National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computing

CS 471 Machine Learning

Fall 2024

Assignment 1

Logistic Regression

Predicting Patient Dropout from a Long-Term Health Treatment Program

Announcement Date: 3rd Oct 2024

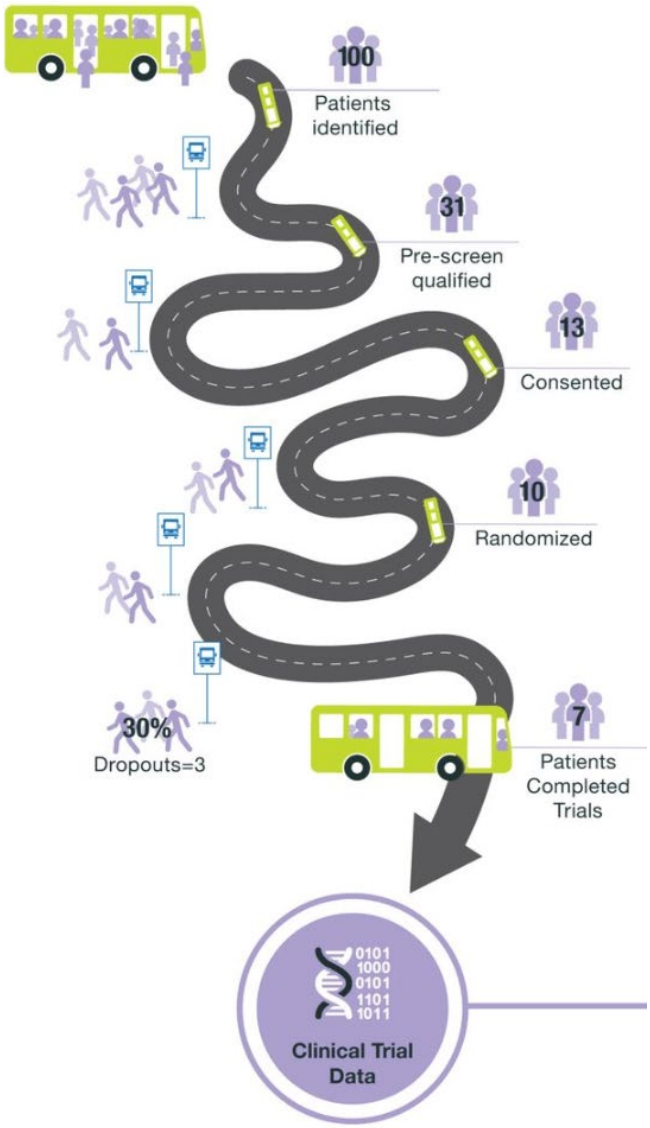
Due Date: 14th Oct 2024 at 11:59 PM (on LMS)

Instructor: Prof. Dr. Muhammad Moazam Fraz

Table of Contents

1. Introduction

You are tasked with predicting patient dropout from long-term health treatment programs. The goal of this assignment is to apply logistic regression, feature scaling, multicollinearity handling, and model evaluation techniques to solve a real-world problem. By the end of the assignment, you will understand logistic regression and how to handle challenges of feature scaling, multicollinearity, and imbalanced data.



1.1 Problem Statement

As a data scientist working with a healthcare provider, your task is to help predict whether patients will drop out of a long-term treatment program based on their interaction with various stages of the program. The dataset provided includes anonymized information about patients' activities, goals, progress reviews, and online interactions with the healthcare system. The aim is to build a logistic regression model that accurately predicts whether a certain patient is expected to drop out or not, identify key factors contributing to patient dropout, and evaluate the trained model using appropriate metrics.

1.2 Purpose and Objectives

The healthcare industry increasingly relies on data-driven decision-making to optimize patient care and outcomes. One of the important challenges faced by healthcare providers is the dropout of patients from long-term treatment programs. Dropouts can lead to worse health outcomes for the patients and increased costs for healthcare providers. Understanding the reasons behind dropout and developing systems to predict it can significantly enhance patient retention and improve treatment effectiveness.

