwords are changed and updated regularly to prevent inappropriate access when individuals are no longer engaged by the Party;
* conducting regular threat assessment or penetration testing on systems.
* Ensuring all staff handling Personal Data have been made aware of their responsibilities with regards to handling of the data.

Data Security Breaches and Reporting

The Parties are under a strict obligation to notify any potential or actual losses of the Shared Personal Data to the other Party as soon as possible and, in any event, within 24 hours of identification of any potential or actual loss to enable the relevant Parties to consider what action is required in order to resolve the issue in accordance with the applicable national data protection laws and guidance.

Each Party will provide reasonable assistance as is necessary to each other to facilitate the handling of any Data Security Breach in an expeditious and compliant manner.

Data Retention and Deletion

The Parties shall not retain or process Shared Personal Data for longer than is necessary to carry out the agreed purposes. Personal Data will be retained in accordance with any applicable statutory or professional retention periods

The Parties shall ensure that any Shared Personal Data is destroyed in the following circumstances:

1. on termination of the Agreement for whatever reason;
2. on expiry of the Agreement (unless extended under the terms of this Agreement);
3. once processing of the Personal Data is no longer necessary for the purposes it was originally shared.

Warranties

Each Party warrants and undertakes that it will:

1. Process the shared Personal Data in compliance with all applicable laws, enactments, regulations, orders, standards and other similar instruments that apply to its personal data processing operations.
2. Respond within a reasonable time and as far as reasonably possible to enquiries from the relevant Data Protection Authority in relation to the shared Personal Data.
3. Respond to Subject Access Requests in accordance with the terms of this Agreement and in accordance with the requirements of the GDPR.
4. Take all appropriate steps to ensure compliance with the security measures set out in this Agreement.

Indemnity

Each Party to this Agreement will indemnify the other against all liabilities, losses, damages, costs or expenses (including but not limited to any direct, indirect or consequential losses, loss of profit, loss of reputation and all interest, penalties and legal costs (calculated on a full indemnity basis) and all other reasonable professional costs and expenses) suffered or incurred by the other Party arising out of or in connection with any claim made against it in relation to any breach of the GDPR or obligations under this Agreement.

Assignment

This Agreement is personal to the Parties and neither Party shall assign, transfer, mortgage, charge, subcontract, declare a trust over or deal in any other manner with any of its rights and obligations under this Agreement.

Changes to the Applicable Law

In case the applicable data protection and ancillary laws change in a way that the Agreement is no longer adequate for the purpose of governing lawful data sharing, the Parties agree that they will negotiate in good faith to review the Agreement in light of the new legislation.

Entire Agreement

This Agreement constitutes the entire agreement between the Parties in relation to its subject matter, namely the sharing of Personal Data, and supersedes all previous agreements, assurances, warranties, representations and understandings between them, whether written or oral, relating to the subject matter.

Resolution of Disputes

In the event of a dispute or claim brought by a Data Subject or the Data Protection

Authority concerning the processing of the shared Personal Data against either or both Parties, the Parties will inform each other about any such disputes or claims and will cooperate with a view to settling them amicably in a timely fashion.

In respect of breaches relating to this Agreement, each Party shall abide by a decision of a competent court or of any binding decision of the relevant Data Protection Authority.

Governing Law and Jurisdiction

This Agreement and any dispute or claim arising out of or in connection with it or its subject

matter shall be governed by and construed in accordance with the law of England. Each Party irrevocably agrees that the courts of England shall have exclusive jurisdiction to settle any dispute or claim arising out of or in connection with this Agreement or its subject matter.

Variation and Termination

Any proposed changes by a Party to this Agreement, to the purposes of the information sharing, the nature or type of information shared or manner in which the information is to be processed and any other suggested changes to the terms of this Agreement must be notified immediately to the other Party so that the impact of the proposed changes can be assessed.

Either Party may terminate this Agreement at any time upon giving the other Party two months’ notice in writing of its intention to do so.

Approval

**For and on behalf of MoreLife (UK) Ltd**

Sign

Dr

Rita Esen

Data Protection Officer

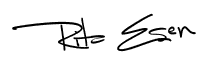
21

/

05

/202

1



Print Name

Position

Date

For and on behalf of LBU

**Project Lead** Dorothy Monekosso

Dorothy

M

onekosso

Director of Research

07/06/2021



Sign

Print Name

Position

Date

Research Team

1. Dr Jackie Campbell 2. Dr Anna Palczewska 3. Dr Gopal Jamnal 4. Vishwanath Reddy Aenugu (Student) 5. Ayotomide Pelumi Adeyeye (Student) 6. Osayande Bright Omobude (Student) 7. Alan Paulson (Student) 8. Ali Alyaqubi (Student)

**ANNEX 1- Data Sharing Details**

The table below provides specific details of the data sharing.

|  |  |
| --- | --- |
| **Description** | **Details** |
| Subject matter of the data sharing: | The processing of this data is to investigate the key determinants of attendance and outcomes of MoreLife services. |
| Duration of the data sharing: | The project will start in June 2021 **for two years** (ending in June 2023) |
| Legal basis for data sharing: | Art. 6(1)(e) - The processing is **necessary for the performance of a task carried out in the public interest** or in the exercise of official authority vested in the controller. |
| Purpose of data sharing: | Research:  The data will be used for MSc thesis by Leeds Beckett postgraduate students. Outputs will include:   * MSc thesis * Internal reports to MoreLife * Conference presentations * Academic peer reviewed publications |
| Types of Personal Data to be shared: | Personal category:   * Age (years) * Gender * Ethnicity * Pre weight * Post weight * Pre height * Post height * Output area of residential address     Special (sensitive) category:   * Asperger’s Syndrome (binary Y/N) * Autism (binary Y/N) * Dyspraxia (binary Y/N) * ADHD (binary Y/N) * Dyslexia (binary Y/N) * Diabetes (binary Y/N) * Asthma (binary Y/N) * Ethnicity * Pre BMI * Post BMI |
| Process of data sharing  (How will it be done) | The data will be fully pseudonymised by MoreLife before being shared. |
|  | The pseudonymised data will be password-protected and shared via e-mail.    Password details will be sent in a separate e-mail. |
| Categories of Data Subject(s) involved: | Customers:  This project will utilise pseudonymised data from individuals who have attended a MoreLife service for secondary data analysis. |
| Disclosing party’s ICO Registration No. | Z2541126 |
| Receiving party’s ICO Registration No: | Z6734933 |
| Disclosing party’s e-mail address: |  |
| Receiving party’s e-mail address: |  |