PREDICTING PATIENT READMISSIONS IN HEALTHCARE

BUS 9430 - Business Analytics Project Management

TEAM: MediForecast Consultants

Rupa Poddar, Sahithi Arnika Modadugu,Sujasna Tamang, Mustafa Ekinci.

RESEARCH QUESTION 1

Predictive modeling for readmission risk assessment: Can we accurately predict patient readmission based on attributes such as medical history, admission details, and treatment plans?

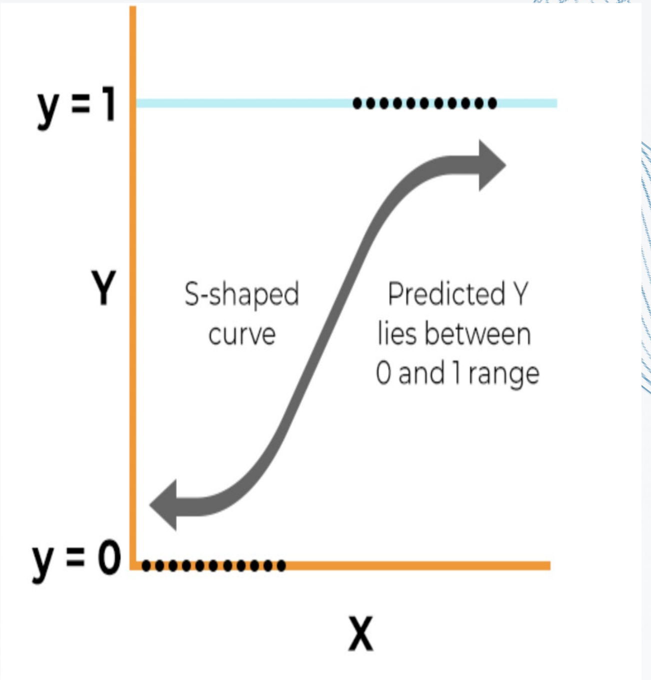
Models

Logistic Regression

Support Vector Machine (SVM)

LOGISTIC REGRESSION

- A statistical method for analyzing datasets with one or more independent variables that determine an outcome.
- The outcome is measured with a binary variable (where there are only two possible outcomes).
- Predicts the probability of the occurrence of an event by fitting data to a logistic curve.
- Uses a logistic function to model a binary dependent variable.
- Commonly used for classification problems in machine learning.



Logistic Regression: Tailored Insights for Patient Readmissions

- Analyze patient data (age, prior admissions, lab results, etc.) to forecast readmission chances.
- The model identifies the strength and nature of the relationship between the likelihood of readmission and various independent variables (predictors). This allows for the identification of risk factors that significantly increase the chances of readmission.
- Result guide healthcare providers in designing personalized follow-up care plans to reduce readmissions.

Pros of Using Logistic Regression:

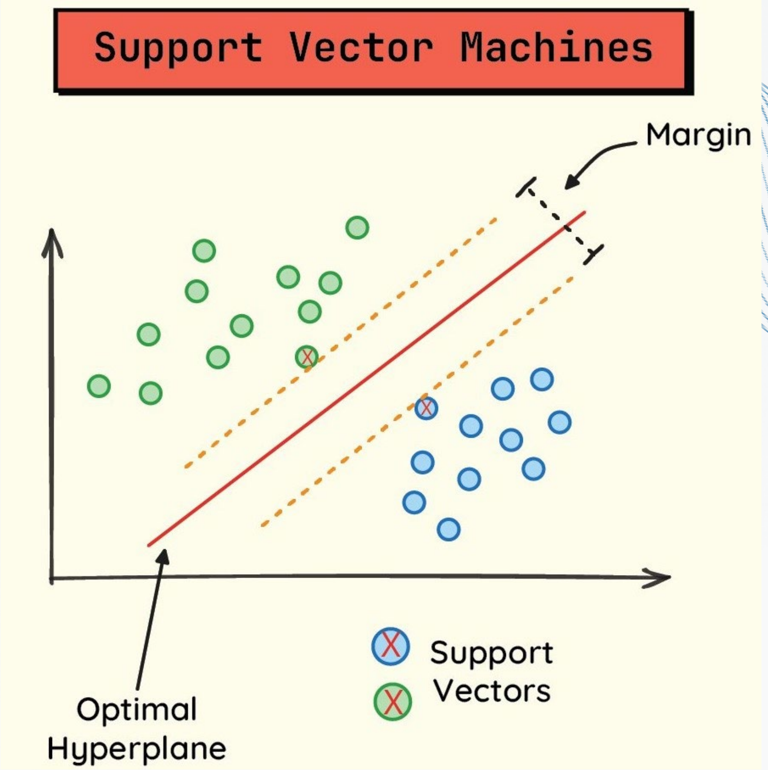
- Suited for binary classification tasks such as predicting yes/no outcomes of patient readmission. Provides transparent and actionable insights, with odds ratios indicating the strength of association between predictors and readmission.
- Straightforward to create, interpret, and deploy within clinical workflows.
- With the right techniques, these models are less likely to overfit compared to more complex models.

