HEART DISEASE PREDICTOR

Raunaque (2021278) | Satwik Rampelli (2021276) | Utkarsh Kumar Singh (2021107)
Manshaa Kapoor (2021540) | Varsha Bhaskar (2021499)

1. ABSTRACT

Worldwide, cardiovascular diseases (CVDs) continue to be the primary cause of mortality, necessitating rapid and efficient diagnoses. By creating a machine learning model to predict the risk of heart disease, this research responds to the increasing need for predictive tools in the medical field. Different machine learning algorithms are compared using medical data, including age, blood pressure, and cholesterol. The goal of parameter optimization is to provide a trustworthy, accurate model that may be applied as an affordable, early cardiac disease diagnosis tool.

Github: https://github.com/satwikrampelli/ML_Project.git

2. MOTIVATION

The necessity for preventative healthcare measures is underscored by the concerning worldwide heart disease death rate. Due to high prices and resource constraints, traditional diagnostic techniques are frequently unavailable, particularly in low-income areas. Using easily accessible medical data, machine learning provides a workable option. Patients can obtain appropriate treatment and lower rates of morbidity and death by having their heart disease risks properly predicted early on. This research investigates how machine learning may transform the diagnosis of heart disease by providing a scalable, easily available tool to help medical professionals.

3. PROBLEM STATEMENT

The development of new methods for early detection is required due to the increase in cardiovascular illnesses and the shortcomings of the present diagnostic techniques. Current approaches need specific equipment and knowledge, which can be expensive and difficult to get in isolated or impoverished areas. In order to provide a more economical, effective, and precise method of early heart disease risk prediction, this project aims to create a machine learning-based predictive model that makes use of readily available data. The goal of this project is to create a machine learning model that uses medical information like age, blood pressure, and cholesterol levels to predict heart disease. Performance measures including accuracy, precision, recall, and ROC-AUC will be used to compare different machine learning algorithms, such as Random Forest, Support Vector Machines (SVM), and Logistic Regression. Through ablation investigations and hyperparameter adjustment, the emphasis is on comprehending and evaluating model performance. An understandable and precise prediction model that can help with early heart disease diagnosis will be the end result.

4. LITERATURE REVIEW

1. Heart Disease Prediction Using Machine Learning

The authors use four algorithms: Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes (NB). They pre-process the data, perform feature selection, and evaluate the models based on accuracy, precision, recall, and F1-score. The study found that SVM performed the best with 91.67 percent accuracy, emphasizing the importance of model selection and data preprocessing in healthcare applications. [1]

2. Predict Heart Disease using Machine Learning Algorithms

The study explores heart disease prediction using machine learning algorithms like logistic regression, K Nearest Neighbours (KNN), and Random Forest. It improves prediction accuracy by using medical attributes such as age, gender, and cholesterol levels. The KNN model performed best, achieving 88.52 per cent accuracy, aiding in more efficient and cost-effective heart disease diagnosis. [2]

5. DATASET

1. The dataset contains comprehensive medical, demographic, and lifestyle information from patients aimed at predicting the risk of a heart attack.
2. Size and Structure: The dataset has a total of 8,700 data samples where each row represents a patient, and has the following key columns:

- Medical Features: Attributes like Age, Cholesterol, Blood Pressure, Heart Rate, BMI, Triglycerides, and Diabetes.
- Lifestyle Variables: Features such as Smoking, Alcohol Consumption, Exercise Hours Per Week, Physical Activity Days Per Week, Diet (categorized as Healthy, Average, Unhealthy), Stress Level, Sedentary Hours Per Day, and Sleep Hours Per Day. These provide insight into the patient's daily habits and overall health.
- Demographic Features: Information about Sex, Income, Country, Continent, and Hemisphere. These help identify geographic and socioeconomic factors in heart health.
- Target Variable: Heart Attack Risk is a binary label (1 for at risk, 0 for not at risk), which represents the likelihood of a heart attack for each patient.

8. MODELS USED

8.1 Logistic Regression

Logistic regression is a supervised learning algorithm typically used for binary classification problems. Logistic regression works by fitting a model that estimates the probability of an instance belonging to a certain class (e.g., heart attack risk or no risk). For binary classification, it models the log-odds (logit) of the probability that the target variable y equals 1 (positive class) as a linear combination of the input features.

