TITLE: A Comparison of Decision Trees and Naïve Bayes Applied to a Heart Attack Dataset

Coursework: INM431 Machine Learning Name: Nicholas Tsioras Student ID: 230020402 Email: Nicholas.Tsioras@city.ac.uk

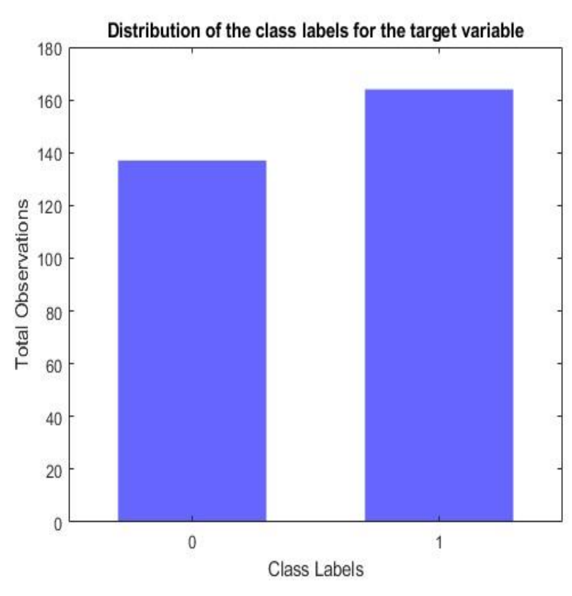
Description and Motivation:

- Building two models for a classification task. One optimized model for each machine learning algorithm (decision trees and naïve bayes)
- Comparing and analyzing the performance of those 2 models from predicting the chance of a patient getting a heart attack (0: No chance of heart attack, 1: high risk of heart attack)

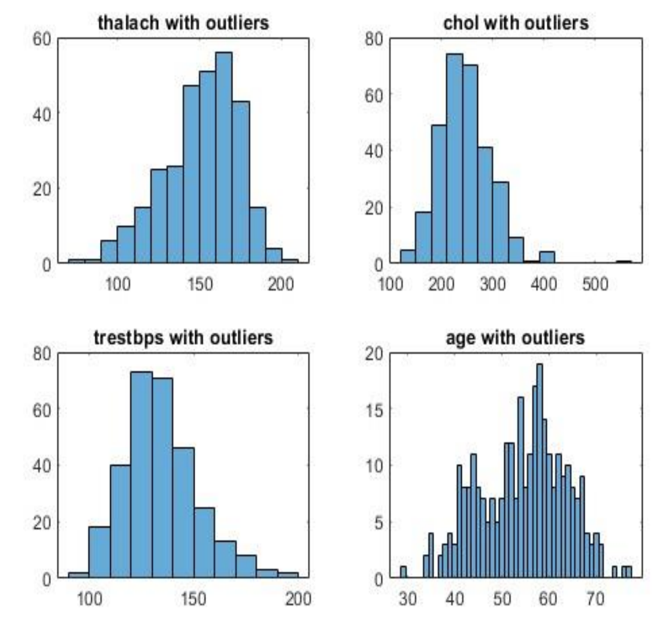
Exploratory Analysis of the Dataset

- Dataset: Heart attack dataset from Kaggle (https://www.kaggle.com/datasets/pritsheta/heart-attack)
- The dataset consists of 12 attributes and 303 rows that describe the medical information that can help us predict whether a patient has a high risk of heart disease or not. We also have 1 binary attribute that describes if the person has a heart disease or not
- We can see in graph 1 that there is a slight imbalance in the target variable that we want to predict. Most observations in the dataset seem to have patients with a heart disease
- In graph 2 we can see that most numerical variables (thalach: Maximum number of beats per minute during exercise, chol: cholesterol, trestbps: resting blood pressure and age) are skewed either to the right or left. It is important to remove the outliers and apply normalization to the data.
- In graphs 3-8, we can see the main differences between patients with and with not a heart attack. From the graphics shown we can see that patients with a heart attack have higher chest pain and slope levels. Also, we can see from the resting electro diagram results that patients with a heart attack have an ST-T wave abnormality (value 1). Another observed difference from the thallium test results is that people with a heart attack have mainly a fixed defect (value 1) while patients with no heart attack have a normal thallium (value 0) and a reversible defect (value 2).
- After performing exploratory analysis and finding the main differences between people with a heart disease (target label 1) and people with no heart disease(target label 0) we created a heatmap (graph 10) to visualize the spearman correlations between the variables. The reason we chose the spearman correlation coefficients to analyze the relationship of our variables is because the dataset contains both ordinal and interval data. So, the data is in different scales, and we can't assume that there is a linear relationship between the variables [4]. Based on the correlation heatmap, we can see that the chest pain levels (cp) have the highest, positive correlation with our target variable. Also, the resting blood pressure (trestbps) seems to have a positive correlation with our target variable. There also seems to be a strong negative correlation between the target variable and the exercise induced angina (exang). We can also see that some independent variables have a strong correlation between them but there is no correlation above 0.5 so we don't need to worry about multicollinearity between the independent variables. All this information is useful for the feature selection before building and evaluating the models

