DS 502/MA543: Statistical Methods for Data Science

Project Report - Group 5

Heart Attack Analysis & Prediction

Submitted By:

Atharva Kulkarni

Shubham Wagh

Sreeram Marimuthu

Nitya Phani Oruganty Santosh

Chithramvel Sanarpalayam Selvamuthukumar

Index:

1. Abstract:

The objective of this project is to construct a predictive model for heart attack classification utilizing the Heart Attack Analysis & Prediction Dataset sourced from the UCI ML Repository. Given that heart attacks represent a prominent global cause of mortality, timely identification and categorization of individuals at risk are imperative for effective intervention and prevention. This initiative endeavours to develop a machine learning model capable of forecasting the probability of an individual experiencing a heart attack, leveraging their medical data for enhanced predictive accuracy and early risk identification.

2. Introduction:

2. a. Description of the Data:

The dataset consists of demographic and clinical features related to heart disease diagnosis. The features include age, sex, chest pain type (cp), resting blood pressure (trtbps), serum cholesterol level (chol), fasting blood sugar level (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalachh), exercise-induced angina (exng), ST depression induced by exercise relative to rest (oldpeak), slope of the peak exercise ST segment (slp), number of major vessels colored by fluoroscopy (caa), and thalassemia (thall). The target variable "output" indicates the diagnosis of heart disease.

The Heart Attack Analysis & Prediction Dataset contains the following variables:

1. **Age:** The age of the patient.
2. **Sex:** The gender of the patient (0 for female, 1 for male).
3. **CP (Chest Pain Type):** Describes the type of chest pain the patient is experiencing (0 - asymptomatic, 1 - typical angina, 2 - atypical angina, 3 - non-anginal pain).
4. **Resting Blood Pressure:** The resting blood pressure of the patient.
5. **Cholesterol:** The cholesterol levels of the patient.
6. **Fasting Blood Sugar:** Indicates whether the fasting blood sugar is greater than 120 mg/dl (0 for no, 1 for yes).
7. **Resting ECG (Electrocardiographic Results):** Describes the results of the resting electrocardiogram (0 - normal, 1 - having ST-T wave abnormality, 2 - showing probable or definite left ventricular hypertrophy).
8. **Max Heart Rate:** The maximum heart rate achieved by the patient.
9. **Exercise-Induced Angina:** Indicates whether the patient experienced exercise-induced angina (0 for no, 1 for yes).
10. **ST Depression:** The ST depression induced by exercise relative to rest.
11. **Slope:** Describes the slope of the peak exercise ST segment (0 – down sloping, 1 - flat, 2 - upsloping).
12. **CA (Number of Major Vessels Coloured by Fluoroscopy):** The number of major vessels coloured by fluoroscopy.

13. **Thal (Thalassemia):** Describes the type of thalassemia the patient has (0 - normal, 1 - fixed defect, 2 - reversible defect).
14. **Target:** Indicates whether the patient had a heart attack (0 for no, 1 for yes).

Notably, the dataset comprises a mix of numerical and categorical features, there are 303 instances or observations in the dataset, and each instance is described by 14 features or variables. The information captured is used to predict the target variable, "output," denotes the presence or absence of heart disease.

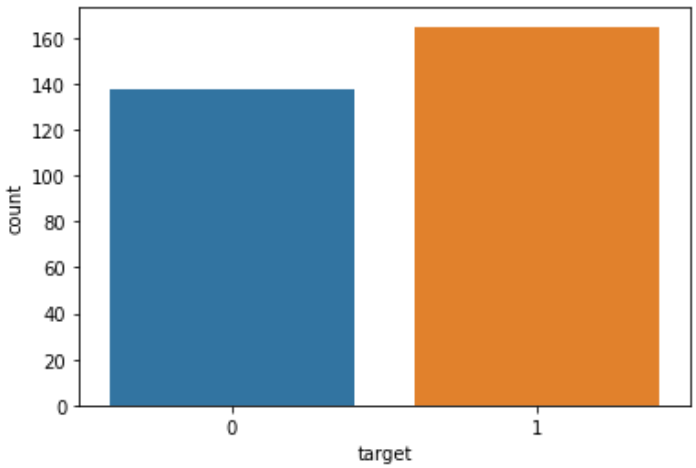
This dataset serves as a valuable resource for exploring the relationships between these features and the diagnosis of heart disease, providing insights into potential risk factors and contributing factors.