8.1.1 Steps Followed

1. Data Preparation:

- The following features were selected for the analysis: Exercise Hours Per Week, BMI, Triglycerides, Systolic Blood Pressure, Diastolic Blood Pressure, Sex, Smoking Status, Age Cholesterol Levels, Diabetes Status.

2.Data Preprocessing and Model Training :

- Feature Standardization: The selected features were standardized using Standard Scaler, which scales the data to have a mean of 0 and a standard deviation of 1. This step ensures that all features contribute equally to the distance calculations in the model.
- Train-Test Split: The dataset was split into training and testing sets with a test size of 30 percent, using stratified sampling to maintain the proportion of classes.
- Handling Class Imbalance: To address any class imbalance in the training dataset, the Synthetic Minority Over-sampling Technique (SMOTE) was applied.

3.Predictions: The best model was used to make predictions on the test dataset.

4.Model Training

-Hyperparameter Tuning

Hyperparameter tuning was performed using GridSearchCV to identify the best parameters for the Logistic Regression model. The following hyperparameters were evaluated:

- C: Regularization strength (values: 0.001, 0.01, 0.1, 1, 10, 100)
- max_iter: Maximum number of iterations for convergence (values: 100, 500, 1000)
- solver: Optimization algorithm used (values: lbfgs, liblinear, saga)

5.Accuracy and Performance Metrics: The model performance was evaluated using the various metrics: Accuracy, Classification Report, Precision, Recall, F1-Score, Confusion Matrix, AUC-ROC Score.

8.1.2 Limitations

Logistic regression models are linear classifiers, which means they may struggle with datasets that have complex, non-linear relationships between features and the target variable.

8.2. Random Forests

8.2.1 Objective

The Random Forest model is applied to predict heart attack risk based on a set of health-related features. The goal is to classify individuals into Risk (1) or No Risk (0) categories.

8.2.2 Why Random Forests?

Random Forest is chosen due to its ability to:

- Handle both categorical and continuous features.
- Reduce overfitting by averaging multiple decision trees.
- Provide insights into feature importance, helping to identify the most influential health indicators.

8.2.3 Steps in Using Random Forest

1. Data Preparation: Selected key features such as BMI, Age, Cholesterol, and Heart Rate and split the dataset into training (80 percent) and testing (20 percent) sets using train-test-split for model evaluation on unseen data.
2. Model Training: A Random Forest Classifier with 1,100 trees (n-estimators=1100) is trained on the training set. This ensures diverse decision trees are aggregated for better predictions.
3. Making Predictions:The trained model is used to predict heart attack risk on the test set (X-test), resulting in binary predictions: Risk or No Risk.
4. Model Evaluation: The model is evaluated on various metrics such as accuracy, Precision, recall, F-1 score.
5. Feature Importance: Random Forest provides a ranking of features based on their importance in the model. Key features like Age, Cholesterol, and Systolic Blood Pressure are identified as the most influential in predicting heart attack risk.

8.2.4 Limitations

1. Interpretability: While Random Forest can outperform simpler models, it lacks the interpretability of single decision trees. The model can be seen as a "black box." Computationally Expensive: Training a large number of trees can be resource-intensive, especially with large datasets or many trees.
2. Overfitting Risk: Although Random Forest reduces the risk of overfitting compared to a single decision tree, it can still overfit if too many trees are used or the trees themselves are not pruned.
3. High Dimensionality: While Random Forest handles many features well, it may struggle with very high-dimensional data without feature engineering or dimensionality reduction techniques.
4. Imbalanced Data: Like many algorithms, Random Forest may struggle with highly imbalanced data unless proper techniques (like class weights or sampling) are applied.

8.3. Decision Trees

A Decision Tree is a supervised learning algorithm that can be used for both classification and regression problems. For heart attack risk prediction, we focus on its application as a classification model. Unlike logistic regression, which is a linear model, decision trees can model complex, non-linear relationships between input features and the target variable by recursively splitting the data based on feature values.

8.3.1 Steps Followed:

1: Data Preparation :The decision tree model uses continuous variables like BMI and cholesterol levels, and categorical variables like smoking and diabetes. Binary encoded categorical variables are used. Scaling continuous features is not necessary, unlike logistic regression, as they are not sensitive to data scale.