The purpose of this assignment is to:

- **Understand** the problem of patient dropout from long-term health programs.
- **Apply** logistic regression to predict dropout based on patient activities and engagement data.
- **Learn** how to handle common machine learning challenges (feature scaling, multicollinearity, class imbalance).
- **Interpret** the model's results to identify key factors that contribute to patient dropout.

By the end of this assignment, you will have a thorough understanding of the logistic regression algorithm, as well as hands-on experience with real-world challenges in predictive modeling for healthcare.

1.3 Assignment Structure

The assignment is divided into six main sections, each designed to help you develop a comprehensive solution to the patient dropout prediction problem:

- **Data Exploration and Preprocessing -** Explore the dataset to understand its **structure**, handle **missing values**, perform **feature scaling**, and prepare the data for modeling.
- **Feature Engineering -** Create interaction terms, **polynomial features** (if needed), and other engineered features to enhance the predictive power of your model.
- **Multicollinearity Handling -** Detect and address **multicollinearity** issues using **Variance Inflation Factor (VIF)** and apply **regularization techniques**.
- **Model Building -** Implement a **logistic regression** model. **Train** the model, **interpret** coefficients, and analyze **which factors** play a significant role in patient dropout.
- **Model Evaluation -** Evaluate the model using different **performance metrics** (accuracy, precision, recall, F1-score, ROC-AUC).
- **Discussion and Conclusion -** Summarize your **key findings**, discuss different factors contributing to patient dropout, **reflect** on the performance of your model, and **explain** what you learnt during this assignment.

2. Data Exploration and Preprocessing

2.1 Handling Missing Data

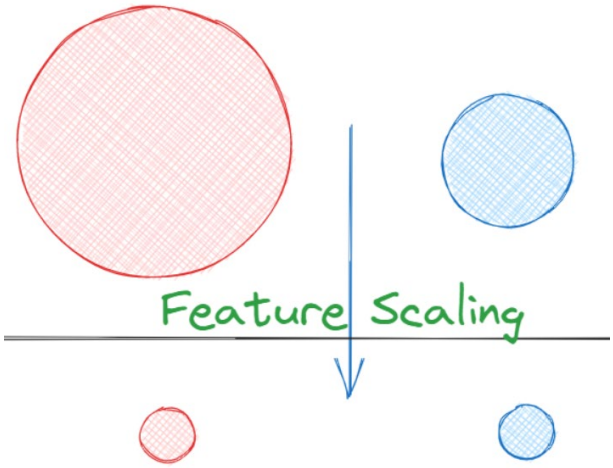
Identify and address any missing values using appropriate techniques. You can try mean imputation or remove rows with excessive missing information. In real-world datasets, missing data is common, and the way you handle it can significantly impact the performance of your model. There are several strategies for dealing with missing data, which depend on the amount and nature of the missing values.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Patient ID	Initial Cor	Number o	Number o	Number o	Number o	Number o	Number o	Number o	Number o	Number o	Patient Se	Treatment	Dropped
2	Session_II	Yes	4	1	0	0	0	0	1	0	0	0	1	
3	Session_II	No	38	0	0	2	0	0	2	0	2	0	1	
4	Session_II	No	8	5	0	0	1	1	1	0	0	1	0	
5	Session_II	No	6	0	0	2	0	0	4	0	0	0	1	
6	Session_II	Yes	31	14	12	1	0	0	4	0	0	2	1	
7	Session_II	Yes	13	6	0	0	0	1	1	0	5	1	1	
8	Session_II	No	14	4	1	0	1	1	1	0	0	1	0	
9	Session_II	No	7	1	0		0	1	1	0	3	0	1	
10	Session_II	No	13	8	1		0	0	1	0	1	1	1	
11	Session_II	No	17	4	1		1	2	2	1	1	1	0	
12	Session_II	No	20	5	10		0	0	1	0	2	2	1	
13	Session_II	No	8		0		0	2	1	0	4	1	1	
14	Session_II	No	9		0		0	1	4	0	1	1	1	
15	Session_II	Yes	8		0		0	1	2	0	1	1	1	
16	Session_II	No	3		0		0	1	2	0	0	1	1	
17	Session_II	No	31		0		0	7	13	0	3	1	1	
18	Session_II	Yes	6		1	0	0	0	1	0	1	1	1	
19	Session_II	Yes	7		0	2	0	1	3	0	0	1	1	
20	Session_II	No	35		8	6	0	2	1	1	9	2	1	
21	Session_II	No	8		0	0	0	0	2	0	3	1	1	
22	Session_II	Yes	6	3	0	0	0	0	1	0	1	1	1	
23	Session_II	No	35	14	0	3	1	3	3	0	9	1	1	
24	Session_II	No	18	11	1	2	0	0	4	0	5	1	1	
25	Session_II	No	6	2	0	1	0	1	0	0	2	0	1	

