

COMP90049: Introduction to Machine Learning - Project Report

Heart Disease Prediction with Hybrid Resampling

Student Name: Matthias Si En Ong

Student ID:

Introduction

According to the World Health Organization, approximately 17.9 million deaths occur globally each year due to heart diseases [1]. This motivated me to explore and predict whether a person is likely to have heart disease using health, lifestyle, and other data. I am personally driven and curious about healthcare. I volunteer at The Royal Children's Hospital biweekly and am very curious how technology can help people and would love to do my research project on healthcare in my final semester. The Heart Disease indicators dataset from Kaggle [2] was already processed for Heart Disease classification from the complex 2022 annual CDC Behavioral Risk Factor Surveillance System (BRFSS) dataset. It has 40 features (further processed by me to 54 features) and 246,022 instances. I chose the one with no NaNs as it has less data and was suitable for my computational resources.

Literature Review

Significant research related to the prediction of cardiovascular heart disease (CVD) using machine learning (ML) has inspired this project. The healthcare industry has seen significant advancements in ML, particularly in diagnosing heart disease. The rapid accumulation of medical data presents an opportunity for researchers to develop and test new algorithms in the field. In a study conducted by D. Shah et al. (2020) [3], the authors aimed to predict CVD using supervised classification methods such as Naive Bayes, Decision tree, Random forest, and k-nearest neighbor (KNN). They used a dataset of 303 instances and 177 attributes. K-nearest KNN, Naive Bayes, and Random forest are the algorithms showing the best accuracy. In a study by K. Drożdż et al. (2022) [4], they used ML techniques to identify the most significant risk variables for CVD in patients with metabolic-associated fatty liver disease (MAFLD). Data were collected through assessments performed on 191 MAFLD patients. Overall, the model performed well, correctly identifying 40/47 (85.11%) high-risk patients and 114/144 (79.17%) low-risk patients with an AUC of 0.87. However, the main downside of past research is the limited dataset, resulting in a risk of overfitting and poor generalisation. Additionally, they often focus on accuracy as a metric, which while important, a model that performs better at recall to minimise false negatives is also paramount in the healthcare context.

Methods

Feature Construction and Preprocessing

I dropped 'ChestScan' as they are usually done after a diagnosis, which could be cheating and uninformative. I also dropped 'State' as although it has meaning, I want to train a model that is more generalisable across countries. I combined 'HadHeartAttack' and 'HadAngina' into 'HeartDisease'. This is my target label to predict general heart disease risk. I created 'DisabilityScore' as a new informative feature as a sum of 'DifficultyWalking', 'DifficultyConcentrating', 'DifficultyDressingBathing' and 'DifficultyErrands' to create a single, ordinal feature that captures overall functional burden. This serves to improve interpretability and emphasize ordinality. I also checked for outliers in numerical features by plotting their distribution but found no significant outliers so nothing needs to be removed. I also did a correlation check between numerical features by computing and plotting their Pearson correlation matrix. I found a correlation of 0.86 between 'WeightInKilograms' and 'BMI'. While this might initially suggest dropping one of the features to reduce redundancy and potential multicollinearity, I chose to retain both. BMI is a standardised and widely accepted metric in healthcare. Even though it's derived, it encodes interactions between weight and height that may not be as easily or reliably learned from the raw features. However, while it is a useful general indicator, it can sometimes oversimplify body composition. Next, I did categorical binning for 'SleepCategory', 'BMICategory'. This could potentially improve model performance by capturing non-linear feature patterns. It allows encoding of some domain knowledge e.g. sleep time below 7 hours is known to be a lot

worse for health than variations between 7 and 9 hours. It also gives interpretability to my tree-based models. I still kept the original columns to keep my dataset versatile with different models. I proceeded to perform Ordinal encoding to 'SleepCategory', 'BMICategory', 'GeneralHealth', 'LastCheckupTime', 'AgeCategory' to maintain order. Following this, I performed one hot encoding on my remaining nominal features. Lastly, I normalised my numerical features (not the encoded features) with StandardScaler, it helps Naive Bayes more than the tree-based models. Before training, I did Univariate feature selection using Mutual Information, however I decided not to remove any features yet despite a handful of features having low MI scores (e.g. 'HighRiskLastYear', 'HadDiabetes', 'BMI', etc), they seemed potentially important and I did not want to make any early decisions just based on mutual information at this point. In total, I now have 246022 rows and 54 features.