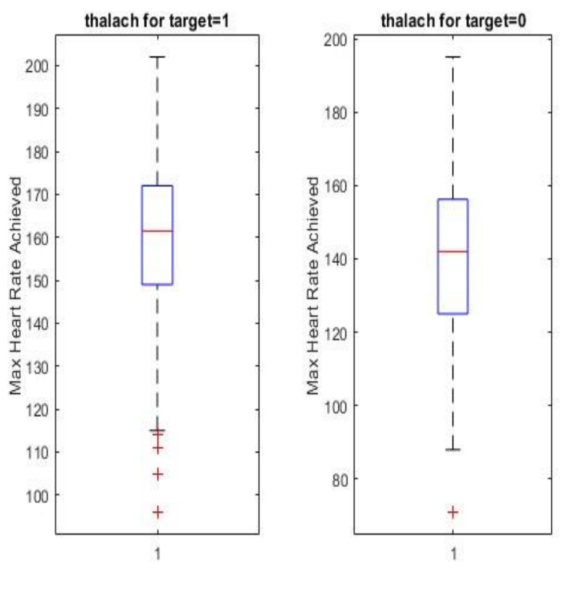
GRAPHICS

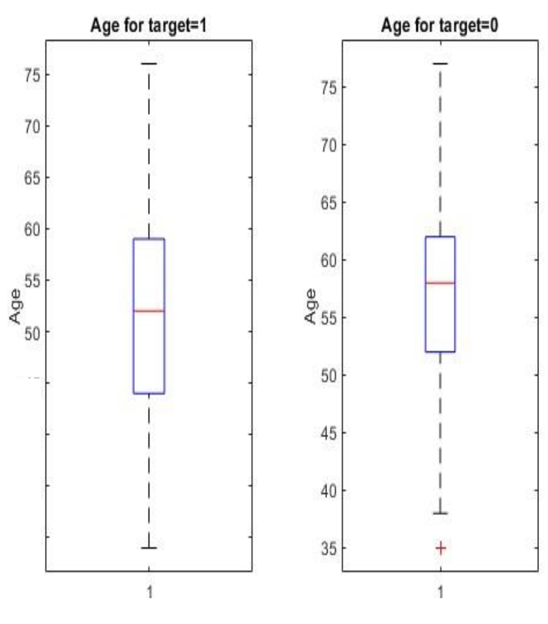


Graph 1. Total class labels for the target variable

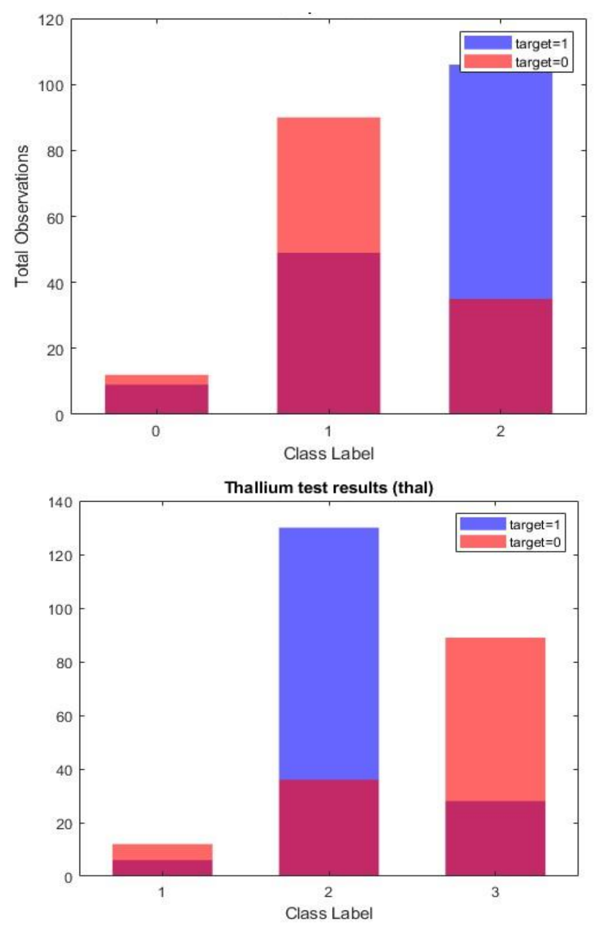


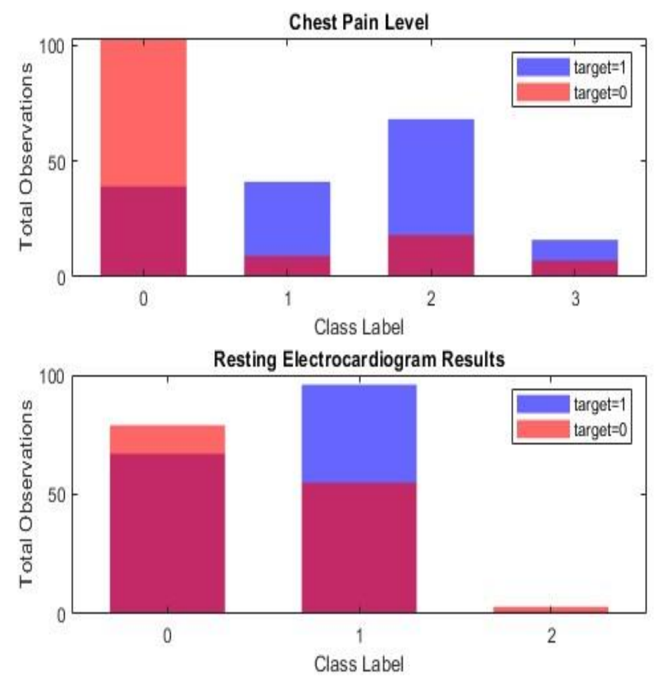
Graph 2. Data Distribution of Numerical Variables before normalization with outliers