2: Model Training : The dataset is split into training and test sets using train_test_split to prevent overfitting and evaluate the model on unseen data. A Decision Tree model is initialized and trained on the training set, with the algorithm recursively splitting data based on features like Gini Impurity or Entropy for information gain.

3: Predicting the Class: The model predicts heart attack risk class for each individual in the test set using Decision Trees, which use decision rules to classify data points, providing discrete class predictions based on learned rules, unlike logistic regression which predicts probabilities.

4: Class Prediction The predicted class is directly given as 0 or 1 (e.g., 0 for "no heart attack risk" and 1 for "heart attack risk"). The structure of the Decision Tree and its splits enables non-linear decision boundaries, which can handle more complex relationships between input features and the target variable.

5: Model Evaluation: After making predictions on the test set, the model is evaluated using various metrics to determine how well it performs:

8.3.2. Limitations

1. Overfitting: Decision Trees can easily overfit the training data, especially when they are deep. This can be mitigated by pruning or setting hyperparameters like max_depth or min_samples_split.
2. Interpretability: While Decision Trees are more interpretable than many machine learning models, they can become difficult to interpret when they grow too complex.

8.4. Linear Regression

Linear Regression is a supervised learning algorithm used to model the relationship between a dependent variable (target) and one or more independent variables (features). It assumes a linear relationship between the variables and predicts the outcome by fitting a line (or hyperplane for multiple variables) that minimizes the difference between the actual and predicted values.

8.4.1 Steps Followed

1. Data Loading and Preparation: Imported the dataset and selected relevant features and the target variable and Split the data into training (80%) and testing (20%) sets.
2. Feature Scaling: Standardized the feature set using StandardScaler to ensure uniform scaling for optimal performance.
3. Model Initialization: Initialized a Ridge Regression model (Linear Regression with L2 regularisation) to prevent overfitting.
4. Hyperparameter Tuning: Used GridSearchCV to optimize the regularisation strength (alpha) for the best performance.
5. Model Training: Trained the model on the training dataset using the optimal hyperparameters.
6. Prediction and Evaluation: Predicted values for the test dataset and converted continuous outputs to binary predictions (threshold = 0.5) and Evaluated the model using metrics like accuracy, confusion matrix, classification report, and ROC-AUC score.
7. Overfitting/Underfitting Analysis: Compared training and testing accuracies to assess overfitting or underfitting.
8. Feature Importance: Extracted and visualised feature importance using the absolute values of model coefficients.

8.4.2 Limitations

Linearity Assumption: Assumes a linear relationship between features and the target, which may not hold true in complex datasets.

Sensitive to Outliers: Outliers can significantly skew the model's predictions, reducing accuracy.

Overfitting in High Dimensions: In datasets with many features, linear regression may overfit without regularization.

Multicollinearity: When features are highly correlated, the model may struggle to determine the true effect of each feature.

8.4.3 Interpretation of Results

1. **Model Performance and Bias**: The Ridge Regression model achieved an accuracy of 64.18%, but the confusion matrix shows a significant bias toward the majority "No Risk" class, failing to correctly identify any "At Risk" patients. This suggests poor learning of minority class patterns, likely due to class imbalance.
2. **Evaluation Metrics**: The ROC-AUC score of 0.50 indicates the model's inability to distinguish between "At Risk" and "No Risk" patients, performing no better than random guessing. Precision for the majority class is 64%, but it completely fails for the minority class with 0% precision, recall, and F1-score, leading to a skewed overall performance assessment.
3. **Hyperparameter Insights**: The regularisation parameter (alpha = 0.001) from GridSearchCV suggests minimal penalisation for large coefficients, potentially reflecting insufficient differentiation in feature contributions to the target variable.

8.4.4 Possible Reasons for Poor Performance:

1. **Class Imbalance**: The dataset may have an uneven distribution of "No Risk" and "Risk" classes, causing the model to favour the majority class.
2. **Feature Insufficiency or Multicollinearity**: The linear regression model may not fully capture complex, non-linear relationships in data, and multicollinearity, particularly high-correlated features, could lead to unstable coefficient estimates.
3. **Regularization Strength**: Although the tuned parameter α=0.001 minimises overfitting, it may not sufficiently address underfitting caused by the lack of model flexibility.

