Patient Risk Analysis –
Predicting Health Risks Using Data Science

Batoul Ballout, Rasha Harb, and Razan Doughman

CMPS262 -Data Science

Dr. Maher Abdallah

Spring 2025

March 29, 2025

Introduction

Understanding Our Topic

Healthcare has seen significant advancements with the integration of data science and machine learning, and one of its most impactful applications is patient risk analysis. By examining patient data, we can predict potential health risks, allowing for early intervention and better treatment outcomes.

Why Is This Important?

Every year, millions of people develop chronic illnesses that could have been prevented or managed more effectively if early risk assessment had been available. According to the World Health Organization (WHO), early detection and intervention strategies significantly reduce the burden of chronic diseases, highlighting the necessity of predictive analytics in healthcare. Therefore, by analyzing key factors like age, lifestyle, and medical history, healthcare professionals can identify high-risk individuals and take preventive measures.

Real-World Impact

With the help of machine learning and predictive analytics, hospitals and healthcare institutions can prioritize patient care, optimize resources, and make more accurate diagnoses. This not only improves patient well-being but also helps reduce the overall burden on healthcare systems.

Dataset Description

We used a Kaggle heart disease dataset containing 10,000 patient records and 21 features.
These features include a mix of numerical (e.g., Age, Blood Pressure) and categorical variables (e.g., Smoking, Alcohol Consumption, Gender).

Overview:

- Source: Kaggle heart_disease Dataset
- Size: 10,000 rows × 21 features
- Target Variable: Heart.Disease.Status

Key Insights from Exploration:

- Missing data was primarily in the Alcohol.Consumption column.
- Distribution of heart disease status is slightly imbalanced but usable.

Explanation of the Objective and Machine Learning Used

Our objective is to build a predictive model that accurately classifies whether a patient is at risk of heart disease using their health data.

Machine Learning Algorithms Used:

- **K-Nearest Neighbors (KNN)**: Classifies based on similarity to nearby data points.
- **Random Forest**: Predicts outcomes using multiple decision trees for robust accuracy.
- **Logistic Regression**: Estimates the probability of a class using a logistic function for binary or multi-class classification.
- **XGBoost**: Boosts weak learners iteratively to enhance predictive performance with optimized gradient boosting.

These algorithms were chosen to compare their strengths in handling different aspects such as feature importance, interpretability, and accuracy.

Best Model turns out to be: Logistic Regression

Cleaning Data

Before we could build any machine learning models, we needed to clean the dataset to make sure it was complete, consistent, and ready for analysis.

Data cleaning steps included:

- Replacing empty strings and "None" values with `NA`.
- Visualized missingness using `gg_miss_var()`.
- **Mode imputation** for `Alcohol.Consumption` (categorical).
- Remaining missing values were removed using `na.omit()`.

This ensured the dataset was complete and consistent for ML model training.

Feature Engineering

After cleaning the data, the next step was to prepare and transform the features to make them more suitable for machine learning models.

Transformations applied:

- **Label encoding** of categorical features with mapping stored for interpretation.

- **Standardization** using scale() to bring all numeric features to zero mean and unit variance
  It turns out to be crucial for distance-based models like KNN.

This transformation ensured that all features were on a similar numerical scale, making them more comparable and easier for machine learning models to interpret and process effectively.

Feature Selection

Once the features were prepared, we needed to identify the most relevant ones for our models by selecting those that contribute most to predicting heart disease. To achieve this, we applied both statistical and model-based techniques:

- ***Correlation Matrix:*** We examined the correlation between numerical features. Since no strong multicollinearity was observed, all numerical features were retained.
- ***Logistic Regression Coefficients:*** A logistic regression model was trained on the SMOTE-balanced dataset. Features with statistically significant coefficients (p-value < 0.05) were selected. This allowed us to focus on variables with a meaningful linear relationship to heart disease risk.
- ***Random Forest Feature Importance:*** Using the MeanDecreaseGini metric from the Random Forest model, we identified the top 10 most important features. This ensemble-based method highlights variables that best split the data and reduce classification error.
- ***KNN-Based Feature Selection:*** We enhanced model performance by removing near-zero variance features and applied Recursive Feature Elimination (RFE) to rank features based on their contribution to classification accuracy. This method allowed us to focus on features most beneficial to the KNN model.
- ***XGBoost Feature Importance:*** An XGBoost model was trained on the SMOTE-balanced dataset, and the top 10 most important features were extracted based on their impact on model performance. This helped improve efficiency by focusing on predictors that had the highest influence in decision-making across boosting rounds.

