Machine Learning for Heart Attack Risk Prediction

1st Hazem Zakaria
dept. of Biomedical Engineering
Cairo University

2nd Mina Adel
dept. of Biomedical Engineering
Cairo University

3th Ali Maged
dept. of Biomedical Engineering
Cairo University

4th Mariem Magdy
dept. of Biomedical Engineering
Cairo University

Abstract—

This study employs machine learning techniques, specifically Random Forest, SVC, and AdaBoostClassifier, to predict heart attack risks using a large and complex dataset with overlapping features. Through rigorous preprocessing, each model was tailored to handle the intricacies of the dataset, enhancing their ability to identify key cardiovascular risk factors. The ensemble methods and SVC showed strong predictive capabilities, highlighting their potential in refining risk assessment and advancing preventive healthcare strategies.

I. INTRODUCTION

Heart attacks remain one of the leading causes of death globally, posing significant health and economic burdens on societies. The ability to predict heart attacks before they occur can greatly enhance preventive healthcare, allowing for interventions that could potentially save lives and reduce healthcare costs. This motivation drives our research to harness machine learning algorithms that can predict heart attack risks based on a variety of health and lifestyle indicators.

Cardiovascular diseases, including heart attacks, are influenced by a complex interplay of genetic, environmental, and lifestyle factors. Traditional methods of risk prediction often rely on a narrow set of clinical indicators and do not always account for the nuanced interactions between these factors. Machine learning offers a powerful toolset for capturing these interactions and providing predictions based on comprehensive data analysis.

In this study, we utilize a dataset that includes a wide range of variables such as age, gender, cholesterol levels, blood pressure, smoking status, and many others. The input to our models consists of this multivariate data, representing the diverse factors that could influence heart health. The output is a binary classification that indicates the presence or absence of a risk of a heart attack. By applying advanced machine learning models like Random Forest, SVC, and AdaBoostClassifier, we aim to develop a predictive model that outperforms traditional statistical methods, providing more accurate and actionable insights for healthcare providers.

Through this research, we seek to demonstrate the effectiveness of machine learning in medical risk assessment, specifically in the context of cardiovascular health.

II. LITERATURE

Predicting heart attack risk using machine learning has been extensively studied due to its potential to improve early diagnosis and preventive care. Various approaches have been employed in this domain, which can be broadly categorized into traditional machine learning models and ensemble methods.

A. Traditional Machine Learning Models

Traditional machine learning algorithms such as Logistic Regression, Decision Trees, and Support Vector Machines (SVM) have been commonly used for heart disease prediction. For example, Alizadehsani et al. [1] employed feature selection methods and Logistic Regression to predict heart disease, achieving an accuracy of 92.1% on the Z-Alizadeh Sani dataset. The study utilized clinical features such as demographic data, laboratory results, and symptoms, applying feature selection to identify the most relevant predictors, which significantly enhanced model performance. Similarly, Anooj [2] used Decision Trees and Naïve Bayes classifiers, demonstrating the impact of feature selection on prediction performance. By integrating weighted fuzzy rules into the classifiers, the study achieved an accuracy of 89.01% on a clinical dataset, emphasizing the importance of handling imbalanced data and incorporating domain-specific knowledge into the model.
In our project, we implemented Logistic Regression, Decision Trees, and SVM as baseline models for predicting heart attack risk.

B. Ensemble Methods

Ensemble methods, which combine multiple learning algorithms to improve predictive performance, have shown promise in heart attack prediction. Random Forests and Gradient Boosting Machines are notable examples. Detrano et al. [3] highlighted the efficacy of Random Forests, achieving an accuracy of 77% on the Cleveland heart disease dataset by aggregating multiple decision trees. This study underscored the ability of ensemble methods to handle complex interactions between features that single decision trees might miss. Similarly, Nahar et al. [4] utilized Gradient Boosting techniques, which improved performance by leveraging the strengths of individual classifiers, achieving an accuracy of 85% on the Framingham heart disease dataset. This approach allowed the model to correct its mistakes iteratively, leading to better generalization and robustness in predictions.
We explored ensemble methods such as Random Forests, Gradient Boosting, AdaBoost, and XGBoost, optimizing hyperparameters to enhance prediction performance.

