

Predicting Mortality in Heart Failure Patients

ECS 171, Spring Quarter 2025

Anna Ayers

*Department of Computer Science
University of California, Davis
anmayers@ucdavis.edu*

Daniel Chu

*UC Davis Cognitive Science
University of California, Davis
dcychu@ucdavis.edu*

Amabel Gale

*Department of Computer Science
University of California, Davis
avgale@ucdavis.edu*

Grace Lim

*UC Davis Cognitive Science
University of California, Davis
grhlim@ucdavis.edu*

Sean Weber

*Department of Computer Science
University of California, Davis
sdweber@ucdavis.edu*

***Abstract*—Heart failure survivors are at risk of further health complications after their initial onset of symptoms, and predicting the likelihood of severe complications and/or death is difficult. To tackle this issue, we employed four different machine learning algorithms to predict survival in heart failure patients after their initial release from hospitalization. All models were trained on two features: levels of creatinine in the patients’ blood and left ventricular ejection fraction. Our Bayesian network outperformed all other models: Linear Regression, Logistic Regression, and Artificial Neural Network (ANN).**

I. INTRODUCTION & BACKGROUND

Heart failure is a leading cause of death worldwide, caused by structural or functional disorders of the heart [4]. More specifically, it is typically caused as a result of other conditions and symptoms such as high blood pressure, diabetes, and heart disease [1]. In the US alone, around 6.7 million Americans over the age of 20 suffer from heart failure [5]. This number is expected to increase by around 2 million in only 5 years [5]. Patients who have experienced heart failure are at an increased risk of health complications requiring hospitalization after the initial event, some of which may even result in death. The incurable nature of the condition leaves patients with an unpredictable clinical journey [4]. As a result of this, accurately assessing survival in the period following heart failure is a challenging task for medical professionals. When diagnosed, heart failure can carry significant risk to the patient’s health, as around 38% of those diagnosed will die within 1 year of this event. In the following 5 years, this number increases to around a 60% chance of death [4].

The multitude of highly dimensional variables being monitored makes it difficult to uncover correlations between patient data and full recoveries. However, machine learning has the potential to shed light on the underlying patterns associated with hospital readmission after heart failure. It is a particularly useful tool in this application, given that it can handle nonlinear relationships within the data and detect these subtle patterns across a large dataset. Predicting survival via machine learning could help identify key indicators of persistent cardiovascular issues and provide a more reliable assessment of a patient’s risk of death.

Our approach included building and training four different models that use different techniques to predict patient survival. The models included a Linear Regression model, Logistic Regression model, Naïve Bayesian network, and an ANN. We performed data analysis and a literature review to determine which features to train our models on. From these models, we compared their performances to find the best trade-off between precision and specificity. These performance measures were evaluated alongside accuracy and F1 scores to find the best possible model.

In previous studies, high accuracies have been achieved using the above models for classification of heart failure outcomes [2]. However, a possible limitation that arises is that some of these models have been trained on electronic health records from single hospitals or institutions, lowering the generalizability to other patient populations. As a result, this current study aims to evaluate the performance of these different models used on a publicly available dataset, in order to assess

their generalizability and robustness in a different clinical setting.

Through this study, we hope to identify the most effective model type for heart failure patient data, in addition to the most conducive features for prediction. The intended purpose of this study is to provide medical professionals with insight regarding patients that may be at risk of death based on levels of creatinine in the patients' blood and left ventricular ejection fraction.

II. DATASET DESCRIPTION

The dataset we are using is a collection of medical records of 2008 patients, from a hospital in Sichuan, China, that have experienced heart failure and were subsequently discharged from the hospital. A similar dataset was used in [1], but due to the limited number of patients and features, we chose to use the dataset discussed in [3]. The data included for each patient span a large number of bodily functions, demographic information, and other medical measurements; in total there are 168 features. Some patients were readmitted, and a subset of these people died within six months of the initial heart failure event. The records span 2016-2019.