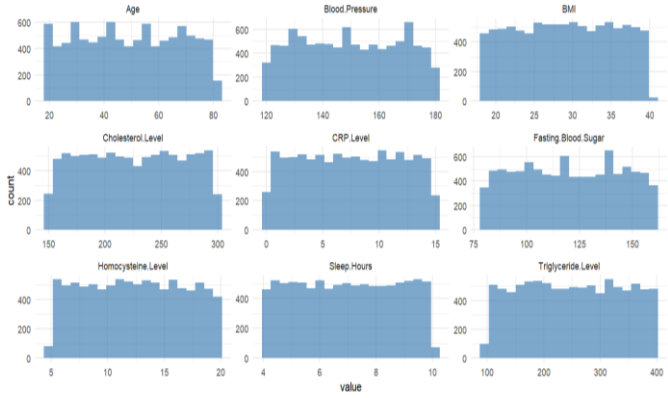
By combining both statistical (logistic regression) and model-based (random forest) approaches, we ensured that only the most predictive and interpretable features were emphasized in subsequent model training.

Exploratory Data Analysis (EDA)

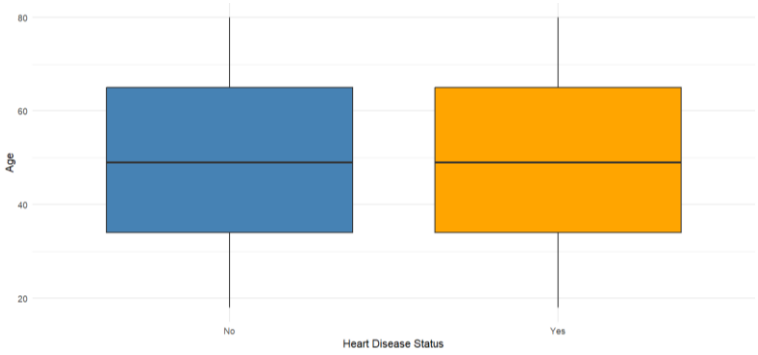
Before building predictive models, we explored the dataset to better understand the distribution of the variables, detect patterns or trends, and uncover any relationships between features and the target variable.

Numeric Features:

- Histograms showed varying distributions for metrics like cholesterol and age.

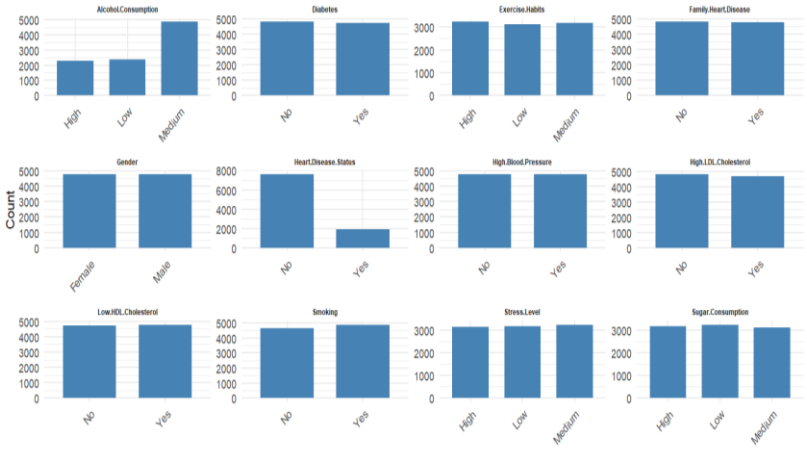


- Boxplot of Age vs. Heart Disease Status revealed older individuals are more likely to have heart disease.

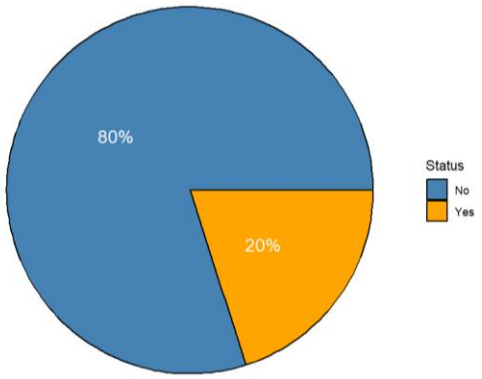


Categorical Features:

- Bar plots highlighted class imbalance in features like alcohol consumption.



