Allurion

DATA/ML ENGINEER WORK SIMULATION REPORT

KSHITIJ MAHAJAN

Introduction:

The Allurion program is a weight loss initiative designed to help individuals achieve a significant reduction in their total body weight. Success in this program is defined as a minimum of 10% total weight loss (TBWL) after a 4-month period. To ensure the program's effectiveness, it is essential to identify patients who are on track to meet this weight loss standard. This project aims to leverage machine learning to develop a predictive model that analyses patient engagement and weight loss performance, ultimately identifying those who are likely to achieve the desired outcome.

Problem Statement:

The primary challenge we face is to create a machine learning model that can accurately predict whether a patient is on track to meet the 10% TBWL target within a 4-month timeframe. To achieve this, we will utilize a comprehensive dataset consisting of patient demographics, weight-related metrics, and treatment information. The model needs to take into account factors such as age, gender, activity status, clinic ID, treatment type, and health metrics like BMI, body fat, and muscle mass.

Significance:

The significance of this project lies in its potential to enhance the effectiveness of the Allurion program by identifying patients who are likely to succeed in achieving the weight loss target. This predictive model can be a valuable tool for healthcare professionals, allowing them to allocate resources more efficiently, provide targeted support to patients, and ultimately improve the overall success rate of the program. Additionally, a successful implementation of this model could have broader implications for personalized healthcare interventions, paving the way for more tailored treatments based on individual characteristics and progress.

Data Files and Description:

1. users.csv: This file contains demographic information for each patient including unique identifier (UID), Name, Last Name, Gender, Unit, Birthday, Age, Height, Created Date, Activity Status, Clinic ID, Login ID, and a Boolean indicating if the user was sampled from the distribution that reached 10% TBWL at 4 months (program success).

2. weights.csv: This file provides patient-specific information related to weight and other health metrics across time. It lists each patient's Master User ID (matching the UID in the users file), Weight, BMI, Body Fat, Body Water, Bone value, Visceral Fat, Basal Metabolic Rate (BMR), Muscle Mass, along with the date of creation and update for these data, and status flags for active records and deleted records.

3. treatments.csv: This file contains the patient IDs (MasterUserID - again, matching the UID in the users file), the type of treatment they were given (TreatmentTypeID), as well as the start date of the treatment.

Project Execution:

In this section, I'll explain the approach I took to develop the predictive model for identifying patients who are likely to meet the 10% total weight loss (TBWL) standard within a 4-month period.

Data Understanding:

The first step was to get to know the data we had. We looked at information about each patient, such as their age, gender, height, and activity status. We also examined the weight-related details for each patient over time, like BMI, body fat, and muscle mass.

Data Pre-processing: Making the Data Ready

Before we could use the data for our model, we needed to clean it up. This meant dealing with missing information and making sure the data was in a format that the model could understand. I had three different datafiles which stored different kind of information required for our model. Hence, I merged these three datafiles into one file using merging techniques of pandas software.
We found some columns that don't give us useful information for predicting weight loss success. These columns have the same value for everyone or are empty, so they won't help our model. It's like having pieces in a puzzle that don't fit, so we're putting them aside.
We had two columns that seemed similar, one showing when the data was created and the other when it was updated. Since they both had the same values, we only need one of them. It's like having two clocks showing the same time – it's better to keep just one. We found a column that doesn't have any information (it's like a blank space). We can remove this column since it doesn't tell us anything about weight loss.
We looked at some columns with names and IDs. For our prediction, we don't need this specific information, so it's like taking out some details that don't affect the main picture. We found a column that shows the birth date of each person. But we also have another column that shows their age, which is more useful for our prediction. It's like having two different ways to tell how old someone is, and we're using the simpler way.
We had some information about patients that was in categories, like whether they succeeded in losing weight (called "success") and their gender. Computers like numbers, so we turned these categories into numbers. Think of it like assigning a 0 or 1 for "success" (0 means they didn't succeed, 1 means they did), and also assigning

numbers to represent different genders, like 0 for "male" and 1 for "female." This helps the computer work with this information.
The data had dates, which included the time of day (like HH:MM:SS, which stands for hours, minutes, and seconds). However, for our analysis, we only needed the date, not the specific time. So, we just kept the date part and removed the time part. This made the data simpler and easier for our model to use.