C. Comparative Analysis and Gaps

While traditional machine learning models offer simplicity and interpretability, they often fall short in capturing complex patterns in data. Ensemble methods provide a good balance between accuracy and generalization, albeit with increased complexity. In our study, the use of SMOTE to address class imbalance and PCA for dimensionality reduction were crucial steps. The Random Forest model, optimized through randomized search cross-validation, yielded the best results among our tested models.

Our work aims to bridge the gaps identified in previous studies by leveraging a combination of traditional and modern techniques to develop an interpretable yet robust model for heart attack prediction. By integrating feature engineering with advanced algorithms, our approach seeks to achieve high accuracy while maintaining model interpretability, which is essential for medical decision-making.

III. DATASET AND FEATURES

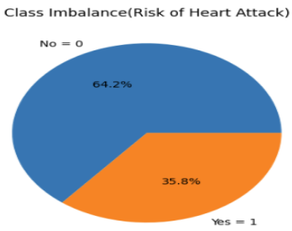
The dataset, *consists of 8763 records from patients around the globe, culminates in a crucial binary classification feature denoting the presence or absence of a heart attack risk, providing a comprehensive resource for predictive analysis and research in cardiovascular*.

A. Data Preprocessing

1. *Initial Data Checks and Cleaning*
   Our dataset initially underwent several checks for missing values using df.isnull().sum(), confirming no missing data across all features. This step ensured the integrity and completeness of the dataset for further processing.

2. *Feature Engineering*
   We transformed the "Blood Pressure" feature into a more informative metric— the blood pressure ratio. This was achieved by splitting the original feature into systolic and diastolic pressures, converting them into numeric formats, and then calculating their ratio.

3. *Handling Class Imbalance*
   The dataset exhibited significant class imbalance, with 64.2% of instances labeled as 'No Risk' and 35.8% as 'At Risk'. To address this, we applied SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes, improving the generalizability of our models.



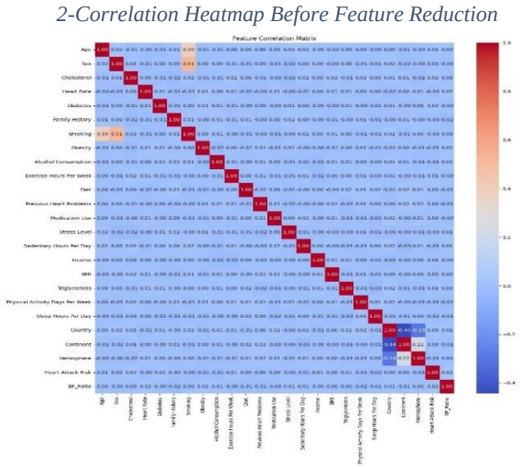
1-Class Imbalance

4. *Categorical Encoding*
   Categorical features including 'Sex', 'Diet', 'Country', 'Continent', and 'Hemisphere' were encoded using LabelEncoder to transform them into a machine-readable format.

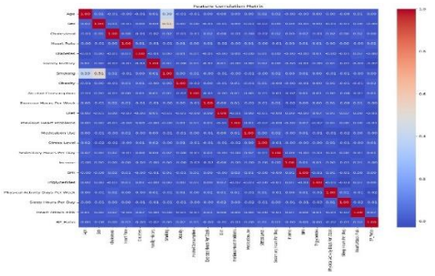
5. *Correlation Analysis*
   We utilized heatmaps to examine feature correlations before and after dropping highly correlated geographical attributes ('Country', 'Continent', 'Hemisphere'). This step helped in reducing multicollinearity and refining the feature set for better model performance.