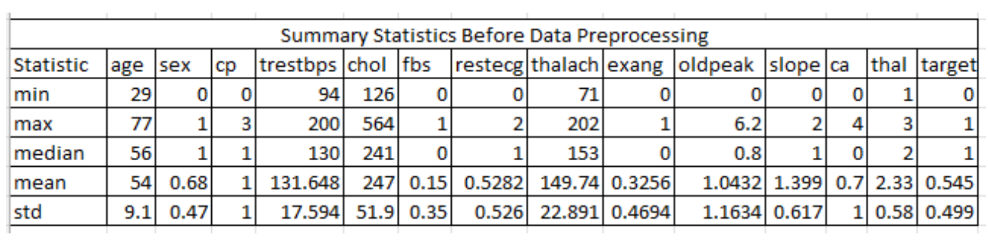


Graphs 3 and 4. Box Plots of 'thalach' and 'age' for 'target' label 1 and 0



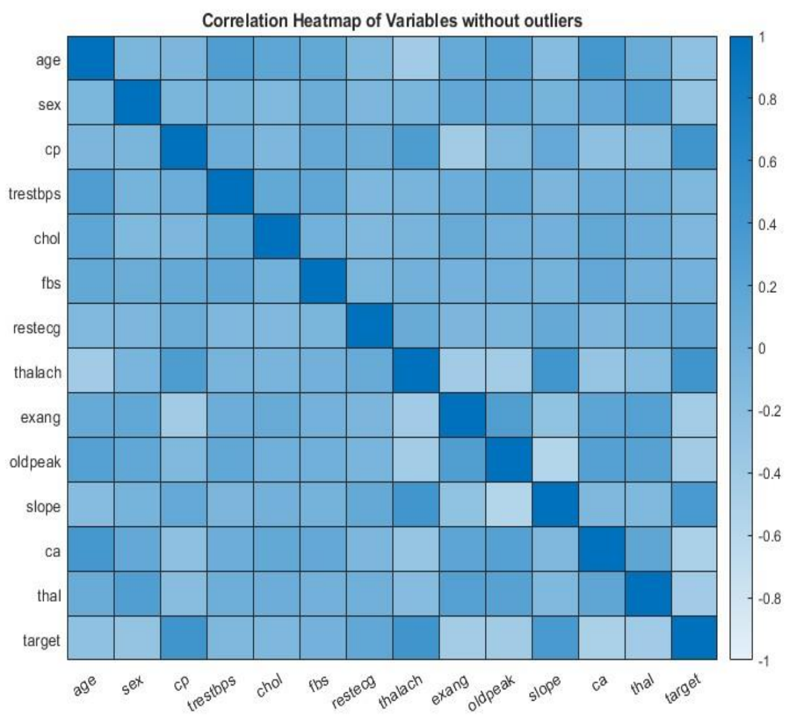


Graphs 5,6,7 and 8. Histograms of 'Slope','Thal', 'cp' and 'restecg'.
Blue bar: 'target' label=1, Red bar: 'target' label=0



Summary Statistics Before Data Preprocessing														
Statistic	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
min	29	0	0	94	126	0	0	71	0	0	0	0	1	0
max	77	1	3	200	564	1	2	202	1	6.2	2	4	3	1
median	56	1	1	130	241	0	1	153	0	0.8	1	0	2	1
mean	54	0.68	1	131.648	247	0.15	0.5282	149.74	0.3256	1.0432	1.399	0.7	2.33	0.545
std	9.1	0.47	1	17.594	51.9	0.35	0.526	22.891	0.4694	1.1634	0.617	1	0.58	0.499

Graph 9. Summary Statistics Before Data Preprocessing



Graph 10. Spearman Correlation Heatmap of Variables (Without Outliers)

NAÏVE BAYES

- The Naïve bayes algorithm is called Naïve because it assumes that there is class conditional independence. This means that all variables are mutually correlated and contribute towards classification [3].
- Bayesian classification is based on the use of Bayes Theorem. Given A naïve bayes model learns how to predict the posterior probability of a specified class C, given a data sample X, where C is a class label of the variable we want to predict [3].
- Bayes theorem is an equation from where we can calculate the posterior probability P(c|x) from the prior probability of each class of our target variable y P(c), the prior probability of the predictor dataset P(x) and the likelihood which is the probability of observing the predictor dataset X given teach class label C of our target variable y P(x|c) [6].
- This algorithm uses a kernel function that estimates the probability density function of the input data, which can help the model improve its performance in complex scenarios [6]
- **Advantages** of the Naïve Bayes Algorithm [3]:

✓ Training time is not long

✓ There is a variety of naïve bayes models which can be used based on the nature and distribution of the data (Gaussian Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Categorical Naïve Bayes). It can be used for both binary and multi class prediction problems [6]

✓ The classification performance improves after removing irrelevant features

- **Disadvantages** of the naïve bayes algorithm:

❖ It requires many observations to obtain good results [3]

❖ For a categorical variable, if there is a class label present to the test set, but not observed in the training set, naïve bayes automatically assigns zero probability and will not be able to make a prediction. This is known as zero frequency [6]

DECISION TREES

- Decision Trees is a machine learning algorithm that can be used for regression as well as classification tasks[1]. It is a non-parametric algorithm which is used in supervised learning. It has a hierarchical tree structure and consists of a root node, branches, internal nodes, and leaf nodes forming a tree-like structure [5].
- A decision tree is a step-by-step guide that begins from a starting point (the root node) and 'branches' into different conditions and possibilities [5]. According to [1], *«DT is a tree-shaped diagram representing a sequential decision process in which attribute values are successively tested to infer an unknown state»*.
- **Advantages** of decision trees [1],[3]:

✓ Easier to understand because they do not require any domain knowledge or parameter settings for classification problems

✓ More robust to noisy data

✓ The have the ability to build classifiers for datasets with both numerical and non-numerical data

✓ It can use different measures such as Entropy and Gini Index to find the best attribute to split the tree

- **Disadvantages** of decision trees:

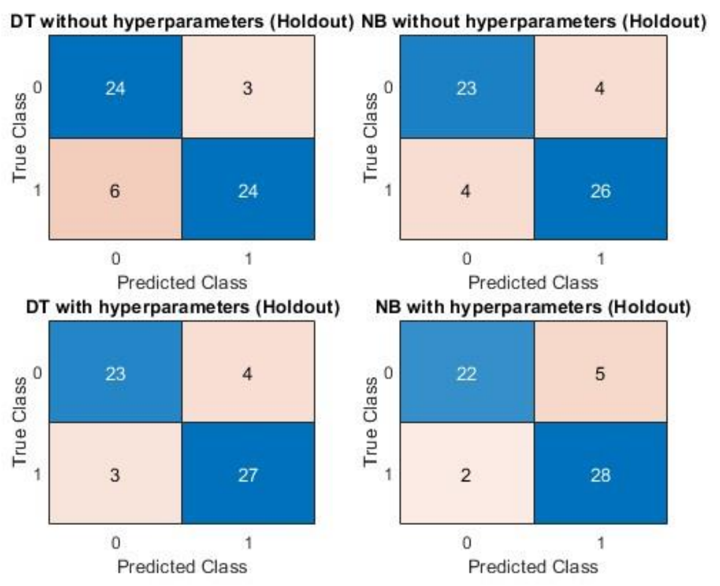
❖ Complex decision trees are prone to overfitting [7]
❖ The output variables must be single and categorical [1]
❖ It has long training time [3]
❖ It is sensitive to variations within the data [7]

Hypothesis Statement

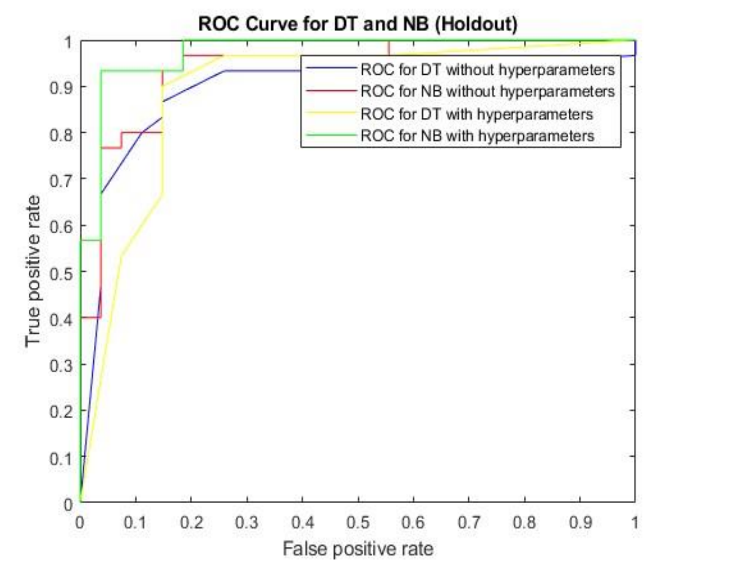
- From a literature review comparing the naïve bayes and decision trees algorithm, the naïve bayes requires a large dataset to perform well. In our case we only have a total of 303 rows in our initial dataset and only 287 rows after removing the outliers. We expect the decision trees model to perform moderately better [3].
- As we can see in graph 1, the dataset is quite imbalanced. In our case, there are more cases with patients diagnosed with heart disease (target=1). We expect the minority class, which is patients not having a heart disease (target=0) to have the most cost of misclassification.

Methodology:

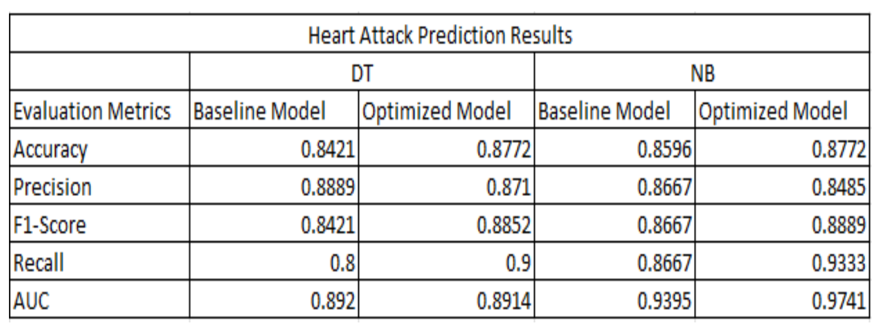
- First, we applied some important data preprocessing techniques that will help with model performance. We removed a total of 16 rows with outliers. We also normalized the numerical variables of our dataset because the scale of values was very different to the categorical variables.
- We analysed the relationship between the dependent and independent variables but also between the independent variables themselves. The reason we did this is to check which variables have a statistically significant relationship with the dependent variable but also to detect multicollienarity. We used the p-value to test the significance of each relationship with a threshold of 0.05 for the hypotheses statement. This will be helpful for the feature selection.
- After analysing the significance of the relationship between the variables, we selected the features that we think will optimize our model. This means that we eliminated the irrelevant features and chose the variables that have the highest, statistically significant correlation with our dependent variable but at the same time have low correlation between them. These variables are 'cp','thalach', 'exang', 'ca','thal'.
- We then split the dataset into an 80:20 split for train and test data using holdout cross validation. This means that 80% of our dataset was used to train the models and 20% to evaluate them. The reason we chose to apply cross validation to our dataset is to avoid the problem of overfitting. An 80:20 split to our dataset should be the most appropriate to avoid overfitting but at the same time to have enough observations to train the models. A 90:10 split would risk the chance of overfitting and a 70:30 split or less would mean we don't have enough data to train on, since our dataset is small [2].
- After training the models and testing them using different evaluation metrics such as accuracy, precision, f1-score, recall, AUC and the ROC curve, we applied hyperparameter optimization to our 2 models to improve their performance. We set the hyperparameter optimization process for matlab to tune the parameters automatically to speed up the process.
- Finally, after saving the optimized models, we evaluated them using the same test set and the same evaluation metrics. This will help us see how the hyperparameters improved model performance but also compare the performances of the 2 machine learning algorithms.