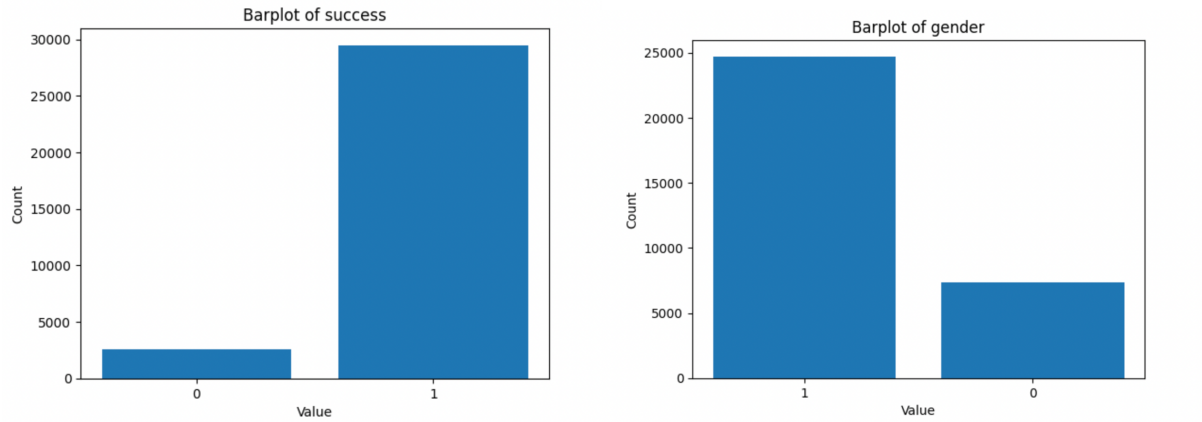
Feature Engineering:

In the feature engineering step, we created new, helpful pieces of information from the existing data. In my effort to create more meaningful information from the data, I encountered a situation where I had two columns: the start date of treatment and the date when a record was created. Initially, these columns might not seem directly useful on their own for predicting success in the Allurion program. However, I found a way to make them much more valuable by combining them. I wanted to understand how many days have gone by from the start of a patient's treatment to the moment when a record was created. To do this, I used a simple math trick – subtraction. I subtracted the start date of treatment from the date when the record was created. This newly calculated value tells us something very useful. It helps us understand the amount of time a patient has been in the Allurion program at the moment a record was created. Think of it as a "time spent in the program" feature. The more time a patient has spent in the program, the more we can learn about their progress, commitment, and the impact of the program on their weight loss journey.
I also included each patient's initial weight for every point in time during their participation in the Allurion program. This helps us track how much weight they've lost from the very beginning, allowing us to better understand their progress relative to their starting point.

Exploratory Data Analysis:

We wanted to make sure that our data didn't have any unusual values that could affect our analysis. We also checked if we had a good balance between the number of patients who succeeded in the program and those who didn't. If the numbers were too imbalanced, it could make our predictions less reliable. We found that for most columns, there were no weird, out-of-the-ordinary values. However, there was one column (BMR) where we noticed some outliers, but they didn't impact our analysis significantly. We looked at how much a specific metric (BMR) is related to the program's success. Surprisingly, we found that BMR wasn't strongly connected to success, so even if we keep the BMR outliers, it won't affect our predictions too much.

We discovered that the number of patients who didn't succeed in the program was much smaller compared to those who did. This kind of imbalance can make our model less effective in predicting both cases, so we need to address this issue in the future to make our model more robust.



Splitting the Dataset:

I took our data and split it into two parts: one part to teach the model (like a teacher showing examples to a student), and the other part to test how well the model learned (like giving the student a quiz).
The thing we want the model to predict is whether a patient will succeed in the Allurion program. I call this our "target" (like the bullseye in a target), and the other information we have about the patients, like their age, weight, etc., are the "independent" variables (like the arrows aimed at the target).
I noticed that most of the patients in our test dataset actually succeeded in the program. Because of this, if our model simply predicts that every patient will succeed, it would be right about 92.30% of the time, just because so many patients in the test dataset passed.

Machine Learning Model Selection and Development:

Machine learning model should be able to identify patients who are on track to meet this weight loss standard. In other words, model should be able to classify patients who are on track to meet this weight loss standard from the patients who are not on track. I needed to select a machine learning model that could learn from the data and make predictions. Think of this like teaching a computer to recognize patterns. I spent time researching and experimenting with different models, considering factors such as the model's interpretability and performance in predicting patient success. For each model, I taught the model using the training data we had. I showed it examples of patients who succeeded in losing 10% of their weight and those who didn't. The model learned

from these examples and tried to figure out what factors were important in making a good prediction.

- **Logistic Regression:** I began with a basic model called "logistic regression." However, because our dataset is large and has a lot of differences, this simple model didn't give us the best results. It struggled to handle the differences between patients, and that's not what we need for this task.
- **Ensemble Models:** Our goal is to have a model that's really good at predicting, even if it's a bit more complex. We also don't need the model to be super easy to understand for now. So, we decided to use what we call "ensemble models." These are like teamwork models, where we combine the predictions of multiple models to get a better result.
  I used three special types of ensemble models:
  - **Random Forest**: This is like a big group of trees working together. Each tree makes its own prediction, and the group combines those predictions to make a better final guess. It's like asking a bunch of friends for their opinion and making a decision based on what they say.
  - **XGBoost**: Think of this as a powerful learner that improves itself over time. It starts with a basic understanding and then gets better and better by learning from its mistakes. It's like a student who studies hard and gets better with each test.
  - **Stacking Ensemble**: This is a cool way of combining the best ideas from different models. It's like having a team of experts who each have their own skills. We let them all give their opinion, and we use their combined wisdom to make our prediction.