It was discovered during exploratory data analysis that some patient records were incomplete and did not include the 5 features of interest. As such, we filtered the dataset to include only patients with complete medical records of the relevant features, resulting in a subset dataset of 621 patients. Additionally, in the original dataset, death was separated into three categories: death within 28 days, death within 3 months, and death within 6 months. Given that our models predict mortality outcomes as a binary output instead of multiple time frames, we aggregated the death records into a comprehensive column. This is the dataset we proceeded with for exploratory data analysis, feature selection, and training/testing.

The original dataset had a significant imbalance between patients who survived and those that did not. There were 57 deaths out of the 2008 patient records, which - given death is positive class for our subsequent model predictions - corresponds to 2.84% positives and 97.16% negatives. This issue was further exacerbated in our filtered dataset. Only 10 deaths remained, yielding 1.61% positives and 98.39% negatives.

III. LITERATURE REVIEW & PROPOSED METHODOLOGY

Our review of Chicco and Jurman's paper [1] established *left ventricular ejection fraction (LVEF)* and

serum creatinine as the top features most strongly correlated with death after heart failure, and led to the most accurate model predictions of mortality. Ejection fraction describes the percentage of blood, in a single contraction, that is being pumped by the heart. More specifically, heart failure can be marked by dysfunction in the left ventricle, leading to an ejection fraction of less than 40% [1]. When muscles break down creatine phosphate during normal metabolism, creatinine is produced as a byproduct and released into the bloodstream, contributing to serum creatinine levels. High amounts of creatinine may be an indicator of renal failure [1]. Close contenders for other possible features were age, serum sodium and sex of the patient, in that order. Like the paper [1], we will only choose 2 of the top features since the existing research suggests that this amount is sufficient. We aim to evaluate the validity of this claim, so keeping the number of features low is important to avoid overusing features and unnecessarily increasing complexity of our models.

Chicco and Jurman's paper [1] evaluates several machine learning models for predicting heart failure patient survival. Of the models they chose, they indicated that the Decision Tree models consistently outperformed other methods, which is confirmed in Wang's paper [2]. The former paper indicates that they also made other models, and three models that stood out were Linear Regression, ANN, and the Bayesian Network.

All of these models are discussed in the context of this class as well as evaluated in the paper, and as such, these models were selected for development. This is partially due to the fact that the team is more familiar with the models, and also to evaluate more thoroughly the models that the paper focused less on. Although Decision Trees were discussed in class, we had very little hands-on experience with making such models, which is another reason we did not choose this model over the three we did choose. With respect to hyperparameter tuning, this was performed on the ANN Model, as Regression and Bayesian Networks had no hyperparameter tuning. According to Chicco and Jurman's paper [1], they used grid search with the Matthews correlation coefficient to identify the best hyperparameters. We will also apply grid search; we will be using the batch size, tolerance, activation function, momentum term, and the number of nodes at each hidden layer. This is done to test variations of the model, and thereby evaluate which parameter values yield the strongest performance. This adds depth to the current model discussed by Chicco and Jurman's paper, as the paper does not describe the hyperparameters

that they used for grid search tuning, only that they had used the Matthews correlation coefficient.

For the ANN model, we will be using Stochastic Gradient Descent, which we chose over other methods since there are very few deaths, and allowing more robust changes to the parameters based on individual data points will help the model be influenced more significantly by the death outcomes in the data. This is unlike Batch Gradient Descent which would most likely overwhelm the death outcomes with survival outcomes and make the model change insignificantly.

The final consideration for the model development is the division of training and testing data. According to Chicco and Jurman's paper [1], they used an 80:20 train-test split for the Linear Regression and Naïve Bayes models. For Linear Regression, we will instead use a 75:25 train-test split to better even out the data to get some death reports in the test data. To evaluate the outcome of the Naïve Bayes model within the bounds of the reported conclusions, we will also be using this same division of training and testing data.

Finally, for the ANN divisions, the paper includes a third data portion: the validation data. This is used iteratively during training in order to more accurately tune the model, and goes deeper than the test data which serves to evaluate the accuracy of a completed model for a training set of data. The divisions of these sets are 60% training, 20% validation, and 20% test [1]. Since our models will be using the test data and the validation data as the same set, we are going to use 75% training data and 25% test/validation data for model training, especially considering that there are few deaths reported in the model. Ensuring that the training data contains these death values is important, and so we chose to use a larger training data percentage so that more deaths are included in the data.

