A Comparison of Decision Tree and Random Forest for Predicting the Survival of Heart Failure Patients

1: Introduction

- This investigation used the Kaggle 'Heart Failure Prediction' dataset [**1**] and sought to compare the performance of Decision Tree (DT) and Random Forest (RF) for the binary classification task of predicting the survival of heart failure (HF) patients based on clinical features.
- Following this investigation, there was insight into the relative predictive power and limitations of using these methods on a class imbalanced dataset.
- This dataset was originally made available by Ahmed and colleagues in 2017 [**2**]. Soon after, Chicco and Jurman compared the ability of numerous machine learning algorithms to predict the survival of HF patients [**3**].
- Successful machine learning models can enable doctors to have predictions into whether a heart failure patient will die or not before being discharged; they can use this knowledge in combination with their expertise to decide the best course of action for a given HF patient.

2: Dataset and Exploratory Analysis

- This dataset had 12 attributes and only 299 observations; it is hard to asses the ability of a model to generalise when it is trained and tested using small datasets [**4**]. Overcoming this limitation was not within the scope of this investigation, however, a common solution in the medical domain is to generate synthetic data [**5**].
- If a patient died before being discharged, their death event (DE) = 1. If they were alive at discharge, their DE = 0.
- Out of the 299 observations, 32% were patients who died and 68% were patients who survived (**Figure 1**). However, as this investigation only sought to compare the relative performance of decision tree and random forest on an imbalanced dataset, the imbalance was only addressed when using performance metrics (**Section 5**).
- There were no missing values in this dataset.
- The 'time' attribute referred to the follow up period in days which is not known at the time a patient is admitted to hospital, therefore, it was removed for this investigation; enabling this model to be valid for newly admitted patients. Furthermore, **Figure 2** illustrates that time is the most correlated variable to the DE, as the DE is what impacts time, not the other way around.
- The 'high_blood_pressure' attribute is a binary feature, however, it was unclear what blood pressure value was used as the cutoff point to determine if a patient had high blood pressure. The lack of definition reduces reproducibility of future high blood pressure assignments for new patients, therefore, it was removed for this investigation.
- **Table 1a** and **1b** show statistical summaries for patients who survived and patients who died respectively. Differences can be seen between the two groups, for example, survivors had a greater mean ejection fraction.
- **Table 2** shows the percentage of survivors and percentage of those who died for each categorical attribute. There is no significant difference in percentage of survivors between each corresponding category, e.g. between males and females.

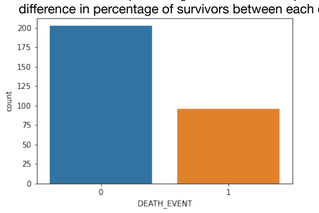


Figure 1.Comparison of the number of patients who died and who survived in HF dataset (0 = survived, 1 = died).

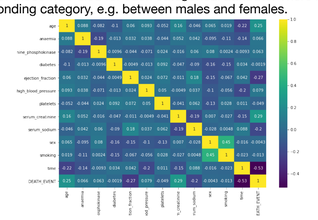


Figure 2. Heat-map showing correlation between all variables.

Table 1a. Statistical summary of numeric attributes for patients who survived.

	count	mean	std	min	max
age	203.0	58.761906	10.637890	40.0	90.0
creatinine_phosphokinase	203.0	540.054187	753.799572	30.0	5209.0
ejection_fraction	203.0	40.266010	10.859963	17.0	80.0
platelets	203.0	266657.489901	97531.202283	25100.0	850000.0
serum_creatinine	203.0	1.184877	0.654083	0.5	6.1
serum_sodium	203.0	137.216749	3.982923	113.0	148.0

Table 1b. Statistical summary of numeric attributes for patients who died.

	count	mean	std	min	max
age	96.0	65.215281	13.214556	42.0	95.0
creatinine_phosphokinase	96.0	670.197917	1316.580640	23.0	7861.0
ejection_fraction	96.0	33.468750	12.525303	14.0	70.0
platelets	96.0	256381.044792	98525.682856	47000.0	621000.0
serum_creatinine	96.0	1.835833	1.468562	0.6	9.4
serum_sodium	96.0	135.375000	5.001579	116.0	146.0

Table 2. Percentage of survivors and those who died in each category.

	Category Feature	Patients who survived / %	Patients who died / %
0	Had anaemia (1)	64	36
1	Didn't have anaemia (0)	71	29
2	Had diabetes (1)	68	32
3	Didn't have diabetes (0)	68	32
4	Male (1)	68	32
5	Female (0)	68	32
6	Smoked (1)	69	31
7	Didn't Smoke (0)	67	33

3: Decision Tree and Random Forest Overview

3.10: Decision Trees

- DTs can be used for making predictions in both classification and regression tasks by learning decision rules based on the data's features.
- They contain non-terminal nodes which test attributes and a search is carried out to determine which attribute split provides the greatest gain [**6**].
- This process is repeated until terminal nodes assign a value to the target variable.