Cons of Using Logistic Regression:

- May not effectively capture the complexities of patient health data, which often have non-linear patterns of influence on readmission.
- Doesn't account for time-based variables without additional feature engineering, which can be crucial in readmission where timing of prior admissions is important.

SUPPORT VECTOR MACHINE(SVM)

- The Support Vector Machine (SVM) is a supervised learning algorithm primarily used for classification tasks, especially in scenarios where the data is separable into distinct classes.
- SVM finds the optimal hyperplane that best separates the classes.
- In the case of binary classification, it aims to find the hyperplane that maximizes the margin between the classes, with the margin being the distance between the hyperplane and the nearest data points from each class.



Support Vector Machine: Tailored Insights for Patient Readmissions

- SVMs find the optimal hyperplane (decision boundary) that separates patients who are likely to be readmitted from those who are not.
- Patients with higher predicted readmission probabilities may receive targeted interventions to reduce their risk and prevent readmission.

Pros of Using Support Vector Machine:

- Predicting patient readmission involves classifying patients into two categories: readmitted or not readmitted. SVM is well-suited for binary classification tasks.
- Medical data often exhibits non-linear relationships between attributes such as medical history, admission details, and treatment plans. SVM's ability to handle non-linear relationships makes it suitable for this problem.
- With multiple attributes contributing to readmission risk assessment, the dataset is likely high-dimensional. SVM is known for its effectiveness in high-dimensional spaces.

Cons of Using Support Vector Machine:

- In our healthcare dataset, the classes may be imbalanced, with a significantly higher number of patients not being readmitted compared to those who are readmitted. SVMs may struggle with imbalanced data, leading to biased predictions towards the majority class.

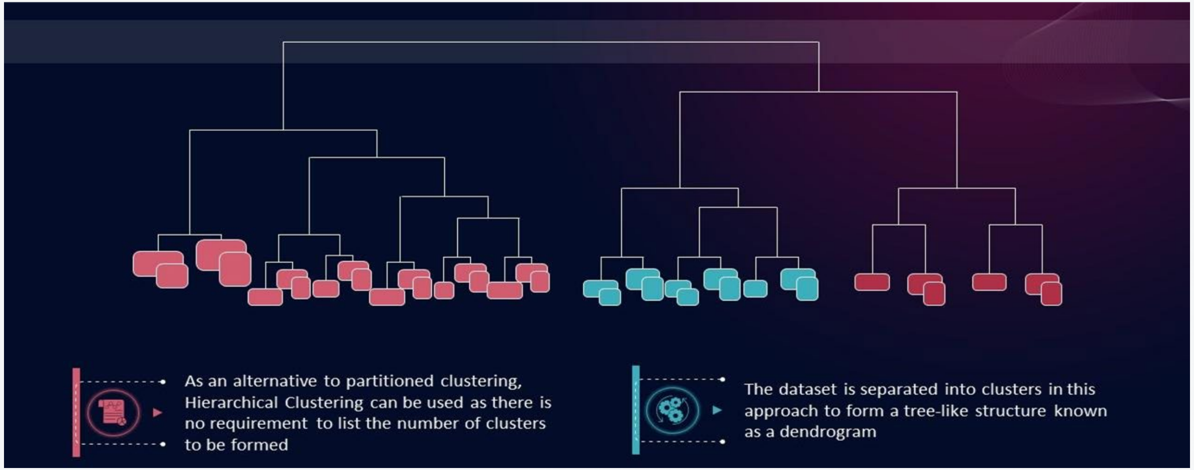
RESEARCH QUESTION 2

Segmentation of patients based on risk factors: Can we segment patients into different risk groups based on their medical profiles and demographic characteristics?

Models

- Hierarchical clustering
- K-Nearest Neighbors (KNN)

HIERARCHICAL CLUSTERING



- Hierarchical clustering organizes data into a tree-like structure by grouping similar data points into clusters.
- It operates continuously, merging or splitting clusters until a hierarchy of clusters is formed, depicting relationships at various level of detail.
- In this method, columns represent data features, while rows represent individual data points or observations.
- Each data point are used to determine similarity or dissimilarity, which guides the clustering process.