Potential problems with this method are that the test data could have less deaths, meaning that there is a likelihood that the model may ignore death predictions entirely due to the lack of deaths in the test data. Other potential options are manually placing death cases into the data, however, this could potentially lead the model to overfit the data and effectively only identify the death cases shown. The goal of this model is to be able to accurately predict death cases without oversimplification, so choosing solely to increase the training size allows us to generally include death cases without causing the model to overfit. The key challenge with this model is that the accuracy will naturally be quite high, as there is a significant number of survivors, so the model

may focus too much on predicting survival and fail to predict death. To avoid this, we will include important hyperparameters, most notably the tolerance, in order to ensure that the model does not halt too early and instead continues to test until it hits the tolerance threshold. With this in mind, the tolerance thresholds will be as low as possible during grid search so that it is more likely for the model to continue to test on data to hopefully become more accurate and properly predict deaths.

Another thing to consider is the complexity of the model. As another hyperparameter, the complexity of how many hidden layers there are and the nodes per layer will help with identifying the most optimal model with respect to complexity. The hope is that the complexity of the model will ensure that no underfitting is done since more complex models will be more likely to identify the complexity of the death data points among the survival points. The issue that can occur is that the sheer quantity of survival data points could overwhelm the models and their respective complexities in the number of hidden layers and number of nodes, and instead still default on a simplified model due to its high accuracy.

For the regression models, the same train-test split of 75:25, respectively, was used for consistency across models to reduce any confounds in comparing the results. In addition, a randomizer was used to ensure reproducibility given the skewed dataset. In the Logistic Regression model, this was done by setting the random state to 0 in the classifier's parameters. Some optimization can be done with regards to the threshold used for a Linear Regression model that makes binary predictions. The decision boundary can be shifted in order to increase the number of predictions of a certain value – in our case, whether we decide a death or life prediction is more important.

The performance metrics used in [1] besides MCC, were: F1 score, accuracy, TP rate, TN rate, Precision-Recall Area Under the Curve, and Receiver Operating Characteristic Area Under the Curve. As such, we incorporated these metrics into comparing our models to each other and to [1]. Since we are performing a binary classification, we did not calculate MSE, R2, and MAE, which are typically used for regression based models.

In the Naïve Bayes model, our team selected the Gaussian variant because of the continuous nature of the clinical features used. The probabilistic algorithm works under the assumption of feature independence and uses the Bayes' Theorem to effectively calculate the probability of class membership. While this assumption may not always hold for all features in a common

clinical dataset, previous research [6] has shown that Naïve Bayes model can still perform well in practice - specifically on smaller or imbalanced datasets like ours.

A reason for including Naïve Bayes model is its computational simplicity and transparency, which are both advantageous when interpreting results in a medical context. Additionally, Chicco and Jurman's paper [1] included this model but focused less on its implementation details or feature impact. So, we took the opportunity to explore it further and analyze its strengths and weaknesses in predicting mortality.

As the Naïve Bayes model is known to be sensitive to skewed class distributions, our team also paid attention to class balance when training the model. To evaluate its effectiveness under imbalanced conditions, we used the same 75:25 training split as the other models to ensure that they are comparable. Stratified sampling was used to maintain class proportions across the split. We also compared the baseline model to a version with adjusted prior probabilities to see whether reweighting the death class would improve recall and F1 score.

Finally, the Naïve Bayes model was evaluated using the same performance metrics as the other models, including accuracy, F1 score, precision, and recall. This was done with a specific focus on its ability to identify death. The goal for our team was not necessarily to outperform ANN or Logistic Regression in all metrics. Instead, it was to see whether Naïve Bayes offers a lightweight, interpretable alternative with strong class separation for high-risk patients, even with fewer tuning parameters and assumptions.

IV. EXPLORATORY DATA ANALYSIS & FEATURE SELECTION

Our dataset includes left ventricular ejection fraction, measured as a continuous variable (with the normal range defined). Due to the high correlation between death and LVEF as analyzed in [1], we selected this as one of the features to train our upcoming models with.