- Pie chart showed around 20% of patients had heart disease.



Relationship Insights:

- Grouped bar plots showed significant differences in mean cholesterol, heart rate, and blood pressure between heart disease groups.

These insights helped shape our expectations and model design.

Applying ML Models

K-Nearest Neighbors (KNN):

- Used k=5
- Accuracy: **~54%**
- The model exhibited relatively low performance. Its sensitivity to feature scaling and irrelevant variables may have contributed to its weaker results despite SMOTE balancing.

Logistic Regression (with significant features):

- Trained using only statistically significant features (p-value < 0.05).
- Accuracy: **~80%**
- Strong performance, likely due to its simplicity and the exclusion of noisy features. Suitable for interpreting key health predictors.

Random Forest (with top 10 features):

- Used ensemble learning with 100 trees and top-ranked features based on MeanDecreaseGini.
- Accuracy: **~77%**
- Robust performance; handles feature interactions well. Slightly lower accuracy than Logistic Regression but still effective.

XGBoost:

- Gradient-boosted decision trees trained on all features with SMOTE-balanced data.
- Accuracy: **~78%**
- Achieved high accuracy and balanced performance. Efficient in handling complex relationships and imbalanced data.

Compare Performance

To evaluate the effectiveness of our models, we compared four machine learning algorithms, **K-Nearest Neighbors (KNN)**, **Random Forest**, **Logistic Regression**, and **XGBoost**, based on multiple performance metrics including **accuracy**, **precision**, **recall**, **F1 score**, and **AUC**.

- **KNN** achieved moderate recall (0.45) but had low precision (0.205) and an F1 score of 0.28, indicating it correctly identified nearly half of actual heart disease cases but also produced many false positives. Its accuracy (0.54) and AUC (0.506) were barely above random guessing.
- **Random Forest** showed higher accuracy (0.768) and slightly better precision (0.212), but its recall (0.058) and F1 score (0.091) were very low. Although the model performed decently in general classification, it struggled to correctly detect actual heart disease cases—likely due to class imbalance.
- **XGBoost** had the highest accuracy (0.78), but this was misleading. It performed poorly in recall (0.06) and F1 score (0.09), catching only a tiny fraction of actual positive cases. Like Random Forest, it was affected by class imbalance and prioritized majority class predictions.
- **Logistic Regression** had high accuracy (0.80), but it completely failed to detect heart disease cases (recall = 0). Since it made no positive predictions, precision and F1 score were undefined. This result highlights a serious limitation in using Logistic Regression on an imbalanced dataset without adequate balancing.

Which Model Performed Better?

Among the four models, **KNN performed the best in terms of recall**, correctly identifying **45% of actual heart disease cases**, which is critical in medical applications where missing a positive case can have serious consequences.

- Although its overall accuracy (0.54) and AUC (0.506) were low, **KNN** outperformed the others in its ability to **capture true positives**, making it more suitable for early risk detection.
- In contrast, models like **Logistic Regression and XGBoost**, despite their high accuracy (~0.80), failed to detect most or all positive cases.
- **Random Forest** also achieved a high accuracy (0.768) and slightly better precision (0.212), but its recall (0.058) was too low to be practical in a healthcare setting.
- Therefore, **KNN is considered the most appropriate model** for this problem because it better fulfills the goal of identifying patients at risk of heart disease, even at the cost of a higher false positive rate.

Conclusion

- ***Comprehensive Preprocessing:***
  We cleaned the dataset by handling missing values, applied label encoding to convert categorical data, and used standardization to scale numerical features for better model performance.

- ***Effective Feature Selection:***
  Important features were selected using statistical significance (Logistic Regression) and model-based importance (Random Forest), which helped reduce dimensionality and improve efficiency.

- ***Model Evaluation and Comparison:***
  Multiple machine learning models—KNN, Logistic Regression, XGBoost, and Random Forest—were trained and evaluated using accuracy, precision, recall, F1 score, and AUC. This allowed us to compare their strengths and identify the most effective model.

- ***Limitations and Challenges:***
  We faced several challenges, including missing data, potential encoding bias, interpretability issues with complex models, and high computational cost for advanced algorithms.

- ***Future Enhancements:***
  Future work could include applying deep learning models, integrating real-time monitoring data, using external datasets for validation, advanced resampling, and fine-tuning hyperparameters to improve overall accuracy and generalizability.