Graph 11. Confusion chart for baseline and optimized models

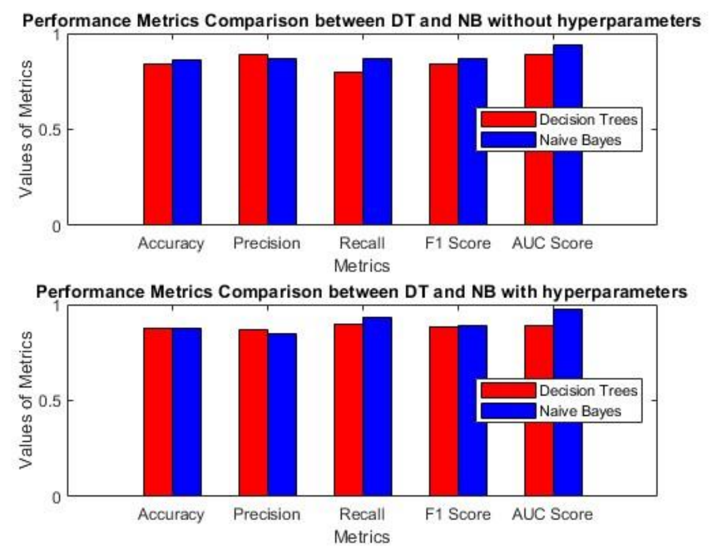


Graph 12. ROC curve for baseline and optimized models

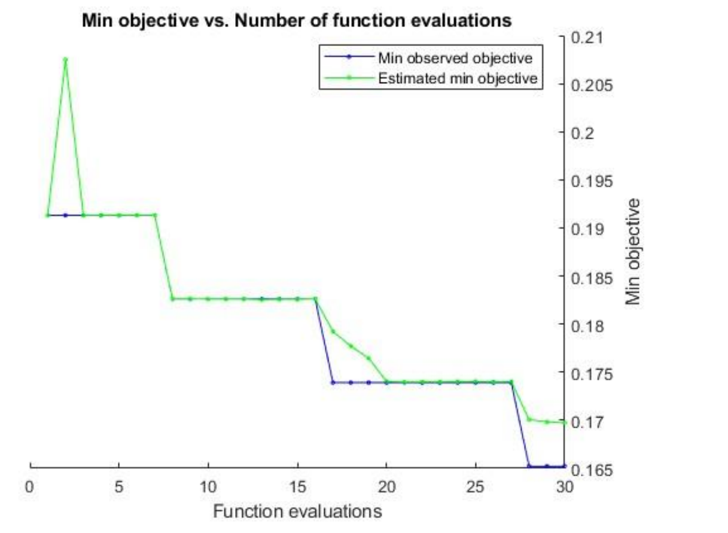


Heart Attack Prediction Results				
	DT		NB	
Evaluation Metrics	Baseline Model	Optimized Model	Baseline Model	Optimized Model
Accuracy	0.8421	0.8772	0.8596	0.8772
Precision	0.8889	0.871	0.8667	0.8485
F1-Score	0.8421	0.8852	0.8667	0.8889
Recall	0.8	0.9	0.8667	0.9333
AUC	0.892	0.8914	0.9395	0.9741