Figure 2: *Missing values in the patient dropout dataset.*

2.2 Feature Scaling

Since logistic regression is sensitive to the scale of input features, apply feature scaling (standardization or normalization) to make sure that all numeric features are on the same scale. Logistic regression models are sensitive to the scale of input features. Features with larger magnitudes can disproportionately influence the model's performance. To prevent this, you need to scale your features so that they are on a similar scale.

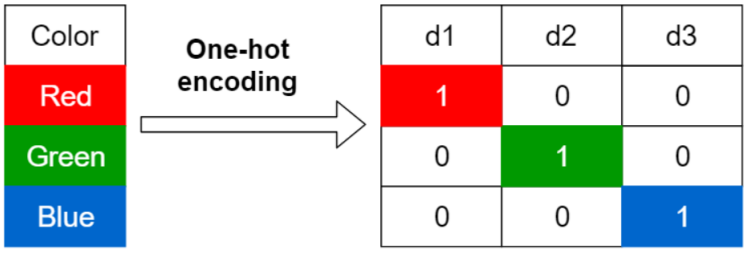


2.3 Encoding Categorical Variables

Convert categorical variables into numerical representations using one-hot encoding, as logistic regression requires numerical input. Machine learning models, including logistic regression, require numerical input. Therefore, any categorical variables in your dataset must be converted into a numerical format. You will need to encode such features into a format that the model can understand.

Types of Categorical Encoding

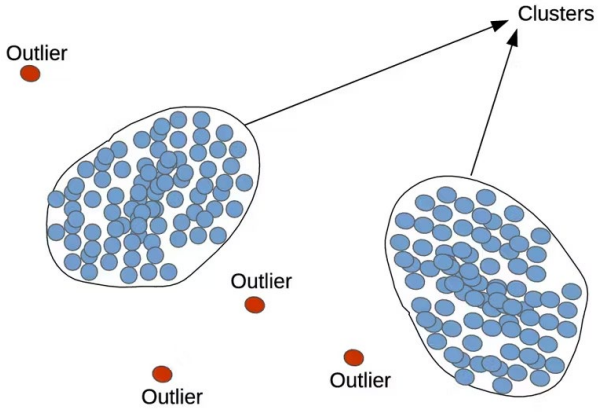
One-Hot Encoding - This method creates a new binary column for each category of the feature. If the feature has k unique categories, k new columns will be created. Each column will have a value of 1 if the category is present in the observation, and 0 otherwise. One-hot encoding is suitable for nominal (unordered) categories.



Ordinal Encoding - If the categorical feature has an inherent order (low, medium, high), ordinal encoding can be used. This method assigns a unique integer value to each category according to its order. However, be cautious with ordinal encoding, as it can introduce bias if the categories do not have a clear ranking.