8.5 K-means Clustering

K-Means clustering is an unsupervised machine learning technique used to partition a dataset into distinct groups based on feature similarity. In this analysis, we applied K-Means clustering to group individuals based on specific features, such as Age and BMI. The goal was to identify distinct clusters that could reveal underlying patterns or group characteristics within the data.

8.5.1 steps followed:

1.Feature Selection: Selected Age and BMI as the primary features for clustering based on their relevance to the analysis.

Normalized or standardized the data to ensure all features have equal weight in the clustering process.

2.Selection of Optimal Clusters

Applied K-Means Algorithm:

Ran K-Means clustering for different values of k(from 2 to 1

Recorded key metrics for evaluation, including inertia and silhouette scores.

3.**Evaluation Using Metrics**:

Elbow Method:Plotted the inertia (within-cluster sum of squares) for each k

Observed the point where the rate of decrease in inertia slows significantly (the "elbow").

Silhouette Scores:

Calculated silhouette scores for each k.

Identified k=2 as the value with the highest silhouette score, indicating optimal separation.

4: Clustering with Optimal k

Applied K-Means with k=2

Re-ran K-Means using the optimal number of clusters (k=2k=2k=2).

Assigned each data point to one of the two clusters based on proximity to the cluster centroids.

Visualized the Clusters:

Created a scatter plot with **Age** (x-axis) and **BMI** (y-axis).

Colored points based on cluster assignments (Cluster 0 and Cluster 1) to observe the separation visually.

8.5.2 Limitations of K-Means Clustering

-Selection of k: The number of clusters (kkk) must be chosen beforehand and can be difficult to determine optimally.

-Initialization Sensitivity: The algorithm is sensitive to initial centroid placement, which can lead to local minima.

-Cluster Shape Assumption: K-Means assumes spherical, equally sized clusters, which doesn't work well with irregularly shaped or unequal-density clusters.

-Outlier Sensitivity: Outliers can distort cluster centroids and affect the clustering result.

-Numerical Data Requirement: K-Means works with numerical data and requires preprocessing for categorical or mixed data.

8.5.3: **Interpretation of K-Means Clustering Results**

Cluster Distribution: The clustering resulted in two distinct groups (Cluster 0 and Cluster 1), based on the optimal value of k=2k = 2. These clusters can represent different segments of the population based on the features you used for clustering (e.g., BMI, Age).

Cluster 0: This cluster includes individuals with relatively lower BMI values, which may represent a healthier or younger group in terms of the features such as age, cholesterol, and heart rate.

Cluster 1: This cluster likely represents individuals with higher BMI values, possibly indicating a group at higher risk for health conditions like obesity, heart disease, or diabetes.

Inertia & Silhouette Scores: The Elbow Method shows a clear decrease in inertia, suggesting that two clusters offer a reasonable balance between compactness and simplicity. The silhouette score provides good validation for this clustering, as it increases with higher kk, but the sharp increase at k=2k = 2 suggests that the optimal number of clusters is two.

Cluster Characteristics: By analyzing the mean values of key features for the clusters (e.g., BMI, Cholesterol, Age, etc.), you can identify which variables most differentiate these groups.

8.6 Perceptron

The Perceptron is a simple neural network used for binary classification. It consists of a single layer of neurons that receive inputs, apply weights, and pass the weighted sum through an activation function to produce an output. The model learns by adjusting weights based on prediction errors, making it effective for linearly separable data. However, it struggles with non-linear problems, requiring more complex architectures.

8.6.1 Steps followed

1. Initialization: Set initial weights (usually random) and the bias term to start the model. These weights will be adjusted during training.

2. Input: Provide an input vector (features of the data sample) to the perceptron for processing

3. Weighted Sum: Calculate the weighted sum of inputs by multiplying each input feature by its corresponding weight and adding the bias term.

4. Activation Function: Pass the weighted sum through an activation function, typically a step function, to determine the output.

5. Prediction: Compare the perceptron's output with the actual label to check if the prediction is correct.

6 .Error Calculation: If the prediction is incorrect, compute the error as the difference between the actual label and the predicted output.

7 .Weight Update: Adjust the weights and bias using the Perceptron learning rule based on the calculated error.