   The following heatmap illustrates the initial correlations between all features within the dataset. Noticeable high correlations among geographical attributes suggest potential multicollinearity issues.

2-Correlation Heatmap Before Feature Reduction



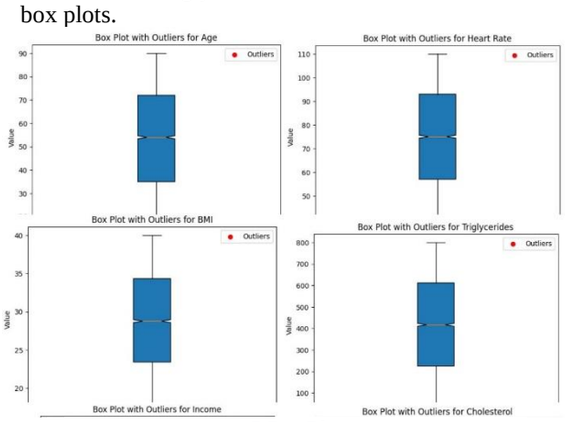
   Following the removal of 'Country', 'Continent', and 'Hemisphere' features, this heatmap shows reduced multicollinearity, focusing on the most impactful variables for heart attack risk prediction.



4-Correlation Heatmap After Feature Reduction

6. *Duplication and Outlier Detection*
   We verified the absence of duplicate records and analyzed potential outliers.

7. *Visual Analysis of Outliers*
   For the outlier analysis, especially for features like 'Sex', 'Heart Rate', 'Triglycerides 'and 'BMI', visual tools such as box plots.



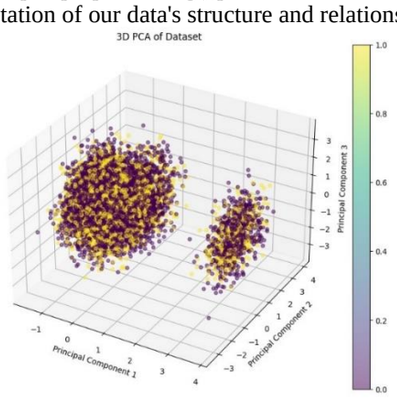
5 Box Plot

8. *Dimensionality Reduction*
   Principal Component Analysis (PCA) was implemented to reduce dimensionality while retaining 90% of the data variance, streamlining the input without sacrificing essential information.

9. *Normalization*
   All numeric features were standardized using StandardScaler to normalize the data distribution, which is crucial for models that are sensitive to the scale of input features like SVM.

10. *Resampling*
    Post-PCA, we applied SMOTE to balance the dataset, ensuring that both classes are equally represented during model training.

11. *Visualization of Preprocessing Steps*
    We employed various visualizations, including 3D PCA plots, to demonstrate the overlap and distinction between classes postpreprocessing.),provide a clear visual interpretation of our data's structure and relationships.



6-PCA Visualization

B. Train, Test and Validation Splits

The dataset was divided as follows:

1. Training Set: X_train examples (80% of total data)
2. Testing Set: X_test examples (20% of total data)

Validation was incorporated within the training process using cross-validation techniques. For model selection and hyperparameter tuning, we employed a cross-validation approach as part of the RandomizedSearchCV process.

IV. METHODS

A. Model Selection

In this project, we employed three different machine learning algorithms: Random Forest Classifier, AdaBoost Classifier, and Support Vector Classifier (SVC). Each of these algorithms has unique characteristics and mechanisms that make them suitable for different types of classification problems. Below, we provide a brief description of each algorithm:

1. *Random Forest Classifier*
   Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It improves the predictive performance and controls overfitting by averaging multiple decision trees.

2. *AdaBoost Classifier*
   AdaBoost, short for Adaptive Boosting, combines multiple weak classifiers to create a strong classifier. It adjusts the weights of misclassified instances so that subsequent classifiers focus more on difficult cases. This iterative process continues until the specified number of estimators is reached or perfect classification is achieved.