Model Evaluation and Results:

In this step, I tested each of our models using a separate set of data to see how well they can predict the success of patients in the Allurion program. I used several metrics, like accuracy, F1 score, precision, recall, ROC AUC score, false positives, and false negatives, to measure how good the models are at making predictions.

Model	Accuracy	F1 score	Precision	Recall	ROC AUC Score	False Positives	False Negatives	Important features
Logistic Regression	94.35%	97%	95.10%	98.97%	68.93%	377	76	Bone, start weight, weight
Random Forest	99.80%	99.89%	99.78%	100%	98.70%	16	0	Start weight
XGBoost	99.85%	99.91%	99.83%	100%	99.02%	12	0	Start weight,weight,bone
Stacked model	99.88%	99.93%	99.87%	100%	99.27%	9	0	Start weight,weight,bone

I looked at how each model performed, and we noticed that two models, Random Forest and XGBoost, did better than the simple logistic regression. These models handled the data's complexity more effectively and gave us better results. The Stacking model performed better than every other model I tested. It had the best scores on the metrics I used to measure success. Here's something really cool – our Stacking model

had zero false negatives and zero false positives. This means it didn't make any major mistakes in its predictions. It was great at identifying patients who were truly on track in the Allurion program, and it didn't mistakenly label anyone as succeeding when they weren't or vice versa.

Conclusion and Recommendations:

In a nutshell, I used different tools to test our models, and I found that Random Forest and XGBoost were better than the simpler logistic regression. However, the real superstar was our Stacking model, which had the highest performance and didn't make any big mistakes. This means our Stacking model can be very useful in predicting how well patients are doing in the Allurion program, helping healthcare professionals provide better support and assistance to the patients.
Additionally, I found that a few specific factors play a crucial role in determining a patient's success in the program. These factors include the initial weight of the patient, their current weight, and a value related to bone health. These three variables are particularly important in predicting success. While other variables also contribute to the prediction, they are not as critical as these three.
Therefore, I suggest focusing on these key factors, especially the patient's initial weight, current weight, and bone value, when assessing a patient's progress. This targeted approach can help healthcare professionals allocate resources more effectively and tailor support to individual patients, ultimately improving the overall success rate of the Allurion program.

Time Allocation:

Below table shows my time distribution among different tasks.

Task	Time spent(Hours
Reading the work simulation document	0.25
Finalizing the problem statement	0.25
Deciding on action plan	0.5
Data preprocessing	0.25
Feature engineering	0.25
Exploratory data analysis	0.25
Machine Learning model selection and development	1.5
Evaluating the results	0.5
Report writing	3
Total time	6.75

I spent 6.75 hours on this project.

Future Priorities:

Fine-tuning the Stacked Model: Although the Stacked model performed exceptionally well, there's always room for improvement. I would spend time fine-tuning the parameters of the individual models within the Stacked model, optimizing their performance even further.

Make Dataset Balanced: In our original dataset, we noticed that there were more samples from one category (e.g., patients who didn't succeed) than the other (e.g., patients who did succeed). This imbalance can lead the model to be biased towards the majority class. Balancing the dataset means adjusting the number of samples from each category to give the model a fairer view of both outcomes. It would certainly be a valuable step to improve the model's performance and overall fairness

Feature Refinement: Continuously refining the features used in the model is crucial. I would explore more advanced feature engineering techniques, look for potential interactions between features, and experiment with domain-specific features that might influence a patient's success in the program.

Model Explainability: While our focus was on predictive power, it's important to enhance the interpretability of the model. I would invest time in developing techniques to make the model's predictions more transparent and understandable, especially for healthcare professionals using the model.

External Validation: Validating the model's performance on completely new, unseen data is essential. I would gather additional data from other time periods or different clinics to ensure that the model's effectiveness holds up in real-world scenarios.

Feedback Loop: Establishing a feedback loop with healthcare professionals and incorporating their insights could significantly improve the model's practicality. This iterative process of collaboration can lead to a more effective model tailored to the specific needs of the Allurion program.

Key Learnings:

Practical Application: I learned how to apply machine learning techniques to solve real-world problems in the healthcare domain. This project reinforced the importance of developing practical and actionable solutions that can have a positive impact on patient outcomes.

Ethical Considerations: The healthcare domain demands a heightened awareness of ethical considerations. Ensuring fairness, transparency, and privacy protection in the model's predictions is paramount, and I experienced how these aspects are vital in a healthcare context.

Effective Communication: Collaboration with non-technical stakeholders, such as healthcare professionals, emphasized the significance of effective communication. I learned how to translate complex machine learning concepts into understandable insights, ensuring that our findings could be easily communicated and acted upon.

Project Management and Time Allocation: The project's time constraints taught me the value of efficient project management. Balancing different phases, allocating time for data exploration, model development, and evaluation, while ensuring quality, is a skill I've refined through this experience.

Practical Impact: The ultimate goal of machine learning is to have a real impact on solving problems. I experienced the satisfaction of creating a model that can genuinely help patients and healthcare professionals. This reaffirmed the significance of applying technical skills to make a positive difference in the world.