2. b. Objective:

The aim of this project is to create a ML model that can accurately and consistently predict the likelihood of patient experiencing a heart attack. This is a binary classification task where '0' signifies the absence of a heart attack and '1' denotes the presence of a heart attack.

3. Data Exploration & Pre-processing:

All the features contain values of type integer except for the “oldpeak” attribute which consists of float values. There are no missing values and the target outcomes are split into 165 positives to 138 negatives, so the dataset is not imbalanced. We performed some normalization of the data to bring the variables/values to a comparable scale, and split the data into training and test datasets with an 80:20 ratio.



4. Machine Learning Models:

4. a. Decision Tree:

4. a. i. Initial Model:

Tree based methods were our first choice, given the suitability for a classification task such as this but also because the final tree/model will be human-readable, so that a professional can verify the structure if required.

Training Set Performance:

The Decision Tree model, configured with a maximum depth of 3, demonstrates impressive accuracy on the training set, achieving an accuracy of 0.885. This indicates the model's ability to capture underlying patterns in the training data.

Testing Set Performance:

However, on the testing set, the model's performance is less robust with an accuracy of 0.705 and an F1 score of 0.727. The disparity between training and testing set performance reflects overfitting of the model.

Analysis:

The testing set accuracy of 0.705 and F1 score of 0.727 indicate a drop in performance compared to the training set. This discrepancy suggests that the model is not able generalize to new, unseen data.

Areas for Improvement:

Addressing Overfitting:

The model's overfitting tendencies, as indicated by the performance gap between training and testing sets, needs to be addressed. Fine-tuning the model's complexity or implementing regularization techniques may mitigate this issue.

Hyperparameter Tuning:

Tuning the hyperparameters using techniques like GridSearchCV will likely enhance the Decision Tree’s test performance. Finding the optimal values for parameters such as maximum depth and minimum samples split will be particularly important.

4. a. ii. Grid Search for Decision Tree:

Best Hyperparameters:

The Decision Tree model underwent hyperparameter tuning using GridSearchCV. The best hyperparameters identified through this process are {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 10}.

Training Set Performance:

Following hyperparameter tuning, the Decision Tree model achieved an impressive training set accuracy of 0.914. This indicates that the refined model improves on an already strong performance on the data it was trained on.

Testing Set Performance:

On the testing set, the model demonstrates an accuracy of 0.721. While this is improved compared to the initial model, there is still some minimum level of overfitting that persists in this approach.

F1 Score:

The F1 score on the testing set is 0.7385. This reflecting the model's fair ability to balance precision and recall. This metric provides a more comprehensive assessment of the model's performance, considering both false positives and false negatives. Given the nature of this project’s objective, it is particularly important to get a good F1 score and precision as the false positives in predicting a heart attack have to be kept to the absolute minimum.

Model Summary:

The hyperparameter-tuned Decision Tree model exhibits improved generalization compared to the initial model. The adjustment of hyperparameters has allowed the model to better capture underlying patterns in unseen data, yielding an improved test performance.

As anticipated initially, this model overfits the data but it is significant enough that the final test accuracy and precision is not nearly as high as we need it to be.

4. b. Random Forest:

4. b. i. Initial Model:

Having seen that even the optimized Decision Tree model overfits the data, the Random Forest model is the next choice as bagging the predictions will help mitigate the overfitting of using one tree alone.

Training Set Performance:

The initial Random Forest model demonstrates remarkable performance on the training set, achieving perfect accuracy of 1.0 with a confusion matrix.

```
Out[230]: array([[108,   0],
                 [  0, 134]], dtype=int64)
```

Overfitting remains to be analysed in testing.