The Models

I chose to train a Gaussian Naive Bayes model (High bias/Low variance), a Decision Tree classifier (Low bias/ High variance) and a Random Forest (Medium bias/Medium variance, reduced via ensembling). This covers a spectrum of bias and variance of different models. In my dataset, there is a large class imbalance of ~91% 'HeartDisease' No, ~9% Yes. Hence, my Cross-Validation (CV) Strategy is using Stratified 5-Fold Cross-Validation in the Outer Loop and Stratified 3-Fold for the inner loop for Hyperparameter Tuning. I also made use of `class_weight='balanced'` in Decision Tree and Random Forest to counter class imbalance. This was essential as I was getting close to 0 F1 and recall scores without class weights and in medical settings, false negatives should be minimised. Upon further testing, I attempted to address the class imbalance issue using hybrid resampling. First, I undersampled the majority class (No) so that the minority-majority class ratio becomes 0.25 and then did SMOTE (Synthetic Minority Over-sampling Technique) oversampling on the minority class (Yes) to be 0.5 of the majority class (No). I left some imbalance as I still made use of class weights in my models to combine the effects of class weights and resampling. This influences the Gini impurity (my criterion for the tree-based models) calculations at each decision point as samples from the minority class contribute more to the impurity measure, making the model favour splits that better separate and identify minority class instances. Since, this is a medical context, I want to minimise false negatives and the class weights ensure that false negatives (missed "Yes", minority class) are caught more often.

Hyperparameter Tuning

To optimize model performance, I implemented nested stratified 3-fold cross-validation with grid search for hyperparameter tuning. I tuned the following hyperparameters for each model:

Gaussian Naive Bayes: tested '`var_smoothing`' values: $1e-4$, $1e-2$, 1

Decision Tree: tested '`max_depth`' values: 5, 10, 15

Random Forest: tested '`n_estimators`' values: 50, 100, 200, tested '`max_depth`' values: 5, 10, 15.

Each model was wrapped in a pipeline that included: Preprocessing via ColumnTransformer with StandardScaler for normalisation of numerical features. I combined 3 elements: class weights, random undersampling (minority class = 25% of majority) using RandomUnderSampler and SMOTE oversampling (minority class increased to 50% of majority) to synthetically generate minority samples. This avoids overfitting to patterns from synthetically generated data in minority class but still uses them to avoid wasting too much majority class data. The use of class weights also helps emphasise recall. The classifier (with hyperparameters tuned via grid search). Nested Cross-Validation Structure: Outer loop (5-fold Stratified CV): Evaluates generalization performance while the Inner loop (3-fold Stratified CV): Performs GridSearchCV for hyperparameter tuning. Scoring metric: F1-score (binary-positive), chosen to balance precision and recall in the presence of class imbalance. For each outer fold: The data was sampled to 1:2 ratio using SMOTE and undersampling (I had previously tested resampling to 1:1 ratio but it led to poor recall scores). A GridSearchCV was run on the training data using the inner folds. The scoring metric was F1 as although recall is the priority, F1 provides a good balance for tuning. The best model from grid search was used to make predictions on the held-out outer fold. Performance was tracked across accuracy, precision, recall, and F1-score. This allowed rigorous evaluation of models on unseen data, reliable selection of hyperparameters and fair comparison between model architectures.

Results and Discussion

Most frequent hyperparameters across all nested folds: Naive Bayes: $\text{var_smoothing}=1e-2$, Decision Tree: $\text{max_depth}=10$, Random Forest: $\text{max_depth}=15$, $\text{n_estimators}=200$, although the brief said to adjust hyperparameter ranges, I chose not to for random forest due to limited computational resources.

During preliminary testing, not using class weights or resampling yielded F1 and precision scores near zero for Decision Trees and Random Forest, confirming that addressing class imbalance (~91% 'No', ~9% 'Yes') is necessary, hence two strategies were explored: Class weights only and combined resampling and class weights, using a hybrid approach with RandomUnderSampler (to reduce 'No' class) followed by SMOTE (to synthetically boost 'Yes' class to 50% of majority), alongside class weights.

Model	Gaussian Naive Bayes	Decision Tree	Random Forest
Accuracy	85.93% \pm 0.15%	72.75% \pm 0.80%	82.11% \pm 0.15%
Precision ('Yes')	29.07% \pm 0.51%	21.04% \pm 0.44%	26.35% \pm 0.26%
Recall ('Yes')	41.75% \pm 0.79%	76.28% \pm 0.90%	57.73% \pm 0.65%
F1 Score ('Yes')	34.27% \pm 0.58%	32.98% \pm 0.52%	36.18% \pm 0.34%
F1 Score Weighted	87.04% \pm 0.13%	78.51% \pm 0.59%	84.90% \pm 0.11%

Table 1: Results using unsampled data, class weights for Decision Tree and Random Forest

Model	Gaussian Naive Bayes	Decision Tree	Random Forest
Accuracy	80.66% \pm 0.16%	77.30% \pm 0.54%	82.40% \pm 0.27%
Precision ('Yes')	24.83% \pm 0.26%	23.48% \pm 0.29%	27.56% \pm 0.40%
Recall ('Yes')	59.24% \pm 0.56%	70.08% \pm 1.45%	61.61% \pm 0.34%
F1 Score ('Yes')	34.99% \pm 0.34%	35.17% \pm 0.28%	38.08% \pm 0.41%
F1 Score Weighted	83.93% \pm 0.12%	81.75% \pm 0.37%	85.20% \pm 0.19%

Table 2: Results using hybrid resampling + class weights for Decision Tree and Random Forest

Without resampling, all models showed relatively high accuracy and F1 weighted scores due to the dominance of the majority class. However, these general metrics are not that useful because of the class imbalance and the priority for Heart Disease prediction is Recall, which can be dangerous if low as a diagnosis goes undetected.

