PREDICTION OF HEART DISEASE USING KNN, LOGISTIC REGRESSION AND GRADIENT BOOST CLASSIFIER

Divya teja Gaddam

Catholic university of America

gaddam@cua.edu

ABSTRACT:
This project aims in the development of predictive models using KNN, Logistic Regression and Gradient boost classifier which can predict whether a person has heart disease or not based on different predictive variables like Age, Sex, Chest pain, blood pressure etc.

1. INTRODUCTION

Heart disease remains a leading global health concern, contributing significantly to illness and mortality rates worldwide. Early identification of individuals at risk is crucial to enable timely intervention and preventive measures. While traditional methods of risk assessment primarily rely on clinical indicators and medical history, advancements in technology now allow us to leverage machine learning to improve predictive accuracy and efficiency.

Machine learning algorithms excel at analyzing vast amounts of patient data, uncovering subtle patterns and risk factors that might go unnoticed with conventional techniques. In this project, we aim to harness the potential of three powerful algorithms— K-Nearest Neighbors (KNN), Logistic Regression and Gradient Boost Classifier—to build predictive models that classify individuals based on their likelihood of having heart disease. These models will analyze a wide range of inputs, Like Age, chest pain, Cholesterol etc.

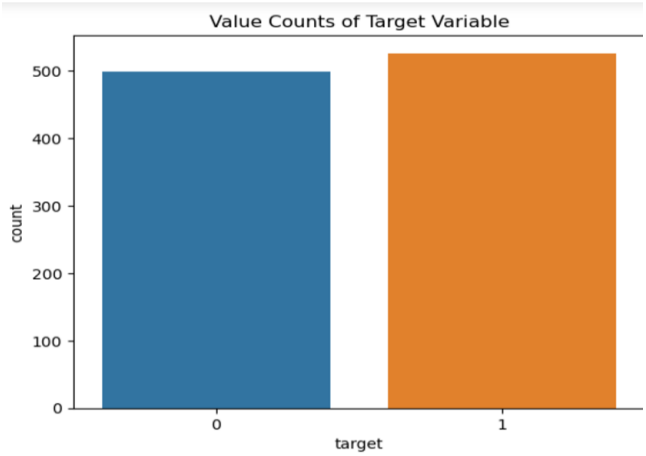
2. DATASET

The dataset utilized in this project was carefully preprocessed to ensure that it was suitable for predictive modeling after being purchased from Kaggle. It includes vital signs including blood pressure, type of chest pain, sex, and age, among others. To make model training and evaluation easier, preprocessing procedures include addressing missing data, encoding categorical variables, and normalizing numerical features have been done in the upcoming steps.

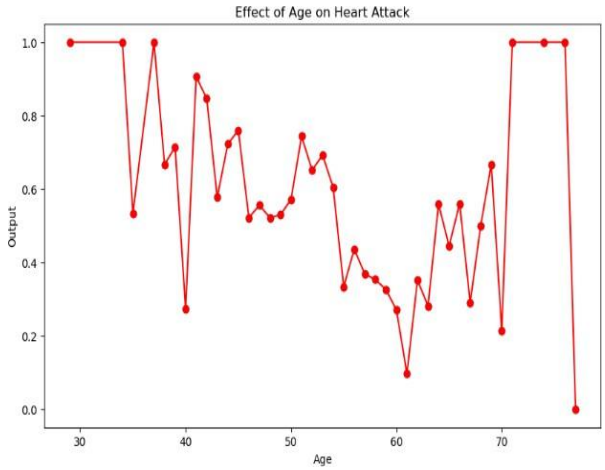
3. METHODOLOGY

3.1 Exploratory Data Analysis:

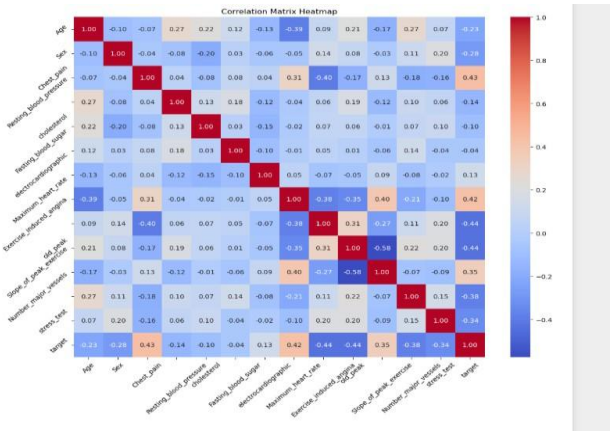
Checking for Balance in Target Variable:
We found that the target variable was well balanced in the data as they were equally seen in the dataset.