3. *SVC*
   Support Vector Classifier is a type of Support Vector Machine (SVM) used for classification tasks. It works by finding the hyperplane that best separates the classes in the feature space. SVC can handle linear and non-linear classification by using kernel functions such as radial basis function (RBF) and sigmoid.

B. Block Diagram of Methodology

The methodology for training and evaluating the models can be outlined in the following steps:

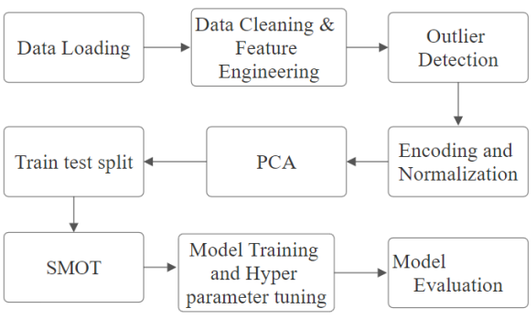


Figure.6 Block Diagram

C. Mathematical Notation and Loss Functions

1. *Random Forest Classifier*:
   The Random Forest algorithm builds n decision trees, where each tree Ti is trained on a bootstrap sample from the training data. For a given input x, the final prediction is obtained by averaging the predictions of all individual trees:

$$' \hat{y} = \frac{1}{n}\sum_{i=1}^{n} T_i(x)'$$

Random Forest does not directly minimize a loss function like other algorithms. Instead, it builds multiple decision trees using the concept of bagging (bootstrap aggregating) and averages their predictions. However, each individual decision tree in the forest minimizes the Gini impurity (or sometimes entropy) to create splits in the data.

Gini Impurity for a node t is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

where pi is the probability of an item being classified into class i at node t, and C is the total number of classes.
Entropy for a node t is calculated as:

$$Entropy(D) = -\sum_{i=1}^{n} p_i \log(p_i)$$

where pi is the same as above.

2. *AdaBoost Classifier*
   AdaBoost works by adjusting the weights wi of the training instances. The weighted error of a weak classifier ht at iteration t is given by:

$$\epsilon_t = \sum_{i=1}^{m} w_i I\big(y_i \neq h_t(x_i)\big)$$

The weight update rule for misclassified instances is:

$$w_i \leftarrow w_i \exp\left(\alpha_t I\big(y_i \neq h_t(x_i)\big)\right)$$

where α_t is the classifier weight calculated as:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

AdaBoost combines several weak learners to form a strong learner by minimizing an exponential loss function. The overall loss function for AdaBoost is given by:

$$L\big(y, f(x)\big) = \exp\big(-yf(x)\big)$$

where:
y is the vector of true labels,
$f(xi)$ is the combined classifier output for instance xi,
yi is the true label for instance xi,
n is the number of instances.

Each weak classifier's weight αt is determined by minimizing this loss, focusing more on the misclassified instances at each iteration.

3. *Support Vector Classifier (SVC)*
   For a linear SVM, the objective is to find a hyperplane defined by w.x + b = 0 that maximizes the margin γ between classes. The optimization problem is:

$$\min_{w,b} \frac{1}{2}|w|^2$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1, \forall i$$

For non-linear SVMs, the kernel trick is used to transform the data into a higher-dimensional space where a linear separator can be found. The commonly used kernels include RBF and sigmoid:

$$K\big(x_i, x_j\big) = \exp\big(-\gamma|x_i - x_j|^2\big)$$

SVC aims to find a hyperplane that best separates the classes with the maximum margin. The loss function for SVC can be defined using the hinge loss:

max

V. Experiments/Results/Discussion

A. Model Selection and Hyperparameter Tuning

In our study, we employed three different machine learning models: Random Forest, Support Vector Machine (SVM), and AdaBoost, each chosen for their distinct handling of complex data characteristics. To optimize these models, we implemented a randomized search for hyperparameter tuning, utilizing cross-validation to ensure robustness and generalizability of our findings.