Testing Set Performance:

On the testing set, the model's confusion matrix reveals some misclassifications as shown below.

```
Out[232]: array([[20, 10],
                 [ 6, 25]], dtype=int64)
```

The accuracy achieved is 0.7377 and the F1 score is 0.7576, indicating reasonable predictive performance but also highlighting areas for improvement. The model shows promising capability in identifying cases but requires further optimization.

4. b. ii. Grid Search for Random Forest:

Addressing Overfitting:

To mitigate potential overfitting observed in the initial model, a hyperparameter tuning approach was implemented using GridSearchCV.

Best Hyperparameters:

The best hyperparameters identified through the grid search are {'criterion': 'gini', 'max_depth': None, 'n_estimators': 100}. This refinement aims to strike a balance between model complexity and generalization.

Enhanced Testing Set Performance:

Following hyperparameter tuning, the refined model demonstrates improved generalization performance on the testing set. The accuracy is slightly enhanced to 0.786, but more notably the F1 score is improved to 0.806. This indicates a more balanced model than the normal Decision Tree that performs better on both the training and testing datasets.

4. c. Additional Approaches:

4. c. i. Using only Important Features (Random Forest):

Training Set Performance:

The Random Forest model applied to a modified dataset consisting of only important features (dropped: 'sex', 'restecg', 'oldpeak', 'thal', 'fbs', 'exang', 'slope'), performs very well on the training set. The confusion matrix indicates accurate classification, achieving a high accuracy of 0.9132.

```
Out[133]: array([[107,   2],
                 [  1, 132]], dtype=int64)
```

Test Set Performance:

On the testing set, the accuracy drops significantly to 0.6557, as shown by the confusion matrix. There is still a need for improvement, particularly in addressing overfitting.

```
Out[135]: array([[21,  6],
                 [ 9, 25]], dtype=int64)
```

4. c. ii. Principle Component Analysis

Since the model didn't improve with the consideration of important features for the data, we considered PCA dimensionality reduction technique for both the (4. a) & (4. b) models.

Decision Tree on PCA Data:

Initial Model Performance:

Training Set Accuracy: The Decision Tree model on PCA data achieves a high training set accuracy of 0.8760, indicating improved learning from the reduced-dimensional data over the original dataset.

Testing Set Accuracy: On the testing set, the accuracy is 0.7541, and the F1 score is 0.7887. The model demonstrates better generalization but there is still room for improvement.

Grid Search Results: The hyperparameter-tuned Decision Tree model on PCA data achieves a training set accuracy of 0.8843 and a testing set accuracy of 0.7705. The F1 score remains 0.7463.

Analysis:

- The initial model performs well on the training set but exhibits a slight drop in accuracy on the testing set, suggesting a need for optimization.
- The hyperparameter-tuned model improves testing set accuracy but still faces challenges in generalization.

Random Forest on PCA Data:

Initial Model Performance:

Training Set Accuracy: The Random Forest model on PCA data achieves high training set accuracy of 0.9587, indicating improved learning from the reduced-dimensional data over the original dataset.

Testing Set Accuracy: On the testing set, the accuracy is 0.7213, and the F1 score is 0.7463. The model exhibits fair performance but still exhibits overfitting.

Grid Search Results: The hyperparameter-tuned Random Forest model on PCA data achieves a training set accuracy of 0.9959 and a testing set accuracy of 0.7049. The F1 score remains at 0.7463.

Analysis:

- The Random Forest model on PCA data demonstrates improved performance but still struggles with overfitting, as shown in testing.
- Hyperparameter tuning improves training set accuracy but does not significantly enhance testing set performance.

Summary:

- Both Decision Tree and Random Forest models on PCA data show promising results in terms of training set accuracy.
- Decision Tree model benefits from hyperparameter tuning, showing improved testing set accuracy but with limited enhancement.
- Random Forest model with hyperparameter tuning does not provide a significant boost in testing set performance with PCA applied data.