3.11: Pros

- Decision trees are easy to interpret.
- Both categorical and numerical variables can be used.
- Relatively low data pre-processing required.

3.12: Cons

- DTs often overfit and consequently may not generalise well to unseen data.
- Minor differences in the dataset can cause large changes to the DT.
- When there are class imbalances, a DT biased to the majority class will form.
- The process of deciding which attributes to split can be computationally costly [**7**].

3.20: Random Forests

- RF is an ensemble method which uses multiple DTs to make predictions in both classification and regression tasks.
- Each tree is grown on a random sample of data with replacement, then aggregated together [**8**]. A chosen number of variables can be sampled for each split prior to working out gain. Bootstrap aggregation can reduce variance.
- For classification problems, each DT in the ensemble makes a class prediction; the class with the most DT 'votes' is the class assigned.
- RF generally has better performances on larger datasets.

3.21: Pros

- RF can reduce overfitting when compared to DTs.
- RF is less sensitive to outliers in training data than DTs.

3.22: Cons

- RF is less interpretable than DTs.
- RF can consume large amounts of memory when using large datasets.

4: Hypotheses

- Due to the class imbalance, both models will predict a greater proportion of the majority class (DE=0) correctly compared to the minority class (DE=1). RF will have better recall than DT.
- The RF will have a slightly better accuracy than the decision tree [**3**]. However, the small dataset means significantly greater accuracy might not be seen. Ali and colleagues only obtained greater accuracy for their random forest when they used a larger number of observations in their investigation (on another dataset) [**7**].
- The random forest will have a greater value for Mathews correlation coefficient, based on Chicco and Jurman's results [**3**].

5: Training and Evaluation Methodology

- The dataset was initially split into two sets; 70% of the data was assigned to be a training set, which would undergo further k-fold cross-validation when tuning hyper-parameters. The remaining 30% acted as a test set which remained unseen by the models until their final evaluations.
- Simply splitting the data into a single training and testing set can often give unrealistically good performance estimations [**9**], hence the use of k-fold cross-validation (k = 10).
- k-fold cross-validation made effective use of the limited data available in this investigation, where no external validation dataset was available. Once final model hyper-parameters were decided, the means of the cross-validated classification errors on the training set were recorded. Training and testing times were also recorded.
- Confusion matrixes were generated from final model testing and used to generate metrics to determine model performance.
- There was a significant class imbalance in this dataset (**Figure 1**), which therefore reduced the use of the accuracy metric. Accuracy doesn't discriminate between the number of correct predictions in each class, meaning a large proportion of misclassifications in the minority class can be overshadowed by a large proportion of correct predictions in the majority class.
- The metrics precision, recall, F1 score and MCC (**Figure 3**) were discussed in the model performance analysis. Precision is effective when the cost of a false positive is high, recall is effective when the cost of a false negatives is high and F1 is effective for imbalanced datasets when both false positive and false negatives are costly [**10**]. MCC is heavily influenced by performance in both class predictions. Domain knowledge assists in deciding which metric to value the most.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Precision = True Positive / (True Positive + False Positive)

Recall = True Positive / (True Positive + False Negative)

F1 Score = (2 x Precision x Recall)/ (Precision + Recall)

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$

MCC formula from: Medium. 2021. *Matthews Correlation Coefficient is The Best Classification Metric You've Never Heard Of*. [online] Available at: <https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the-matthews-correlation-coefficient-3bf50a2f3e9a> [Accessed 13 December 2021].

Figure 3. Performance metrics.

6: Hyper-Parameters and Experimental Results

- All hyper-parameter optimisation was done via grid-search method with the goal of minimising cross-validation classification loss.
- After each optimisation, the hyper-parameter values which gave the lowest losses were recorded and the optimisation process was repeated over a narrower range of hyper-parameter values. The new range was narrowed down around the previously recorded values. The process ended when loss values ceased to decrease.
- Feature selection was performed by referencing the p-values generated using binary logistic regression. Logistic regression was used for feature selection as it gave Chicco and Jurman their best performing model [**3**]. Statistically significant values ($p < 0.05$) were: ejection fraction, age and serum creatinine. The inclusion of ejection fraction and serum creatinine was consistent with Chicco and Jurman [**3**]. Removing redundant features can reduce computation time and improve learning accuracy [**11**].