2.4 Outlier Detection

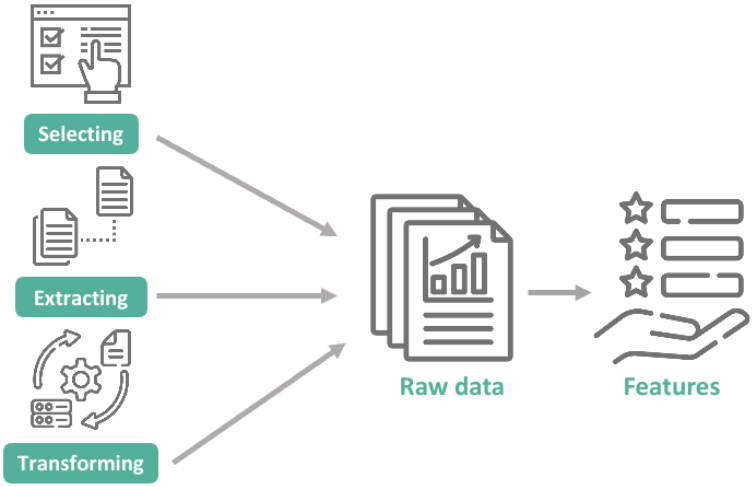
Outliers are extreme values that differ significantly from the majority of the data. They can skew the results of your logistic regression model, particularly in terms of coefficient estimation. In the context of this dataset, outliers could represent patients who exhibit highly unusual behavior in their treatment engagement. There are several ways to identify outliers in the dataset including but not limited to **Boxplots**, **Z-Score Method**, and **Interquartile Range (IQR)**.



3. Feature Engineering

Feature engineering is an important process in machine learning where raw data is transformed into meaningful features that better represent the underlying patterns in the data. Well-engineered features can significantly improve model performance, especially when working with relatively simple models like logistic regression. In this assignment, feature engineering will focus on

creating new features, transforming existing ones, and identifying interactions that can provide more useful information to the model.



Creating New Features - Sometimes, the features available in a dataset do not fully capture the underlying patterns in the data. You may derive new features based on domain knowledge or logical combinations of existing features.

Feature Transformations - Transformations are used to modify existing features to make them more useful for the model. Logistic regression assumes a linear relationship between the features and the log-odds of the target variable. If your features have highly skewed distributions or nonlinear relationships with the target, feature transformations can help make them more linear or normally distributed.

Interaction Features - Interaction features are created by combining two or more features to capture the relationships between them. These features allow the model to understand how different aspects of patient behavior interact with each other, potentially leading to dropout.

4. Multicollinearity Handling

Multicollinearity occurs when two or more predictor variables in a regression model are highly correlated, meaning they contain redundant information. This can negatively impact the model's performance by inflating the variance of the coefficient estimates, leading to unstable predictions and reduced interpretability. In logistic regression, where we are particularly interested in understanding the relationship between features and the log-odds of the target variable, multicollinearity can obscure these relationships and lead to misleading conclusions.

4.1 Identifying Multicollinearity

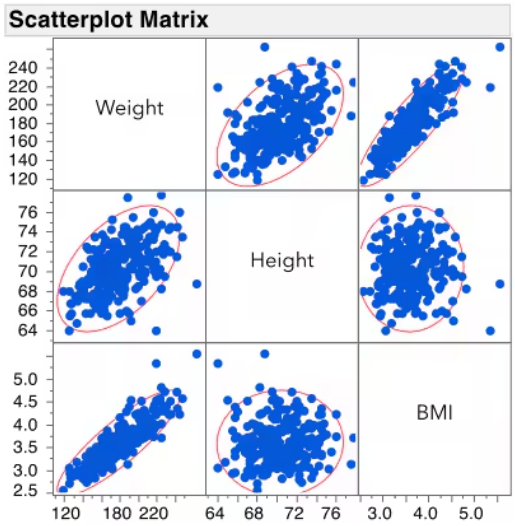
A common first step in detecting multicollinearity is to examine the **correlation matrix** of predictor variables. The correlation matrix provides a pairwise comparison of all features, showing the degree of linear relationship between them. If two features are highly correlated (typically with a correlation coefficient above 0.8 or 0.9), they are likely to introduce multicollinearity into the model. Another widely used method for detecting multicollinearity is the **Variance Inflation Factor (VIF)**. VIF

measures how much the variance of a regression coefficient is inflated due to multicollinearity with other predictors. A higher VIF indicates a higher degree of collinearity.