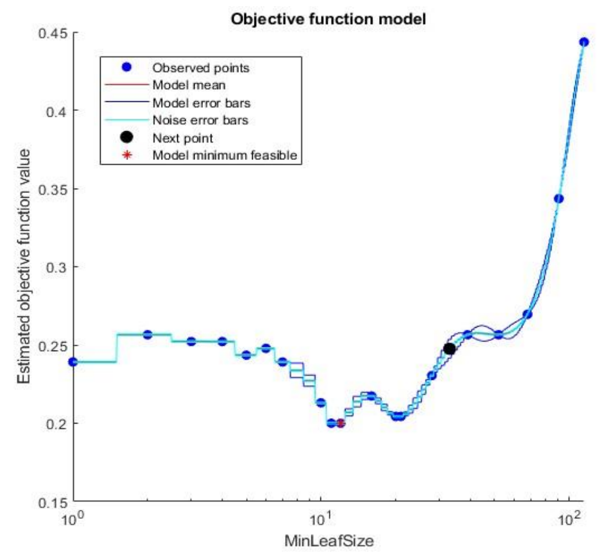
Graph 13. Table of performance metrics comparison between DT and NB



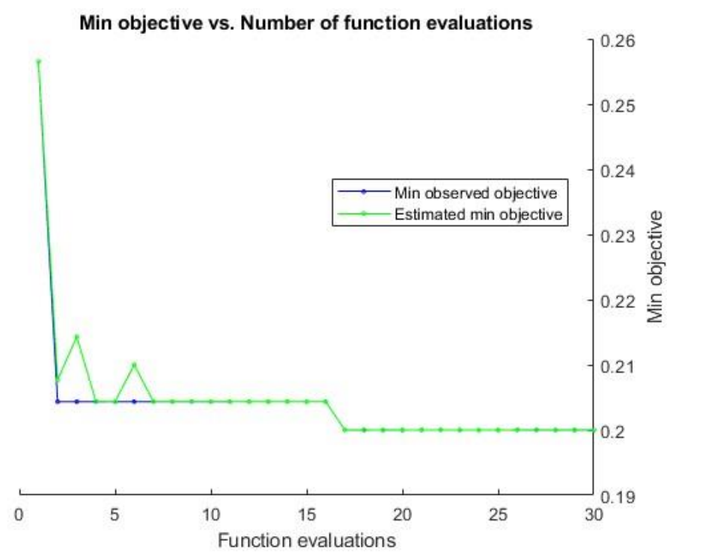
Graph 14. Bar chart of performance metrics comparison between DT a NB



Graph 17. Min objective vs Number of Function evaluations for Naïve Bayes hyperparameters



Graph 16. Objective function model for decision trees hyperparameter optimization



Graph 18. Min objective vs Number of Function evaluations for Decision Trees hyperparameters

ANALYSIS AND CRITICAL EVALUATION OF THE RESULTS

- It seems that the naïve bayes baseline model is performing overall better than the decision trees model. What is interesting though is that the precision is the only metric that has a lower value for naïve bayes. This indicates that when the decision trees model predicts that a patient has a heart attack, it is more likely to be correct.
- It seems that the optimized decision tree model has a'MinLeafSize' of 11 with a measure of loss equal to 0.2. After running the evaluation models on the best estimated feasible point of 'MinLeafSize' equal to 12, the model provided a slightly worse performance.
- The optimization process for the naïve bayes model has identified that the hyperparameters optimize the model with a minimized objective function value to 0.16622 and a 'Width' of 0.42513 without standardizing the features.
- Looking through the charts on the right that compare the baseline with the optimized models for each machine learning algorithm, we can see that both models improve most of their metrics, except their precision.
- In our case, we have a slightly imbalanced dataset with most instances being patients with a heart attack (graph 1). So it is better to evaluate the models based on the precision, f1 score and recall which are more accurate measurements for imbalanced datasets. Overall, it seems that the best naïve bayes model performs better than the best decision trees model, which makes our hypothesis statement false of assuming that the naïve bayes model will struggle to make accurate predictions due to the small number of instances in our dataset.
- Looking at the confusion charts (graph 11), we notice that our other assumption seems to be also false for the decision trees and naïve bayes optimized models. The minority class (no heart attack, 'target'=0) has less misclassification than the majority class (heart attack, 'target'=1).
- Finally, we notice that even after hyperparameter optimization the decision trees model has a higher precision than the naïve bayes but the rest of the metrics are lower. This indicates that a greater proportion of the positive predictions are correct for the decision trees model. In our case, this means that the decision trees model has a higher chance of predicting a higher proportion of patients who were correctly identified of having a heart attack. Also, the decision trees model predicts less false positives, meaning that the naïve bayes model has a higher chance of predicting that someone has a high chance of a heart attack, when in reality they don't. So we also need to take this into account while comparing the two models.