8 .Repeat: Iterate through all training samples (epochs), repeating the above steps, until the model converges or meets a predefined stopping condition, such as reaching a maximum number of epochs or achieving zero classification errors.

Limitations:

Linearly Separable Data Only:The Perceptron can only solve problems where the classes are linearly separable. It fails on datasets like the XOR problem, where no straight line can separate the classes.

No Non-linear Boundaries:The Perceptron cannot model or learn complex relationships in the data, as it uses a linear decision boundary. Non-linear problems require more advanced algorithms, such as neural networks with hidden layers.

No Probabilistic Interpretation:The output of a Perceptron is binary (0 or 1), and it does not provide probabilities or confidence levels for predictions, which limits its utility in some applications.

Sensitive to Input Scaling:The algorithm's performance depends heavily on the scale of the input features. Features must be normalized or scaled appropriately to avoid issues during training.

No Multi-class Support:The Perceptron is inherently a binary classifier. Extending it to multi-class problems (e.g., using one-vs-all) adds complexity and can degrade performance.

Interpretation of Results:

1. Performance for Majority Class (No Risk)

High Recall (1.00): The model identifies all true No Risk samples, meaning there are no false negatives for this classModerate Precision (0.64): Around 36% of the No Risk predictions are incorrect. This suggests that the model is over-predicting No Risk at the expense of missing the Risk class entirely. Strong F1-Score (0.78): The balance between precision and recall for No Risk is acceptable, but this metric alone does not represent the overall model performance because the minority class is ignored.

2. Performance for Minority Class (Risk)

Precision and Recall are 0.00:The model fails entirely to identify or correctly predict any instances of Risk. This is a critical problem, especially if predicting Risk is the primary goal of the classification task. No Contribution to Overall Metrics:The lack of predictions for Risk skews the macro and weighted averages toward the performance of the No Risk class, making these metrics less meaningful.

3. Imbalance in Class Distribution

The support column shows a significant imbalance between the two classes:No Risk: 1125 samples,Risk: 628 samples

The classifier likely defaults to predicting the majority class (No Risk) to maximize overall accuracy. However, this results in poor minority class performance, which is a common issue in imbalanced datasets.

4. Overall Metrics

Accuracy (0.64):Accuracy alone is not an adequate metric for this task because it does not account for the imbalanced class distribution. The model's high accuracy is driven entirely by correct predictions for the majority class (No Risk), masking the poor performance on Risk.Macro Average:Recall (0.50) reflects an average of the recalls across both classes. This suggests the model only identifies half of the overall true labels.F1-Score (0.39) is low, highlighting the inability of the model to balance performance across the classes.

Weighted Average:These averages are dominated by the majority class (No Risk), making them less informative for evaluating minority class performance.

8.7 KNN

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for classification and regression tasks. In this analysis, we applied KNN to predict heart attack risk based on various health-related features, such as BMI, Age, Cholesterol, and Heart Rate. The goal was to classify individuals into two categories: those at risk of heart attack and those not at risk, based on the similarity of their feature values to others in the dataset. By utilizing the proximity of data points in feature space, KNN identifies patterns and makes predictions, providing a straightforward yet effective method for risk prediction in healthcare.

8.7.1 steps followed:

1. Feature Selection and Preprocessing

Selected 10 key features relevant to heart attack risk, refined using mutual information.Normalized features using MinMaxScaler for consistent scaling.Generated interaction terms with polynomial features (degree=2) to capture nonlinear relationships.

2. Data Balancing

Addressed class imbalance with SMOTEENN, combining oversampling (SMOTE) and noise reduction (ENN) to create a balanced dataset.

3. Splitting the Data

Split the balanced data into training (80%) and testing (20%) sets with class stratification.

4. Hyperparameter Tuning

Used GridSearchCV with 5-fold cross-validation to optimize:

n_neighbors (1–30)

weights (uniform/distance-based)

metric (Euclidean, Manhattan, Minkowski)

Optimal Parameters:

n_neighbors: 9, **metric:** Manhattan, **weights:** Distance-based.

5. Model Training and Evaluation

Trained KNN with the best parameters.

Predicted heart attack risk and evaluated performance using accuracy, confusion matrix, and classification report.