Multivariate

Correlations

	Weight	Height	BMI
Weight	1.0000	0.5129	0.8668
Height	0.5129	1.0000	0.0220
BMI	0.8668	0.0220	1.0000



4.2 Addressing Multicollinearity

Once multicollinearity has been identified using the correlation matrix or VIF, the next step is to address it in a way that minimizes information loss while improving model stability.

The simplest and often most effective way to handle multicollinearity is to **remove one of the correlated features**. When two or more features are highly correlated, they carry redundant information, and keeping both can cause issues in logistic regression by distorting coefficient estimates. On the other hand, if multiple features are highly collinear but carry valuable information that you don't want to lose, **Principal Component Analysis (PCA)** can be a powerful tool to reduce dimensionality. PCA transforms the original set of features into a smaller set of uncorrelated components (principal components) while retaining most of the variance in the data.

5. Model Building

5.1 Feature Selection

After completing data exploration, preprocessing, and addressing multicollinearity, the next step is to select the final set of features for your logistic regression model. The quality of the features has a significant impact on the model's predictions. Some features may be more relevant based on their clinical or operational importance. **Recursive Feature Elimination (RFE)** ranks features by recursively fitting the model and removing the least important features until the optimal subset is selected.

5.2 Train-Test Split

To evaluate the model's ability to generalize to unseen data, the dataset should be split into training and testing sets. The training set is used to train the logistic regression model, while the testing set is used to assess its performance.

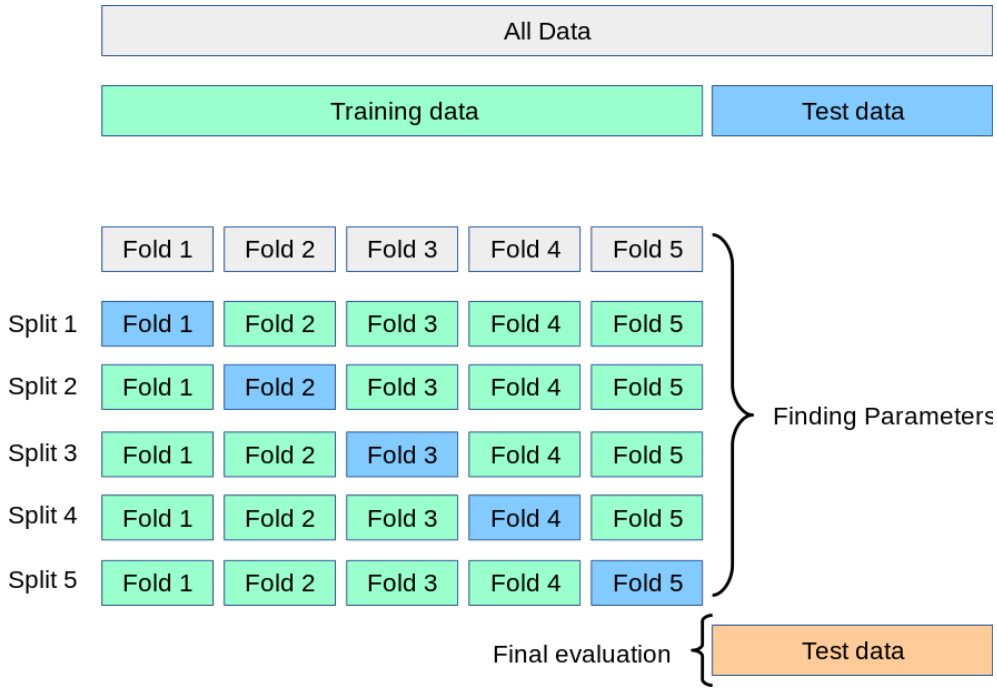
5.3 Model Fitting

Once the data has been split, you can fit the logistic regression model to the training data. Logistic regression is a statistical method used to model binary outcomes, which is perfect for this task of predicting whether a patient will drop out or not. You can explore different regularization techniques like L1 (Lasso) or L2 (Ridge) regularization to prevent overfitting. Regularization helps to avoid overfitting by penalizing large coefficients. Try experimenting with different values of the regularization parameter C. A lower value of C imposes a stronger regularization, while a higher value imposes weaker regularization.