Selecting a second feature involved a number of possibilities, given that there were multiple variables that could potentially relate to or substitute serum creatinine due to its indication of kidney function. Our dataset includes acute renal failure (defined by an increase in serum creatinine within one of three specified ranges over time), moderate to severe chronic kidney disease, and creatinine levels in the blood i.e. serum creatinine.

It was determined that none of the patients in the filtered dataset had acute renal failure, so we immediately discarded this as a potential feature. To determine

whether moderate to severe chronic kidney disease or creatinine would be a more useful feature to use, we conducted statistical analyses to find the correlation between these variables and death. Given that both death and moderate to severe chronic kidney disease are recorded as binary variables, we calculated the Phi coefficient. This established a slight positive correlation of 0.0827, with a p value of 0.0395, implying that there was a significant correlation albeit weak. On the other hand, creatinine using the enzymatic method is a continuous measurement, so we calculated the point-biserial correlation coefficient. This resulted in a slightly stronger positive correlation of 0.163, with a p value of 0.0000469. This suggested that creatinine would be a better feature to use to train our model, instead of chronic kidney disease.

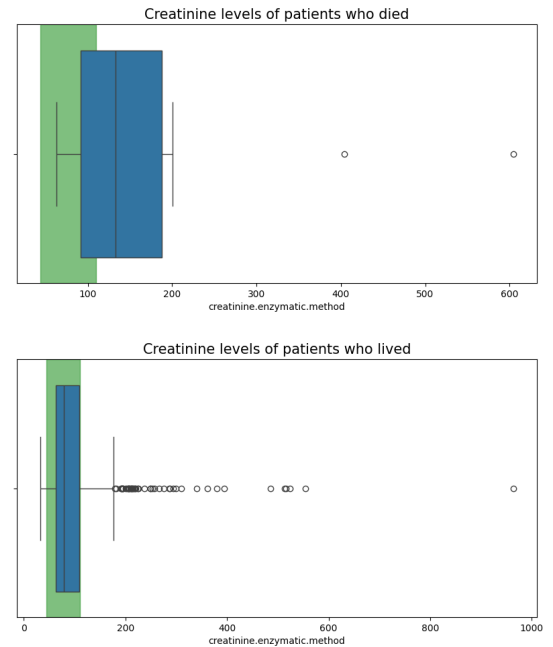


Fig. 1: The boxplots above compare the creatinine levels of patients who lived and died. The normal range for creatinine in the blood, 44–110 μmol/L [3], is highlighted in green. The majority of patients who lived (72.8%) had creatinine levels within the normal bounds, while the majority of patients who died (60%) had abnormal creatinine levels.

Given that there were several top-ranked features beyond LVEF and serum creatinine in [1], we took a look at the next highest ranked features. We chose not to include age due to the lack of specificity in our dataset, which had age measured by the decade. The next ranked feature was sodium, for which we tested its correlation to

death via the point-biserial correlation coefficient. This showed a slight negative correlation of -0.0612 , with a p value of 0.127 . Given these results, creatinine remained the strongest contender, and so we selected this as our second feature.

Descriptive statistics were calculated along with visualizations in order to determine the need for standardization. The observed range of creatinine values was from 32.7 to 963.4 $\mu\text{mol/L}$, with a mean of 100.18 $\mu\text{mol/L}$ and a median of 79.6 $\mu\text{mol/L}$. For reference, the normal range is reported to be 44 – 110 $\mu\text{mol/L}$ [3]. LVEF values ranged from 5.0% to 82.0% , with a mean of 50.51% and a median of 51.0% . The reference range for normal LVEF is 55% – 70% [3]. Based on these visualizations, we standardized creatinine, through z-score standardization.

V. EXPERIMENTAL RESULTS & EVALUATION

The comparative performance metrics can be seen in Figure 3.

The ANN and Logistic Regression performed similarly across all performance metrics, with the exception of F1 score, where Logistic Regression performed better than the ANN. Logistic Regression used a train-test split of $80:20$, while ANN used $75:25$. Our ANN used the ReLU activation function and a simple structure with three hidden layers that contained 2, 4, and 4 neurons respectively. This model architecture was determined by performing grid search, through which we tuned momentum, the number of hidden layers, batch size, stopping criteria, and the activation function. The grid search model prediction identified the lowest possible values for each parameter option in batch size, number of hidden layers, and momentum, indicating a simplified model.

Though boasting a high accuracy of 99.4% for ANN and 98.3% for Logistic Regression, both models performed poorly across other evaluation metrics. In the case of ANN, precision, recall and F1 score were all 0. For Logistic Regression, the only difference was achieving a higher, but still under chance level F1 score of $\sim 50\%$. In both cases, we deduced that this was due to a tendency to over-predict survival - no deaths were predicted, but due to the dataset's heavy skew towards survival, making it unlikely for death to occur already, the accuracy remained high. For ANN, looking at how the model used low values for the number of hidden layers and batch size, it is clear that this model was incredibly simple, but overly so. Given the sensitive medical context of patient mortality prediction, under-predicting death poses a particularly high risk. Erring

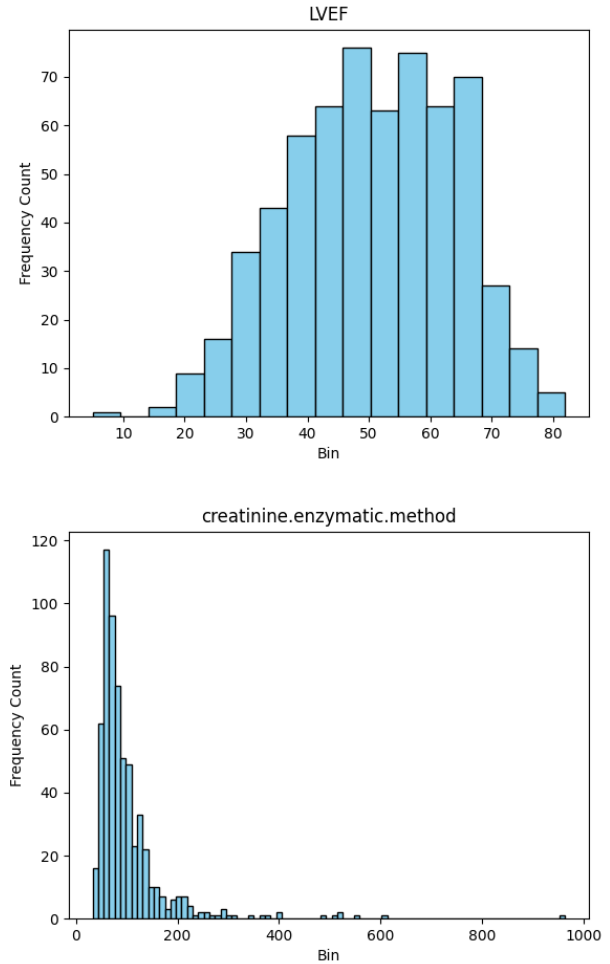


Fig. 2: Histograms of LVEF measured in % (above) and creatinine.enzymatic.method (below) measured in $\mu\text{mol/L}$. In the histograms above, we can see that LVEF is relatively normally distributed and symmetric, with an outlier at 5% and creatinine is positively skewed with a few outliers from 200 – $1000\mu\text{mol/L}$.

on the side of caution and over-predicting mortality as opposed to under-predicting might be more desirable, given that it could encourage high levels of caution and care to mitigate death. In essence, false negatives (incorrectly identifying survival) is particularly dangerous, and should be avoided. However, all incorrect predictions were false negatives, which is why the F1 score and recall were 0. These values need to be high, as they are effectively what identifies how good our model is when false negatives are so detrimental, given the context of predicting death.