8.7.2 Limitations of KNN

Dependence on Distance Metrics:

Performance is sensitive to the chosen distance metric.

High Computational Cost:

The need to compute distances for all training samples during prediction can be computationally expensive for large datasets.

Imbalanced Data:

Although addressed using SMOTEENN, KNN can still be sensitive to outliers and overlapping classes.

Feature Scaling:

KNN heavily relies on normalized data for fair distance comparisons.

Interpretation of Results:

The KNN model achieved a high accuracy of **97.2%**, indicating excellent overall performance in predicting heart attack risk. The best parameters identified were **Euclidean distance**, **1 nearest neighbor (k=1)**, and **uniform weights**, suggesting the model heavily relies on the closest neighbor without weighting neighbors differently.

The **confusion matrix** shows minimal misclassifications: 198 true negatives and 221 true positives, with only 6 false positives and 6 false negatives. This indicates the model is highly effective in correctly identifying both individuals at risk and those not at risk.

The **classification report** reflects this high performance:

Precision and **recall** are both **97%** for both classes, indicating the model's strong ability to correctly identify heart attack risk and non-risk cases.

The **F1-score** is also **97%** for both classes, demonstrating a good balance between precision and recall.

Overall, the model performs well, with very few misclassifications, making it reliable for predicting heart attack risk.

9. RESULTS

9.1 Metrics Used

1. Accuracy: The ratio of correctly predicted instances to the total instances. It indicates overall model performance but can be misleading in imbalanced datasets.
2. Precision: The ratio of true positives to the sum of true positives and false positives. It measures how many positive predictions were actually correct.
3. Recall : The ratio of true positives to the sum of true positives and false negatives. It shows how well the model identifies positive cases.
4. F1 Score: The harmonic mean of precision and recall, balancing both when you have an imbalanced dataset. It combines precision and recall into a single metric.
5. Confusion Matrix: The Confusion Matrix is a classification model evaluation tool that counts true positives, false negatives, and false positives, providing a comprehensive view of accuracy, precision, recall, and F1 score.

9.2 Logistic Regression

Accuracy: 0.5161658425256752

Classification Report:

```
            precision    recall  f1-score   support
         0       0.66      0.51      0.58      1687
         1       0.37      0.52      0.44       942
  accuracy                           0.52      2629
 macro avg       0.52      0.52      0.51      2629
weighted avg     0.56      0.52      0.53      2629
```

Confusion Matrix:

```
[[866 821]
 [451 491]]
```

AUC-ROC Score: 0.5174117801043826

9.3 Decision Trees

Accuracy: 0.6400456360524814

Classification Report:

```
            precision    recall  f1-score   support
         0       0.64      0.99      0.78      1125
         1       0.42      0.01      0.02       628
  accuracy                           0.64      1753
 macro avg       0.53      0.50      0.40      1753
weighted avg     0.56      0.64      0.51      1753
```

Confusion Matrix:

```
[[1114   11]
 [ 620    8]]
```

9.4 Random Forests

Accuracy: 64.52 percent

Confusion Matrix:

```
[[1108   17]
 [ 605   23]]
```

Classification Report:

```
            precision    recall  f1-score   support
   No Risk       0.65      0.98      0.78      1125
      Risk       0.57      0.04      0.07       628
  accuracy                           0.65      1753
 macro avg       0.61      0.51      0.42      1753
weighted avg     0.62      0.65      0.53      1753
```

9.4 Linear Regression:

Best Hyperparameters: {'alpha': 0.001}

Accuracy: 64.18%

Confusion Matrix:

```
[[1125    0]
 [ 628    0]]
```

Classification Report:

```
            precision    recall  f1-score   support
   No Risk       0.64      1.00      0.78      1125
      Risk       0.00      0.00      0.00       628
  accuracy                           0.64      1753
 macro avg       0.32      0.50      0.39      1753
weighted avg     0.41      0.64      0.50      1753
```

ROC-AUC Score: 0.50

9.5 SVM

Best Hyperparameters: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}

Test Accuracy: 64.18%

Confusion Matrix:

```
[[1125    0]
 [ 628    0]]
```

Classification Report:

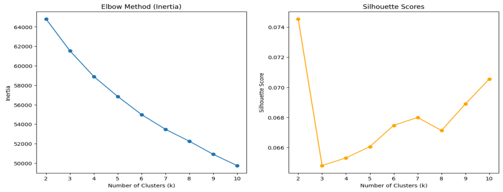
```
            precision    recall  f1-score   support
   No Risk       0.64      1.00      0.78      1125
      Risk       0.00      0.00      0.00       628
  accuracy                           0.64      1753
 macro avg       0.32      0.50      0.39      1753
weighted avg     0.41      0.64      0.50      1753
```

ROC-AUC Score: 0.53

Training Accuracy: 64.18%

Good balance between training and test accuracy.