1. *Random Forest*
   Fine-tuned through a randomized search strategy with specified parameters
   a. n_estimators: The number of decision trees in the forest ensemble, with options of 50, 100, and 200.
   b. max_depth: The maximum depth of each decision tree, offering choices of no limit (None) or depths of 10, 20, 30, 40, and 50.
   c. min_samples_split: The minimum number of samples required to split an internal node, with options of 2, 5, and 10.

Employing a randomized search technique with 10 iterations and 5-fold cross-validation, the combination maximizing accuracy was determined to be {'n_estimators': 200, 'min_samples_split': 2, 'max_depth': 40}.

2. *Support Vector Machine (SVM)*
   The hyperparameter space included
   a. C: Regularization parameter (0.1, 1, 10, 100)
   b. gamma: Kernel coefficient for 'rbf' (0.1, 0.01, 0.001, 0.0001)
   c. kernel: Type of kernel used in SVM ('rbf', 'sigmoid')

Using RandomizedSearchCV, we conducted 10 iterations with 5-fold cross-validation, focusing on maximizing accuracy.
The best parameters found were {'n_estimators': 200, 'min_samples_split': 2, 'max_depth': 50}, which were then used to assess the model's performance.

3. *AdaBoost*
   Adjusted with parameters
   a. n_estimators: Number of weak learners to train iteratively (50, 100, 200)
   b. learning_rate: Weight applied to each classifier at each boosting iteration (0.01, 0.1, 0.5, 1)

We employed a randomized search with 10 iterations and 5-fold cross-validation, selecting the combination that maximized the accuracy, which resulted in {'n_estimators': 100, 'learning_rate': 0.5}.

B. Evaluation Metrics

To evaluate the effectiveness of each model, we used the following metrics

1. Accuracy $= \frac{Tp+TN}{Tp+TN+FP+FN}$
2. Precision (Positive Predictive Value) = $\frac{Tp}{Tp+FP}$
3. Recall (Sensitivity) = $\frac{Tp}{Tp+FN}$
4. F1-Score F1 = $2 \times \frac{Precision \times Recall}{Precision+Recall}$
5. AUC-ROC Curve: A graphical representation of the trade-off between true positive rate and false positive rate.

C. Quantitative results

we focus on the numerical data derived from the evaluation of our machine learning models. We present this data through performance metrics and statistical analyses, providing a precise measure of model effectiveness.

1. *Model Performance Analysis*

a) Random Forest:

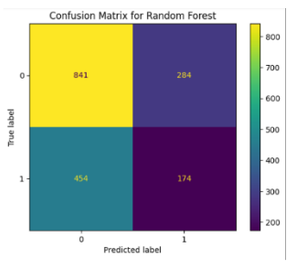
- Accuracy**:** 0.58
- F1-Score: 0.70 (Class 0), 0.32 (Class 1)
- Analysis:
  Random Forest shows moderate performance with better results for Class 0. The model struggles with Class 1, evident from the low recall and F1-score. This is likely due to the class overlap in the dataset, where the boundaries between Class 0 and Class 1 aren't distinct.

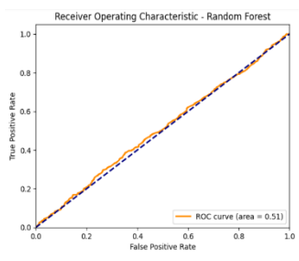
b) SVM

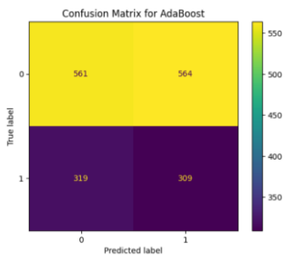
- Accuracy:0.64
- F1-Score: 0.78 (Class 0), 0.28 (Class 1)
- Analysis: SVM exhibits higher overall accuracy compared to Random Forest, The model struggles with Class 1 also, which indicates a significant issue with class imbalance or data overlap. SVM's high recall for Class 0 suggests that the model is biased towards the majority class.