4. d. XGBoost Classifier

The initial XGBoost model exhibited commendable performance with an accuracy of 0.852 and an F1 score of 0.857. For this binary classification problem, the model was configured with 'objective': 'binary:logistic' and 'eval_metric': 'logloss', reflecting its task of calculating the probability of the class. The model was trained over 100 rounds.

Upon conducting hyperparameter tuning through grid search, substantial enhancements were observed. The accuracy surged to 0.918, accompanied by an improved F1 score of 0.92. Noteworthy hyperparameter modifications include adjustments to 'colsample_bytree' (0.2), 'learning_rate' (0.2), 'max_depth' (7), 'n_estimators' (50), and 'subsample' (0.6). Cross-validation, employing 5 folds, was also integral to this process.

When subjecting the model to Principal Component Analysis (PCA), its performance experienced a decline, resulting in an accuracy of 0.7705. This suggests a reduction in predictive capability, indicating that the principal components exhibit weaker relationships between the predictors and the target variable compared to the original dataset.

Insights:

The selected hyperparameters for the grid search align with the initial model's configuration, indicating that the initial configuration was already near-optimal.

Key Findings:

Optimized Performance: The XGBoost model, starting with a robust accuracy of 0.852 and F1 score of 0.857, underwent significant refinement through hyperparameter tuning, achieving an accuracy of 0.918 and an F1 score of 0.92.

Exploration for Further Improvement: Given the high initial performance, potential avenues for further improvement may involve exploring additional features, refining data preprocessing techniques, or engaging in feature engineering.

PCA Impact: The reduction in accuracy to 0.7705 when using PCA indicates that the principal components captured in this analysis exhibit weaker relationships between the predictors and the target variable compared to the complete feature set.

In summary, the initial XGBoost model demonstrated strong capabilities, and hyperparameter tuning further optimized its performance. The findings underscore the importance of considering alternative strategies for improvement and highlight the impact of dimensionality reduction techniques like PCA on model performance.

4. e. Logistic Regression

The initial Logistic Regression model demonstrated commendable performance with an accuracy of 0.885 and an F1-Score of 0.89, utilizing default model settings. These metrics indicate the model's proficiency in correctly classifying instances within the dataset.

Hyperparameter Tuning through Grid Search:

In pursuit of optimizing the model, a grid search was conducted, resulting in the identification of hyperparameters 'C' (0.1), 'penalty' ('l2'), and 'solver' ('sag'). Intriguingly, despite the changes in hyperparameters, the accuracy remained consistent with the default model. Cross-validation, performed with a specified number of folds, added a robustness check to the hyperparameter tuning process.

Classification Report Insights:

The optimized model, post grid search, exhibited a classification report showcasing balanced precision, recall, and F1-Score for both classes (0 and 1). This balanced performance suggests the model's ability to generalize effectively to previously unseen data, capturing instances from both classes with similar accuracy.

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.86	0.88	29
1	0.88	0.91	0.89	32
accuracy			0.89	61
macro avg	0.89	0.88	0.88	61
weighted avg	0.89	0.89	0.89	61

Key Findings:

Baseline Performance: The Logistic Regression model displayed commendable performance with an initial accuracy of 0.885 and an F1-Score of 0.89, even with default settings.

Hyperparameter Tuning Impact: The grid search process suggested changes to hyperparameters, leading to the same accuracy but with refined model settings ('C': 0.1, 'penalty': 'l2', 'solver': 'sag'). This indicates that subtle adjustments in hyperparameters can influence the model's behavior.

Balanced Metrics: The balanced precision, recall, and F1-Score for both classes signify the model's ability to handle different data patterns and generalize well. This characteristic is crucial for real-world applicability.

Further Optimization Opportunities: While the logistic regression model performs well, there is room for further improvement. Exploring additional features or engaging in feature engineering may unveil latent patterns in the data.

Comparison with XGBoost Model: Although Logistic Regression shows good performance, it is acknowledged that the XGBoost model outperforms it. This indicates that, for this specific dataset, the XGBoost algorithm may capture more complex relationships and nuances.