6.1: Decision Tree Hyper-Parameters

- The maximum number of splits, minimum leaf size, and the minimum parent size influence the depth of a DT. Increasing the depth of trees can increase model complexity.
- The trees were split using the standard CART, which splits the predictor that gives the greatest split-criterion gain over all possible splits of all predictors [**12**]. Gini Diversity Index was the split criterion used.
- This investigation sought to optimise the maximum number of splits and minimum leaf size.
- **Figure 4a** shows lowest losses for optimisation 1 had a minimum leaf size of 21 and varying maximum number of splits values.
- The optimisation was repeated for only maximum number of splits (**Figure 4b**); 21 was the final value of minimum leaf size used for the DT. Multiple maximum number of splits gave the minimum loss, therefore, the smallest value of 4 was chosen, reducing the risk of overfitting to the training dataset at the cost of increasing bias, which was a favourable tradeoff. Overfitting would have been more costly given the small dataset size.

Figure 4a.

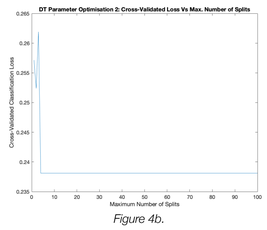


Figure 4b.

Figure 4. Decision tree hyper-parameter optimisation 1 and 2 visualisations.

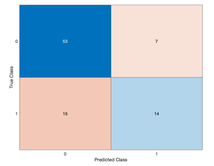
6.2: Random Forest Hyper-Parameters

- The DTs in RF can also have their depth and complexity varied.
- The ensemble aggregation method used was 'Bag', and the number of variables sampled for each split was 3.
- The maximum number of splits, minimum leaf size and the number of learning cycles were optimised, aiming to gain control over the number of DTs and the complexity of each tree. Each optimisation searched a progressively narrower range of hyper-parameter values.
- The smallest loss values was observed in optimisation 2, using 53 learning cycles, a minimum leaf size of 7, and a maximum number of splits of 75. Therefore, these hyper-parameters were used (**Table 3**).

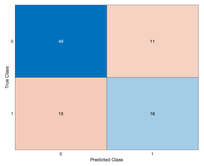
Table 3. Table showing minimum loss values achieved in each RF hyper-parameter optimisation run.

Optimisation Run	Number of Learning Cycles	Minimum Leaf Size	Maximum Number of Splits	Minimum Cross-Validated Classification Loss in Run
1	136	5	35	0.23333
2	53	7	75	0.22857
3	43	4	80	0.23810

6.3: Results



Decision Tree



Random Forest

Metric	Decision Tree	Random Forest
Mean cross-validated classification error on training set.	0.2381	0.2571
Accuracy	0.75	0.73
Precision	0.67	0.59
Recall	0.48	0.55
F1	0.56	0.57
MCC	0.40	0.38
Model Execution Time on Test Set / seconds	0.001103	0.016530
Final Training Time/ seconds	0.015483	0.346420

	Final Parameters Used		
Model	Minimum Leaf Size	Maximum Number of Splits	Number of Learning Cycles
Decision Tree	21	4	N/A
Random Forest	7	75	53

Figure 5. Confusion matrixes produced following testing on the hold-out set for final DT and RF models and tables showing performance metrics and final hyper-parameters used.

7: Analysis and Critical Evaluation of Results

- Models trained using only this dataset have no experience with young patients as all patients were above the age of 40; if there are different relationships between the explanatory variables and response variables for patients under the age of 40, this model would be unreliable.
- Unexpectedly, the DT had a lower mean cross-validated training error than the RF. The minimum leaf size was significantly higher and maximum split size was significantly lower for the DT compared to the RF (**Figure 5**), suggesting the DT was more robust and generalisable to the whole training set during the cross validation, despite only being one tree. This indicates the effect of having more trees in RF was not significant enough to overcome the overfitting caused by each tree's greater tree depth.
- Moreover, the DT had slightly greater accuracy and precision than the RF, potentially also due to the small dataset size. It is likely the RF would have outperformed the DT on accuracy if the dataset was larger; for example, Ali and colleagues found increasing the number of instances from 286 to 699 increased the percentage of correctly classified instances by the RF from 69.23% to 96.13% for their classification problem [**7**].
- As the dataset had a large class imbalance (**Figure 1**), with the DE = 0 being the majority class, the correct predictions of the majority class overshadowed the poor minority class predictions when using accuracy. Therefore, MCC, recall and F1 score were used to give more insight to model performances, particularly their ability to identify the minority class.
- As originally expected, RF had a greater recall and F1 score than DT and predicted a greater proportion of the minority class correctly. Some trees may have been effective at identifying DE = 1, whereas the DT was heavily biased to the majority class (DE = 0).
- A solution to the class imbalance issue is Synthetic Minority Over-sampling Technique (SMOTE). Variations of SMOTE attempt to address class imbalances in different ways; an example is over-sampling the minority class by creating synthetic observations [**13**].
- When optimising hyper-parameters, RF took a significantly longer time to optimise than DT, even when all hyper-parameter optimisation options were kept the same. Furthermore, RF had an additional hyper-parameter to optimise. To save time, a lower number of grid divisions were used for RF. This led to less values for each hyper-parameter being searched compared to the DT optimisation. Insufficient optimisation compared to the DT is another possible explanation for the unexpected lower performance of RF in accuracy, precision and MCC.
- Feature selection was performed with reference to p-values. Values close to 0 meant the feature was closely correlated to death [**3**]. The p-values generated were demonstrating the importance of features to the DE in a binary logistic regression model, however, Chicco and Jurman found several machine learning algorithms gave similar p-values results when selecting features for this dataset [**3**] , reducing the importance of feature selection algorithm choice.
- The MCC obtained by the DT (0.40) was greater than the MCC Chicco and Jurman achieved (0.376), showing improved performance [**3**]. Unexpectedly, the MCC for this investigation's DT was high than for the RF, again likely due to the small dataset and overfitting of the RF trees (the overfitting was not adequately compensated by having an increased number of trees).
- The RF had a lower MCC than achieved by Chicco and Jurman when they only used ejection fraction and serum creatinine, suggesting the inclusion of age in this investigation was damaging to the model performance.
- The model execution and training times were much greater for RF than DT (**Figure 5**), due to the computation of many more DTs. However, all times were less than 1 second long, thus, the additional time of RF was negligible in this investigation.
- Standard CART predictor-splitting algorithm was used, however, this favours continuous variables. Curvature and interactions tests are alternatives.
- If the doctors are concerned with overall performance, they should use the DT which gave greater accuracy and MCC. However, if false negatives are more costly, they should refer to the RF which had a greater recall and F1.