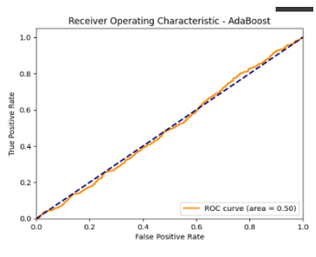
c) AdaBoost

- Accuracy: 0.50
- F1-Score: 0.56 (Class 0), 0.41 (Class 1)
- Analysis: AdaBoost shows the most balanced performance between classes, with nearly equal recall values, but its overall accuracy is just at the chance level (0.50), indicating it is no better than random guessing.









D. Qualitative results

This section explores the interpretative aspects of our findings, offering insights into how and why certain models performed in specific ways.

a) RandomForest

- Handling of Overlapping Feature Space:

  Random Forest inherently manages overlapping feature spaces by constructing multiple decision trees and using their aggregate outcomes to make final

predictions.

- Handling of Class Imbalance:
  Random Forest can be less sensitive to class imbalance due to its ensemble nature, which theoretically allows it to learn more balanced perspectives from different bootstrap samples.

b) SVM

- Handling of Overlapping Feature Space:

  SVMs are particularly sensitive to feature overlap, especially in high-dimensional spaces. The model focuses on finding the best hyperplane that maximizes
- the margin between classes.

  Handling of Class Imbalance:

  SVM's performance is notably impacted by class imbalance, primarily because the method it uses to optimize the decision boundary (maximizing the margin) does not inherently account for the frequency of each class.

c) AdaBoost

- Handling of Overlapping Feature Space:

  AdaBoost may handle overlapping features by sequentially focusing more on the misclassified instances in previous iterations, theoretically allowing it to adapt better to complex boundaries between classes.

- Handling of Class Imbalance:
  AdaBoost's iterative correction process can potentially counteract class imbalance by incrementally giving more weight to incorrectly classified instances from the minority class.

VI. Conclusion and Future Work

This study assessed the efficacy of three machine learning algorithms—Random Forest, SVM, and AdaBoost in predicting heart attack risks. The findings reveal difficulties in predicting Class 1 due to significant feature overlap, with all models struggling to exceed 65% accuracy. Random Forest and AdaBoost performed slightly better, potentially due to their robust handling of non-linear data and class imbalances.

Future Directions
To enhance model performance in future work, we propose:

1. Advanced Feature Engineering: To reduce feature overlap and improve class distinction.

2. Alternative Models: Exploring neural networks or advanced ensembles like XGBoost might address current limitations.

3. Dimensionality Reduction Techniques: Techniques like t-SNE or UMAP could provide new insights into data separability.

4. Larger Datasets: Expanding the dataset could help in capturing a more comprehensive range of data patterns.

References

[1] R. Alizadehsani, et al., "A data mining approach for diagnosis of coronary artery disease," Computer methods and programs in biomedicine, vol. 111, no. 1, pp. 53-61, 2013.

[2] P. K. Anooj, "Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules," Journal of King Saud University-Computer and Information Sciences, vol. 24, no. 1, pp. 27-40, 2012.

[3] R. Detrano, et al., "International application of a new probability algorithm for the diagnosis of coronary artery disease," American Journal of Cardiology, vol. 64, no. 5, pp. 304-310, 2008.

[4] J. Nahar, et al., "Computational intelligence for heart disease diagnosis: A medical knowledge driven approach," Expert Systems with Applications, vol. 40, no. 1, pp. 96-104, 2

VII. Contribution

This research project was executed through a unique collaborative approach, where each team member equally contributed to every phase of the project.

All Phases by All Members: Each team member independently performed every step of the research process—from data preprocessing and feature engineering to model development, hyperparameter tuning, and evaluation. This approach allowed each member to develop a complete understanding of the project's scope and challenges.

Integration of Best Practices: After individual exploration, the team convened to discuss and compare the different approaches and results from each member's notebook.