Taking this and the full evaluation criteria into consideration, the ANN was our weakest model, alongside

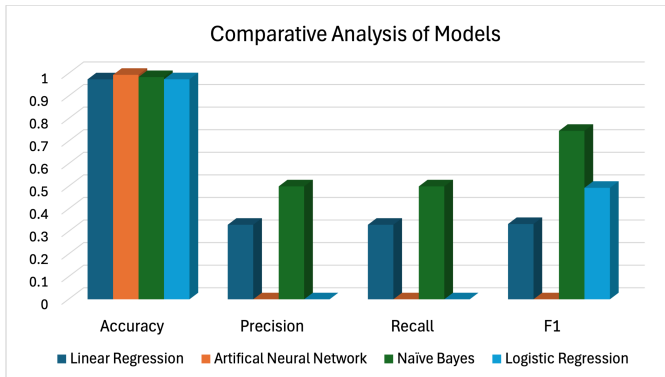


Fig. 3: The bar graph above shows a comparison of evaluation metrics for the Linear Regression model, Artificial Neural Network, Naïve Bayes model, and the Logistic Regression model. The Naïve Bayes model had the highest precision, recall, and F1 scores of 0.5, 0.5, and 0.7459, respectively. The Artificial Neural Network had the highest accuracy overall, of 0.994.

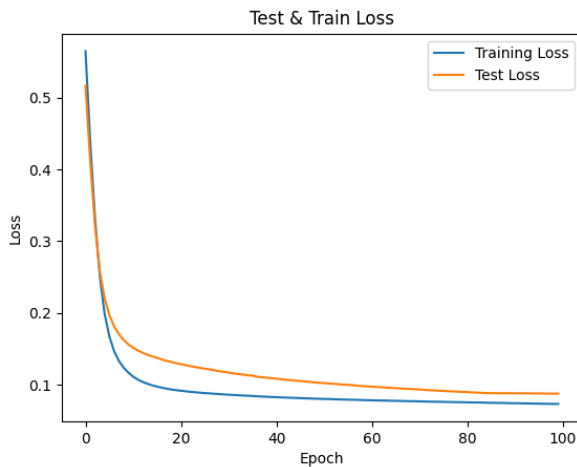


Fig. 4: The graph above shows the change in loss over the epochs for the Artificial Neural Network. This loss gets minimized to <0.1 at around 30 epochs, and gets close to a 0.5 loss by the 100th epoch. This indicates that the model is effective at minimizing the loss function. However, due to the few quantities of deaths reported, we can see this graph does not properly identify the low amount of loss required to accurately predict deaths.

Logistic Regression. As discussed, erring on the side of caution is an idea that can be implemented into these models, and further development would include this to help with properly identifying deaths by rewarding models for more death predictions, or by penalizing false negatives more significantly in the loss function. Furthermore, the large size of the data with the low amount of deaths recorded made it very difficult to identify the patterns of deaths. In the grid search optimization code, attempts at lowering the tolerance (the point at which the model halts) to try to make the model more accurate led to exceeded training tests. This caused the model training to fail, which meant that achieving this level of accuracy was incredibly difficult for the model to identify.

Furthermore, as Fig 4 describes in the loss graph, the loss is already incredibly low, less than 0.1 by 30 epochs, which further enforces the idea that the model's low death count was insufficient to make a meaningful impact on the ANN. This leads to the conclusion that the model chose instead to take a simpler model with a decently high accuracy, but in the end this accuracy was too tolerant and led to improperly predicting deaths.

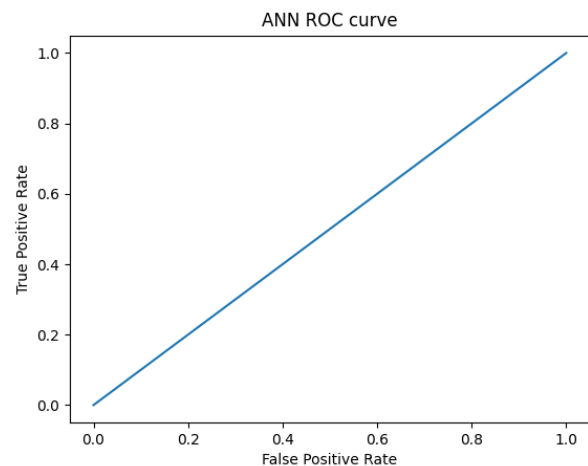


Fig. 5: The ROC curve above shows the poor classification abilities of the ANN. It performed no better than a random classifier, since all patients were predicted to survive.