8: Future Work and Lessons Learned

- During this investigation, a notable performance limitation came from the imbalanced dataset. Therefore, future work should consider the use of SMOTE technique in order to increase the proportion true positive values (the minority class) obtained.
- Given more time, it would insightful to attempt different feature selection techniques and test the subsequent models. For example, Chicco and Jurman found their best performing model to only use two features (ejection fraction and serum creatinine).
- When optimising the RF hyper-parameters, it would allow more valid comparisons of results if the same number of grid divisions was used as during DT hyper-parameter optimisation.

9: References

[1]"Heart Failure Prediction", *Kaggle.com*, 2021. [Online]. Available: https://www.kaggle.com/andrewmvd/heart-failure-clinical-data/code. [Accessed: 04- Dec- 2021].

[2]T. Ahmad, A. Munir, S. Bhatti, M. Aftab and M. Raza, "Survival analysis of heart failure patients: A case study", *PLOS ONE*, vol. 12, no. 7, p. e0181001, 2017. Available: 10.1371/journal.pone.0181001.

[3]D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone", *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, 2020. Available: 10.1186/s12911-020-1023-5.

[4]T. Shaikhina and N. Khovanova, "Handling limited datasets with neural networks in medical applications: A small-data approach", *Artificial Intelligence in Medicine*, vol. 75, pp. 51-63, 2017. Available: 10.1016/j.artmed.2016.12.003.

[5]A.Sabay, L.Harris, V.Bejugama, and K.Jaceldo-Siegl, "Overcoming Small Data Limitations in Heart Disease Prediction by Using Surrogate Data" ,*SMU Data Science Review*: Vol. 1 : No. 3 , Article 12, 2018. Available: https://scholar.smu.edu/datasciencereview/vol1/iss3/12.

[6]S. Kotsiantis, "Decision trees: a recent overview", *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261-283, 2011. Available: 10.1007/s10462-011-9272-4.

[7]J. Ali, R. Khan, N. Ahmad, and I. Maqsood, " Random Forests and Decision Trees ", *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 3, Sep 2012 .

[8] G. Biau and E. Scornet, "A random forest guided tour", *TEST*, vol. 25, no. 2, pp. 197-227, 2016. Available: 10.1007/s11749-016-0481-7.

[9] C. An, Y. Park, S. Ahn, K. Han, H. Kim and S. Lee, "Radiomics machine learning study with a small sample size: Single random training-test set split may lead to unreliable results", *PLOS ONE*, vol. 16, no. 8, p. e0256152, 2021. Available: 10.1371/journal.pone.0256152.

[10]"Accuracy, Precision, Recall or F1?", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. [Accessed: 12- Dec- 2021].

[11]J. Cai, J. Luo, S. Wang and S. Yang, "Feature selection in machine learning: A new perspective", *Neurocomputing*, vol. 300, pp. 70-79, 2018. Available: 10.1016/j.neucom.2017.11.077.

[12]"fitctree", *Mathworks.com,* 2021. [Online]. Available https://www.mathworks.com/help/stats/fitctree.html. [Accessed: 04- Dec- 2021].

[13] N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence*