HEART ATTACK DISEASE PREDICTION USING MACHINE LEARNING

Artificial Intelligence Class Group C

DIEGO ACOSTA, KEVEN QUEVEDO, ANURIKA OSUJI
UNIVERSITY OF EUROPE FOR APPLIED SCIENCES
Potsdam

Problem Statement

Heart disease remains one of the leading causes of death globally, with heart attacks being one of the most critical manifestations. Early prediction and prevention are crucial to reducing mortality rates and improving patient outcomes. Traditional risk assessments often rely on clinical judgment or simplified scoring systems, which may not capture the full complexity of the patient's health profile. In this context, leveraging machine learning offers an opportunity to model intricate patterns and interactions within medical data, potentially leading to more accurate and timely predictions.

This project aims to develop a machine learning model capable of predicting a patient’s risk of experiencing a heart attack based on a range of lifestyle, biometric, and medical history features. We used a publicly available dataset that includes demographic information, health metrics like blood pressure and cholesterol levels, and lifestyle choices such as smoking, exercise habits, and obesity.

Our primary focus was to implement and evaluate a logistic regression model, given its interpretability and common use in medical research. However, we recognized the limitations of logistic regression when applied to non-linear datasets. This realization prompted us to explore more robust alternatives such as Random Forest and XGBoost for comparison.

Ultimately, our goal was not only to create a predictive model with reasonable accuracy but also to understand which features most strongly contribute to heart attack risk. The findings from this analysis could offer valuable insights for preventive healthcare strategies and further research in clinical decision support systems.

Roles and Responsibilities

Diego Acosta Cantu

- Designed and implemented the entire codebase in multiple Kaggle notebooks.
- Responsible for all aspects of the machine learning pipeline: data cleaning, feature engineering, preprocessing, model training, hyperparameter tuning, evaluation, and visualization.
- Conducted in-depth research on Logistic Regression and explored advanced techniques like feature selection, SMOTE balancing, polynomial interaction terms, and model regularization.
- Developed multiple variants of models (Logistic Regression, Random Forest, XGBoost) for comparative analysis.
- Ensured reproducibility, performance, and alignment with ethical and technical standards.

Keven Quevedo Contreras

- Led the written report preparation.
- Compiled and organized technical content provided by Diego.
- Ensured clarity, formatting, and alignment with academic and project documentation standards.
- Structured the report according to the specified steps (What, Why, How, Result).

Anurika Osuji

- Designed and delivered the oral presentation.
- Created visual materials including slides, confusion matrices, and performance graphs.

Methodology

Step 1: Dataset Exploration

What: Inspected feature distributions, correlations, and missing values.

Why: To assess data quality and identify features that were irrelevant or weakly correlated with the target variable.

How: - Used .describe() and .info() to understand value ranges. - Visualized correlation matrix and value counts for target imbalance. - Discovered moderate class imbalance (64% negative, 36% positive).

Result: Identified redundant features such as ‘Patient ID’, ‘Country’, ‘Hemisphere’, ‘Diet’, ‘Income’ and others, which were removed.

Step 2: Data Preprocessing Strategy

What: Handled categorical encoding, feature scaling, and missing value removal.

Why: Machine learning models require numerical inputs and normalized scales for optimal performance.

How: - Dropped nulls instead of imputation due to small fraction of missing rows. - Encoded ‘Sex’ using OneHotEncoder and scaled numerical features with StandardScaler. - Binary features passed through without transformation.

Result: Produced a clean dataset with consistent types and distributions. Final shape: (8763, 17)

Step 3: Train-Test Split

What: Stratified split into 80% training and 20% test sets.

Why: Preserved class distribution across splits to ensure unbiased evaluation.

How: Used train_test_split() with stratify=y and random_state=42.

Result: Confirmed proportional class distribution in training and test sets.

Step 4: Logistic Regression Model Setup

What: Implemented Logistic Regression with L1 regularization.

Why: Logistic Regression is interpretable and was the required model. L1 helps with feature selection in high-dimensional settings.

How: - Used LogisticRegression(penalty='l1', solver='liblinear', class_weight='balanced') - Applied PolynomialFeatures (degree=2) to capture interaction effects. - Incorporated SelectFromModel with Random Forest to retain only important features.

Result: Built a robust pipeline balancing interpretability and complexity.

Step 5: Model Training Process

What: Fit model on training data.

Why: To learn the parameters that minimize classification error on heart attack risk.

How: - Applied SMOTE to address class imbalance. - Used pipeline.fit(X_train, y_train) on original training data.

Result: Model trained without convergence warnings, but performance was limited due to data non-linearity.

Step 6: Prediction Generation

What: Predicted heart attack risk on test set.

Why: To evaluate generalization performance on unseen data.

How: - Called pipeline.predict(X_test) - Compared predicted values to ground truth y_test

Result: Accuracy: **0.5368**, F1 Score: **0.4484**

Step 7: Performance Evaluation Implementation

What: Measured accuracy, precision, recall, F1 score, and confusion matrix.

Why: To understand both overall performance and class-specific strengths/weaknesses.

How: - Used classification_report() and confusion_matrix() - Visualized with ConfusionMatrixDisplay

Result: - Class 0 (no risk): Precision 0.67, Recall 0.54 - Class 1 (risk): Precision 0.39, Recall 0.53 - Macro Avg F1: **0.52**, Accuracy: **0.5368**

Step 8: Results Interpretation

What: Analyzed model coefficients, confusion matrix, and feature importance.

Why: To explain why the model performs the way it does.

How: - Coefficients showed BMI, Previous Heart Problems, and Cholesterol as strong signals. - Confusion matrix revealed many false positives and false negatives.

Result: Logistic Regression struggled to separate classes cleanly, likely due to the dataset's non-linear relationships.

Extended Evaluation: Alternative Models

Random Forest

- Trained using same SMOTE-balanced and preprocessed data.
- No need for explicit scaling or encoding of binary variables.
- Achieved accuracy: 0.6058

Why: Handles non-linear patterns and feature interactions better than logistic regression.

XGBoost

- Applied RandomizedSearchCV for hyperparameter tuning.
- Focused on maximizing F1 score.
- Best Accuracy: 0.5824, F1 Score: 0.2591

Why: Gradient boosting tends to outperform other models in tabular data but still struggled with minority class recall.

Final Discussion and Conclusion

Feature Importance

- Random Forest and Logistic Regression both highlighted BMI, Cholesterol, Heart Rate, and Exercise Hours as influential.

Model Limitations

- Logistic Regression assumes linear separability, which doesn't hold for this dataset.
- SMOTE improved recall for minority class but introduced noise.
- Confusion matrices showed significant misclassification.

Comparison Summary

Model	Accuracy	F1 Score
Logistic Regression	0.5368	0.4484
Random Forest	**0.6058**	~0.50+
XGBoost	0.5824	0.2591

Real-World Applications

- While models can guide early diagnosis, they must be used alongside medical expertise.
- Even a 60%+ accurate model could help flag high-risk patients for further testing.

Conclusion

This project gave us a valuable opportunity to apply machine learning techniques to a real-world medical problem, predicting heart attack risk. While our main focus was on Logistic Regression, we quickly realized the dataset posed significant challenges due to its complexity, non-linearity, and class imbalance. Despite careful preprocessing, feature selection, and hyperparameter tuning, the logistic model struggled to achieve high performance.

To better understand the problem and push for stronger results, we explored alternative models like Random Forest and XGBoost. These models performed better in some areas, but overall, the task remained difficult, highlighting the complexities of medical data.

Even so, this was a rich learning experience. We gained deeper insights into model limitations, evaluation techniques, and real-world data handling. The process helped sharpen both our technical skills and critical thinking.

We thank our professor for the guidance and support throughout the semester, it made this journey both rewarding and educational.