Hierarchical Clustering: Tailored Insights for Patient Readmissions

- Hierarchical clustering can help identify distinct patterns or clusters of patients with higher or lower readmission risks.
- It allows us to segment patients based on their medical profiles, demographic characteristics, and healthcare encounters.
- By analyzing the hierarchy of clusters, we can gain insights into the relationships between different patient groups and their readmission risks.

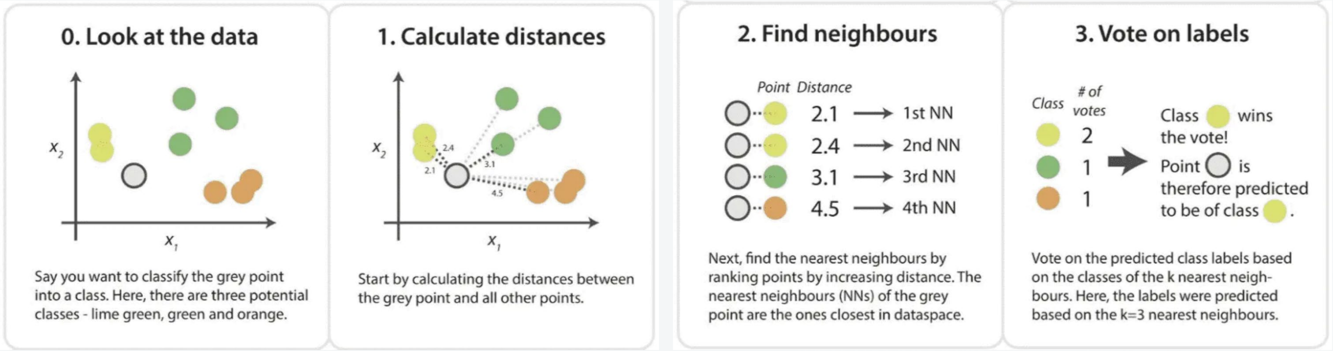
Pros of Using Hierarchical Clustering:

- Supports Mixed Data Types: Can handle mixed data types, such as numerical and categorical variables, facilitating comprehensive patient profiling.
- Incremental Analysis: Enables exploration of data at multiple levels of detail by examining clusters at different heights in the dendrogram.
- Reveals Nested Structures: Captures hierarchical relationships between clusters, identifying subgroups within larger clusters.

Cons of Using Hierarchical Clustering:

- May be computationally intensive, especially with large datasets containing numerous attributes and instances.
- Sensitivity to outliers and noise in the data may affect the clustering results and subsequent analysis.
- Interpretation of the hierarchy of clusters can be complex, requiring careful analysis and domain expertise.

K-Nearest Neighbors (KNN)



- K-Nearest Neighbors (KNN) is a simple, yet powerful, algorithm used for both classification and regression tasks in machine learning.
- It operates by calculating the distance between a query example and the specific examples in the training data, selecting the nearest data points, known as neighbors.
- In KNN, the choice of the parameter 'k' (the number of nearest neighbors to consider) is crucial as it directly influences the performance of the model.
- The distances can be calculated using various methods such as Euclidean, Manhattan, or Hamming distance.

K-Nearest Neighbors: Tailored Insights for Patient Readmissions

- KNN can be employed to predict patient readmissions by analyzing the similarities between patients based on their medical history, demographic factors, and hospitalization details.
- This model is particularly useful for classifying patients into categories such as 'high risk' or 'low risk' for readmissions based on the proximity to other patients within the training set.
- By understanding these patterns, healthcare providers can allocate resources more efficiently and implement targeted interventions.

Pros of Using K-Nearest Neighbors:

- Simplicity and Efficiency: KNN is easy to implement and understand, making it an accessible option for many predictive tasks.
- No Model Training Needed: Unlike many other algorithms, KNN does not require model training since it uses the entire dataset for generating predictions, which can speed up the preparation phase.
- Flexibility in Distance Metrics: Can use different types of distance calculations to best suit the nature of the data.

Cons of Using K-Nearest Neighbors:

- High Memory Requirement: Requires storing the entire dataset, which can be memory intensive and inefficient with very large datasets.
- Sensitive to Irrelevant Features: The presence of irrelevant or redundant features can significantly degrade the performance of a KNN model.
- Scalability Issues: As the size of the dataset grows, the prediction speed can become slower, which may not be suitable for time-sensitive tasks.

THANK YOU!