Lessons Learned

- The importance of each metric is different in each classification problem. Depending on the nature of the problem, sometimes certain metrics are more important than others. In our case, where we want to predict the chance of a patient getting a heart attack or not, the choice of evaluation metrics should reflect on the importance of identifying true positive and true negative cases. The optimized naïve bayes model has a higher recall but the decision trees has a higher precision. In other words, the main difference between the 2 models is that the naïve bayes model has less false negatives and the decision trees model has less false positives. This means that when the decision trees model predicts a heart attack, it is more likely to be correct and there is less chance to predict a heart attack when in reality there isn't. On the other hand, the naïve bayes model is more accurate in identifying actual positive cases. This means that there is less of a chance that the model will predict a heart attack, when in reality there isn't one. In our cases, the differences are very small but in other situations where the difference in these metrics is higher, we need to take all of this into consideration.
- Feature engineering is important for optimizing the models. Selecting independent variables that have a high correlation between them (multicollinearity) can lead to inefficient model performance and false predictions. Depending on the nature of the data, data preprocessing is an important step to optimize machine learning models.

Future Work

- Comparing this analysis to other papers that use the same dataset. This will give us the opportunity to compare the results with other studies that use different approaches and will help us identify how we could improve model performance. One of these studies could be
- Using methods that tackle the problem of target imbalance e.g., SMOTE (Synthetic Minority Oversampling Technique)
- Increase the samples in our dataset to improve the models, especially for the naïve bayes algorithm.
- Apply k- fold cross-validation and compare the results to the holdout method we used. The main difference is that k-fold splits the data into k equal parts with k-1 parts used for training and one part is used for the testing process. K-fold is an iterative process and ends when the testing has been applied to all different k parts of the dataset. This gives a more robust estimate of the models performance and can be useful to see how the models generalize to new unseen data [2].

REFERENCES

[1] M. Al Hamad and A. M. Zeki, 'Accuracy vs. Cost in Decision Trees: A Survey', in *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakhier, Bahrain: IEEE, Nov. 2018, pp. 1–4. doi: 10.1109/3ICT.2018.8855780. Accessed: Dec 16, 2023

[2] S. Yadav and S. Shukla, 'Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification', in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Bhimavaram, India: IEEE, Feb. 2016, pp. 78–83. doi: 10.1109/IACC.2016.25. Accessed: Dec 18, 2023

[3] S.D. Jadhav and H.P. Channe, 'Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques', *IJSR*, vol. 5, no. 1, pp. 1842–1845, Jan. 2016, doi: 10.21275/v5i1.NOV153131. Accessed: Dec 18, 2023

[4] J. Hauke and T. Kossowski, 'Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data', *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, Jun. 2011, doi: 10.2478/v10117-011-0021-1. Accessed: Dec 18, 20

[5] A. Saini, 'Decision Tree Algorithm – A Complete Guide' . Accessed: Dec 17, 2023. [online]. Available at: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/?fbclid=IwAR11UTKNrPnE-UyBeSenTgixouhS7TisaM7yEdQH8PR8LAajElF12GlwQjA#:~:text=A%20decision%20tree%20algorithm%20is,each%20node%20of%20the%20tree

[6] S. Ray 'Naïve Bayes Classifier Explained: Applications and Practice Problems of Naïve Bayes Classifier'. Accessed Dec 18, 2023. [online]. Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/?fbclid=IwAR0fRZ6vJaebbn6WbTuRyppu6dUlDrEv2yeA_qUz5YpTDL_nowwZMpF0qVM

[7] 'What is a Decision Tree?' – IBM . Accessed: Dec 18, 2023. [online]. Available at: https://www.ibm.com/topics/decision-trees?fbclid=IwAR0C27SKnBKSaW2kMFjbi8qc6CkGrjQqzr_RRiFQ_MtbdiR0_sWvjIliUjk