Effect of age on heart disease:
The heart disease is more seen in the age group of 30-45 and above 65. Which states that young adults and old age people are more prone to heart disease.



Heatmap:
Variables like Chest pain, Maximum heart rate, exercise induced angina, slope of peak exercise and electrocardiographic have a high correlation to the target variable.



3.2 Feature Engineering:

1. Categorical and Numerical Column Separation:
The dataset is split into features (X) and the target variable (y). The features in X are further divided into categorical and numerical columns:

- Categorical Columns Identification: A column is classified as categorical if it contains a discrete set of values (defined as 5 or fewer unique values). These columns are added to the cat_col list.
- Numerical Columns Identification: All other columns are considered numerical and added to the num_col list.

2. Encoding Categorical Features:
To prepare the categorical features for machine learning, they are encoded using one-hot encoding. This technique creates binary columns for each unique category in the original columns, representing the presence or absence of each category with 1s and 0s.

3. Normalization of Numerical Features:
The numerical columns, stored in num_columns, undergo normalization to standardize their scale. This ensures that all numerical features have values ranging between 0 and 1, which can improve the performance and convergence of machine learning algorithms.

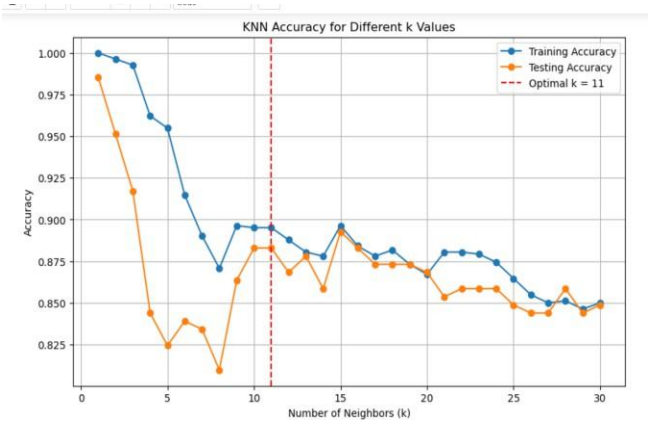
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

4. MODELS

4.1 K-Nearest Neighbors (KNN):

In this project, we implemented the K-Nearest Neighbors (KNN) algorithm to classify data and evaluate its performance. The choice of the hyperparameter (number of neighbors) was optimized by assessing the training and testing accuracies for values of k ranging from 1 to 30. Using a stabilization criterion based on minimal improvement thresholds for both accuracies, we identified the optimal value of k as 11. This optimal k provided a balance between model overfitting and

underfitting. The evaluation metrics, including accuracy, classification report, and a confusion matrix, further demonstrated the model's effectiveness in handling the given dataset.



4.2 Logistic Regression:

In this project, we used Logistic Regression to classify our data. We trained the model on the dataset by setting it up to learn the relationship between the features and the target. Once trained, the model was ready to make predictions based on what it had learned. This approach allowed us to apply a straightforward and effective classification method

4.3 Gradient Boost Classifier:

Gradient Boosting Classifier, a powerful ensemble learning technique that builds a series of decision trees to optimize predictive performance. We trained the model on the dataset, leveraging its ability to iteratively improve by minimizing errors from previous iterations. The classifier was configured with a fixed random state to ensure consistency and reproducibility. After training, the model was used to make predictions. This approach allowed us to integrate a robust machine learning technique.

5. MODEL EVALUATION

- **Logistic Regression:**

Logistic Regression provided solid and reliable performance, achieving an accuracy of **84.39%** and an F1-score of **0.84**. It served as a great starting point, offering balanced results and simplicity. However, its limitations became evident when dealing with more complex patterns in the data.