Regularization Tip: Start with L2 regularization (penalty='l2'), as it generally works well with logistic regression and helps mitigate multicollinearity as well.

5.4 K-Fold Cross Validation

K-Fold Cross Validation is a powerful technique used to evaluate the performance and generalizability of machine learning models. Instead of splitting the dataset into just one training and testing set, K-fold cross-validation divides the data into **multiple folds** (or subsets), providing a more robust estimate of performance. This method helps avoid overfitting, ensures the model generalizes well to unseen data, and makes better use of the data by training and testing on different portions of it. **Randomly partition** the dataset into **K equally sized folds**. First, train the model on **K-1** folds and test it on the **remaining fold**. Repeat this process **K times**, with each fold used exactly once for testing. The overall model performance is averaged over the K iterations, giving a more accurate estimate of its ability to generalize.



The choice of K in K-fold cross-validation affects the balance between **bias** and **variance**. A lower K value can lead to higher variance (due to fewer data points in the training set), while a higher K value may lead to lower variance but increased computational cost. **Manage this tradeoff accordingly.**

5.5 Model Evaluation

Evaluating the performance of the logistic regression model is important to make sure that it is making accurate predictions. Logistic regression models can be evaluated using several metrics, but in this case, since the target variable (patient dropout) is binary, common evaluation metrics include **accuracy**, **precision**, **recall**, **F1-score**, and the Area Under the Receiver Operating Characteristic **(ROC-AUC) curve**.

6. Submission Guidelines

Prepare a comprehensive report that includes the following:

- Submit a well-documented **Python notebook** with comments explaining each step of your analysis. Use markdown cells to provide context and explanations.
- Summarize your findings, model, details of the training process, and conclusions inside the notebook. Focus on key points, challenges, results, and discussion.
- Include the final **cleaned**, **processed**, and **transformed dataset** with your submission which you used for model training.
- Any challenges you encountered during the entire process and how you overcame them.

Deliverables

- Detailed code along with markdown explanations, visualizations, results, discussion, key findings, and challenges. **(Jupyter / Python Notebook) (code.ipynb)**
- Cleaned, processed, and transformed dataset used for model training. **(Updated CSV File) (updated_data.csv)**

Compress both deliverables into a zip file. Name the zip file as **<YourName_Your-CMS-ID.zip> e.g. Adnan-Khan_456734.zip and Submit on LMS before Due Date. Late submissions are not acceptable.**

(Make sure to follow the naming convention for both deliverables as well as the final zipped file)

Note: All work submitted must be your own. Adherence to academic integrity is mandatory.

7. Grading Rubric

- Data Exploration and Preprocessing - **20 marks**
- Feature Engineering / Multicollinearity Handling - **15 marks**
- Model Building - **25 marks**
- Model Evaluation - **15 marks**
- Presentation of Results, Discussion, Explanations - **25 marks**

8. Dataset Details

Patient ID - Unique identifier for each patient in the program (4284 unique patients)

Features

- Initial Consultation Attended - Indicates if the patient attended the initial consultation session
- Number of Treatment Sessions Attended - The total number of treatment sessions the patient has attended
- Number of Treatment Goals Set - Number of goals the patient has set during the program, such as exercise targets or milestones
- Number of Treatment Goals Revised - Number of goals the patient has revised or dropped throughout the program
- Number of Progress Reviews Attended - Number of progress review sessions attended by the patient
- Number of Times Treatment Plan Confirmed - Number of times the patient has confirmed or recommitted to their treatment plan
- Number of Times Treatment Phase Initiated - (Number of times the patient has started a new phase of the treatment program
- Number of Treatment Options Explored - Number of alternative treatment options the patient has reviewed or explored
- Number of Times Logged Into Health Portal - Number of times the patient has logged into their health or progress tracking portal
- Number of Educational Resources Viewed - Number of educational resources or materials the patient has viewed, such as articles or videos related to their treatment
- Patient Segment Type - Categorizes patients into different segments, such as high-risk, medium-risk, or low-risk of dropout

Target Variable

- Treatment Dropped - Indicates whether the patient dropped out of the treatment program