In conclusion, the Logistic Regression model exhibits robust performance, and the optimization through hyperparameter tuning reinforces its capabilities. The balanced metrics suggest generalizability, but opportunities for improvement exist, especially in comparison to more sophisticated models like XGBoost.

5. Why XGBoost?

In our comprehensive evaluation of various machine learning models for the given task, we assessed each model's performance based on accuracy and F1-Score. Among the models considered, XGBoost stands out as the most promising and effective choice for several key reasons.

The optimized XGBoost model showcases a substantial accuracy improvement from 0.819 to 0.918, indicating its robustness in capturing complex patterns in the data. The F1-Score also sees a noteworthy boost from 0.825 to 0.92, underscoring the model's ability to balance precision and recall effectively.

XGBoost consistently outperforms other models, including the Random Forest and Logistic Regression, both in their initial and optimized states. The balanced precision, recall, and F1-Score for XGBoost indicate its superior generalizability, crucial for reliable predictions on new, unseen data.

Finally, XGBoost's capacity to capture complex relationships in the data positions it as the preferred choice for this particular classification task.

In conclusion, the XGBoost model, particularly in its optimized state, emerges as the most effective and robust choice for the given dataset. Its ability to handle intricate patterns, combined with significant accuracy and F1-Score improvements through hyperparameter tuning, makes it the optimal model for this classification problem.

6. Summary of Results:

Upon conducting an extensive analysis of various models and exploring the dataset's features, the following insights and observations emerged:

- Visual examination of feature-target variable plots and other Exploratory Data Analysis (EDA) revealed discernible patterns within the raw data.
- The distribution of values displayed a somewhat normal distribution, prompting consideration for further standardization. However, it was noted that excessive standardization might diminish the significance of outliers, which are valuable indicators of potential heart attacks.
- Since the dataset does not have too many features (>20), removing the extra features (using only best features) were not expected to make a significant improvement in general. This was attempted only to try and lower the overfitting of the tree-based methods.

- PCA was also initially dismissed due to the good separation/patterns in the original data, but was attempted to confirm our hypothesis. Using the PCA dataset (after standardization of original data) with variations of the number of components/axes (5, 8, 10) only slightly improved the performance of Tree based models and actually lowered the performance of the other models.

The models' performance, as highlighted in the table, aligns with the initial hypothesis. Notably, tree-based models (Decision Tree and Random Forest) showcased sensitivity to feature variations, while Logistic Regression and XGBoost demonstrated robustness.

Model	**Accuracy**	**F1-Score**
Decision Tree (DT)	0.705	0.727
Random Forest (RF)	0.7377	0.7576
GridSearch DT	0.721	0.7385
GridSearch RF	0.786	0.806
RF (Important Features)	0.6557	0.672
DT with PCA	0.7541	0.7887
GridSearch DT with PCA	0.7705	0.7463
RF with PCA	0.7213	0.7463
GridSearch RF with PCA	0.7049	0.7463
XGBoost	0.819	0.825
Logistic Regression	0.8852	0.89
XGBoost (Optimized)	0.918	0.92
Logistic Regression (Optimized)	0.8852	0.89

7. Conclusions

While feature engineering and exploratory analyses were undertaken, the impact on model performance varied. The consideration of feature subsets and PCA did not uniformly enhance predictive capabilities across all models.

The detailed model evaluation, as illustrated in the table, accentuates the superior performance of XGBoost, particularly after hyperparameter tuning. This model consistently outperforms others, reaffirming its suitability for this specific classification task.

In conclusion, the strategic application of feature engineering techniques and careful consideration of model selection underscore the importance of tailoring approaches to the unique characteristics of the dataset. The optimized XGBoost model stands out as the most effective choice, aligning with both initial hypotheses and extensive model evaluation results. The resulting model shows good performance in testing and unseen data compared to any other model as shown above.

8. References

[1] Janosi,Andras, Steinbrunn,William, Pfisterer,Matthias, and Detrano,Robert. (1988). Heart Disease. UCI Machine Learning Repository - https://doi.org/10.24432/C52P4X.