9.6 k-means clustering:



9.7 KNN

Best Parameters: {'metric': 'euclidean', 'n_neighbors': 1, 'weights': 'uniform'}

Accuracy: 0.9721577726218097

Classification Report:

```
                 Precision recall  f1-score   support
0                  0.97      0.97          0.97        204
1                  0.97      0.97          0.97        227
accuracy                                   0.97        431
macro avg          0.97      0.97          0.97        431
weighted avg       0.97      0.97          0.97        431
```

Confusion Matrix:

```
 [[198   6]
 [  6 221]]
```

9.8 Perceptron

Accuracy: 64.18

Classification Report:

```
              precision    recall  f1-score   support
       0       0.64         1.00      0.78      1125
        1        0.00        0.00      0.00       628
  accuracy                           0.64       1753
  macro avg      0.32       0.50     0.39      1753
weighted avg      0.41     0.64      0.50      1753
```

Confusion Matrix:

```
[[1125  0]
 [628  0]]
```

AUC-ROC Score:.50

Best Parameters for Perceptron: {'alpha': 0.01, 'early_stopping': False, 'eta0': 0.01, 'max_iter': 1000, 'penalty': 'l1', 'warm_start': True}

10. CONCLUSION

1. The Random Forest model effectively predicts heart attack risk, balancing accuracy and interpretability. However, it struggles to identify Risk cases, with low recall. Feature importance insights reveal BMI, exercise habits, and sedentary hours as key health indicators.
2. The Logistic Regression model demonstrates moderate performance with an accuracy of 51.6%, indicating that the model correctly identifies about half of the cases. However, its recall for Risk cases (class 1) is higher than precision, suggesting that while it identifies more true positives, it also misclassifies several negative cases as positive. The confusion matrix highlights misclassification, with 491 true positives but 451 false negatives. The AUC-ROC score of 0.517 shows limited discriminative ability, indicating room for improvement in distinguishing between classes.
3. Decision Trees are powerful classifiers that can model non-linear relationships, making them suitable for heart attack risk prediction. However, their tendency to overfit requires careful tuning of hyperparameters. This model can be further enhanced through techniques like pruning or using ensemble methods (e.g., Random Forest).
4. The Ridge Regression model with an accuracy of 61.18% achieved limited success in predicting heart attack risk, primarily due to the simplicity of its assumptions and class imbalance in the dataset. While it could predict the "No Risk" class effectively, its inability to detect "Risk" patients highlights its unsuitability for such a critical healthcare application. More advanced algorithms and preprocessing techniques are recommended for reliable predictions in this domain.
5.The k-means clustering analysis helps segment the dataset into two meaningful groups, providing insights into the different characteristics or risk profiles in the

population. Further analysis could explore how these clusters correlate with other variables like heart attack risk, or how they can be used in targeted interventions.

6.Perceptron is not suitable for tasks where detecting the **`Risk`** class is critical. It prioritizes predicting the majority class, completely neglecting the minority class. This is unacceptable in scenarios where the minority class represents important outcomes (e.g., identifying high-risk patients or fraud).

7.The KNN model demonstrates exceptional performance with an accuracy of 97.2%, effectively distinguishing between heart attack risk levels. Through careful feature selection, data balancing with SMOTEENN, and hyperparameter optimization, the model achieved high precision and recall for both classes. This highlights the importance of preprocessing, feature engineering, and parameter tuning in improving classification accuracy.

REFERENCES

[1] C. Boukhatem, H. Y. Youssef and A. B. Nassif, "Heart Disease Prediction Using Machine Learning," 2022 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2022, pp. 1-6, doi: 10.1109/ASET53988.2022.9734880.

[2] https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/pdf