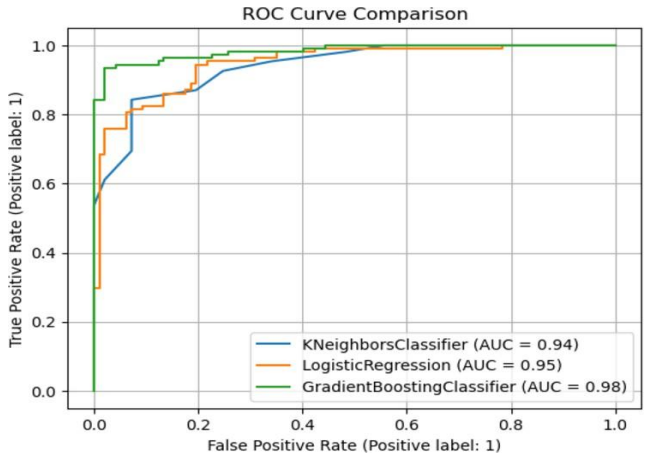
- **K-Nearest Neighbors (KNN):**

With fine-tuning to find the optimal k=11, KNN performed better, achieving an accuracy of **88.29%**. It demonstrated strong generalization and was able to capture non-linear relationships effectively. Despite this, it required careful tuning and became computationally demanding as the dataset size increased.

- **Gradient Boosting:**

Gradient Boosting truly stood out, achieving an impressive accuracy of **95.61%** and an AUC-ROC score of **0.98**. This model excelled at handling complex patterns in the data and provided the most accurate results. While it was computationally more intensive, its performance made it the best choice for this task.

6. RESULTS AND DISCUSSION



The ROC curve comparison gives a clear picture of how the models performed. Gradient Boosting stands out as the top performer, with its curve almost perfectly hugging the top-left corner, indicating near-perfect classification (AUC = 0.98). Logistic Regression follows closely with an impressive AUC of 0.95, showing its reliability as a straightforward and interpretable model. KNN also performs well, achieving an AUC of 0.94, though it slightly lags behind the other two models. Overall, the curves highlight the ability of each model to distinguish between the classes, with Gradient Boosting consistently leading the pack.

The ROC curve results demonstrate the balance between model complexity and performance. Logistic Regression, while simple and easy to interpret, proves to be a solid performer but struggles to capture more intricate data patterns. KNN performs competitively but depends heavily on proper parameter tuning and data structure. Gradient Boosting, on the other hand, excels by leveraging its iterative learning process to handle even the most complex patterns, making it the most powerful model for this dataset. However, its computational demands might make it less practical for very large datasets or real-time applications. This analysis highlights the importance of selecting a model based on the specific needs of the problem, balancing factors like accuracy, interpretability, and computational efficiency.

7. FUTURE DIRECTIONS

In the future, we could use more advanced techniques, like Grid Search or Random Search, to fine-tune the model settings and further improve performance. Incorporating tools like SHAP or LIME would also be valuable to make complex models, such as Gradient Boosting, easier to understand and explain. Testing the models on larger and more diverse datasets would help

evaluate how well they adapt to different scenarios. Additionally, focusing on improving computational efficiency would make these models more practical and scalable for real-world applications**.**

8. ACKNOWLEDGEMENT

We sincerely thank Dr. Chaofan Sun, our course instructor, for his guidance and support throughout the course, which provided the foundation and inspiration for this project.

9. REFERENCES

- Scikit-learn: Supervised Learning. *Scikit-learn User Guide*. Retrieved from https://scikit-learn.org/stable/supervised_learning.html.
- **Austin, P. C., et al. (2013).** Using methods from the data-mining and machine-learning literature for disease classification and prediction: A case study examining classification of heart failure subtypes. *Journal of Clinical Epidemiology.*
- **Latha, C. B. C., et al. (2019).** Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked.*
- 5 Minutes Engineering. *(YouTube Playlist: Machine Learning)*. Retrieved from https://www.youtube.com/playlist?list=PLYwpaL_SFmcBhOEPwf5cFwqo5B-cP9G4P.