Un-resampled Naive Bayes had surprisingly the best accuracy (85.93%) and precision but the worst recall (41.75%), likely because it did not address the class imbalance as it did not accept class weights and its independence assumption failed to capture complex interactions in the features. It can not be chosen as it would result in lots of false negatives which could be deadly. Un-resampled Decision Tree achieved the highest recall (76.28%) but had the lowest precision (21.04%), accuracy and F1 scores indicating a high false positive rate. This is likely because the class weights caused the model to overcompensate by classifying too many examples as positive as the splitting is skewed to capture the minority class. While this is technically the safest, it performs the worst in everything else and will be annoying to patients due to many false alarms. Un-resampled Random Forest delivers a balanced performance, with recall values and F1 noticeably better than the Un-resampled Naive Bayes while possessing better accuracy and F1 than the un-resampled Decision Tree. This is likely due to ensemble averaging and robustness to

overfitting compared to a single tree. Class weights help bias the model toward the minority class ('Yes'), but there is an issue of insufficient training examples for the minority, hence I did resampling to ensure the model sees more 'Yes' examples during training, improving generalisation for rare cases. I resampled to a 1:2 ratio, to avoid generating too many synthetic examples/undersampling too much. I also wanted to allow class weights to play a role in emphasising false negatives to improve recall. I had previously tested resampling to 1:1 ratio but it led to poorer recall scores although higher accuracy. Resampled Naive Bayes showed the greatest improvement in recall (from 41.75% to 59.24%), though at the expense of a drop in precision. This is deemed acceptable as recall is more important to me. The Resampled Decision Tree saw an improvement in accuracy, precision and F1 score, but a drop in recall. Even though it is our priority metric, it already had the best recall and it is still good as the second highest. It is a decent balance in safety and accuracy. Resampled Random Forest saw improvements in every metric, and it achieves the best balance between precision and recall. It has the 2nd best accuracy and precision but the 3rd best recall. Its ensemble nature and deeper trees allowed it to better model complex interactions while managing variance through bagging.

Un-resampled Naive Bayes had the fastest training time and simplest implementation, and it had the best accuracy and precision, but it has the poorest recall which is our most important metric so it is the most dangerous model. Un-resampled Decision tree has the best recall and is the safest model. It can be chosen if safety is the ultimate priority. The next safest would be the resampled decision tree which provides improvements in precision and accuracy but a drop in recall. Resampled Random Forest offers the best balance in both precision, accuracy and recall out of all models and it still has room to possibly improve as discussed in the conclusion.

Conclusion

In conclusion, a hybrid strategy of moderate resampling and class weighting proved most effective for training. This balanced approach helped models better recognise rare but critical 'Yes' cases, improving recall and F1 scores while maintaining reasonable overall accuracy. Among all models, resampled Random Forest with class weights provided the best trade-off between recall, precision and accuracy, making it a strong choice for this heart disease prediction task generally. My next choice would be the resampled decision tree as it had the 2nd best recall and better precision and accuracy than the un-resampled variant. However, if safety is the utmost concern, the un-resampled Decision Tree can be chosen for the strongest recall (76.28%).

Even with resampling and class weights, it's hard to make accurate positive predictions when the real-world data is heavily skewed, as seen by the low precision across all my models. Additionally, I could also increase the hyperparameters for Random Forest as the highest hyperparameter of 200 and max depth of 15 was selected in all 5 folds after tuning. I chose not to increase it any further due to additional training time, but this would likely help the model to improve and achieve even better recall. I could also explore XGBoost. Lastly, I also considered improving my models' performance with Recursive Feature Elimination (RFE) using Decision Tree Classifier to reduce the high dimension space and noise. However, it is computationally expensive and I would explore this as a future work.

References

- [1] WHO, "Cardiovascular Diseases," who.int, 2022. <https://www.who.int/health-topics/cardiovascular-diseases>
- [2] K. Pytlak, "Indicators of Heart Disease (2022 UPDATE)," Kaggle.com, 2022. <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease?select=2022> (accessed May 26, 2025).
- [3] D. Shah, S. Patel, and S. K. Bharti, "Heart Disease Prediction using Machine Learning Techniques," SN Computer Science, vol. 1, no. 6, Oct. 2020, doi: <https://doi.org/10.1007/s42979-020-00365-y>.
- [4] K. Drożdż et al., "Risk factors for cardiovascular disease in patients with metabolic-associated fatty liver disease: a machine learning approach," Cardiovascular Diabetology, vol. 21, no. 1, Nov. 2022, doi: <https://doi.org/10.1186/s12933-022-01672-9>.