Our second-best performing model for predicting death after heart failure used Linear Regression. The data had a 75:25 train-test split and used stratified sampling in order to ensure both sets had similar proportions of data. Due to the high prevalence of negative death values, a threshold of 0.06 – as opposed to the more common 0.5 – was used to predict whether a patient would

die. This led to a higher number of false positives, or predicting a patient would die when they would instead live, which we determined to be a necessary trade-off when considering the cost of “missing” a prediction of death. The accuracy was 0.974, which was largely influenced by the high prevalence of survival in our dataset. The mean squared error was 0.018 and the mean absolute error was 0.0325. While less relevant when a Linear Regression model is being used to classify binary data, they still show an overall low presence of errors. Precision was 0.33, recall was 0.33, and the F1 score was 0.33. These scores show that the Linear Regression model is indeed being impacted by the imbalanced data, and isn’t good at predicting the smaller class of deaths.

Overall, the Bayesian model yielded the strongest performance. A key driver of these results is the pronounced class imbalance: survival cases outnumber deaths by roughly 8:1. Such skew inflates overall accuracy - models that always predict “survival” exceed 88% accuracy, but masks poor performance on the minority “death” class. Logistic Regression and the ANN defaulted to predicting no deaths, yielding high accuracy but zero recall and precision score; Linear Regression improved recall only at the expense of precision. In contrast, the Bayesian network’s explicit modeling of conditional dependencies between creatinine and LVEF helped it detect 70% of death cases ($F1 = 0.53$, $AUC = 0.81$). These findings highlight the necessity of both reporting and addressing unevenness in medical datasets. Going forward incorporating resampling strategies, cost-sensitive loss functions, or threshold adjustments will be critical to boost minority class detection without sacrificing specificity.

Based on these model’s performance, we compared the linear separability of our two features with Chicco and Jurman [1] as a possible explanation for this difference.

In Chicco and Jurman [1], the linear separability between serum creatinine and ejection fraction was far more distinct between the mortality class. By computing the scatterplot seen in Figure 6, it provided evidence for the predictive power of both features because of the separability. In the current study, there is far less of this separability, which could be due to the lack of deaths in the dataset compared to survivors.

In addition to comparing the feature separability with Chicco and Jurman [1], we also compared the relationship between follow-up time, or readmission in our dataset, to survival. In their paper, this was explored to rule out the possibility of follow-up visits relating to the chances of survival [1].

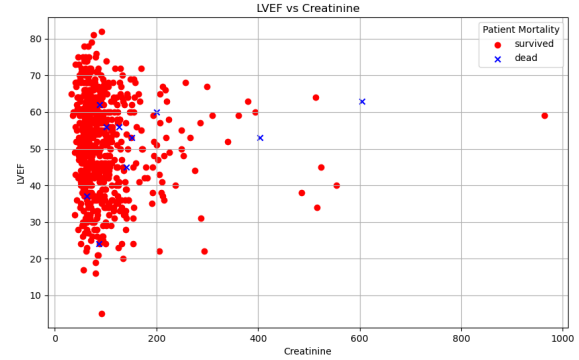


Fig. 6: Scatterplot of LVEF and creatinine values in relation to mortality class.

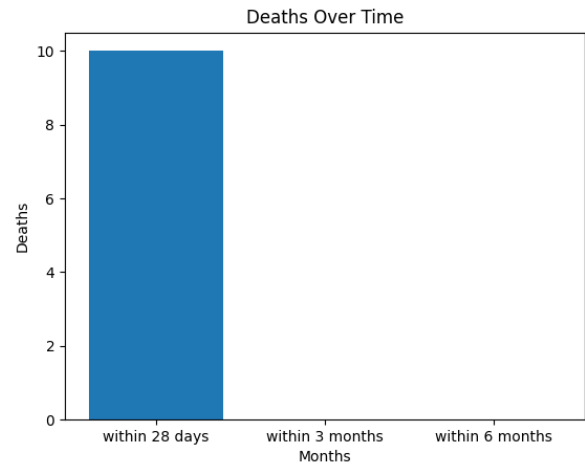


Fig. 7: Bar chart of deaths following initial hospital visits, in bins of months. These deaths represent the total amount of deaths of all patients in the dataset.

As seen in Fig. 7, all deaths occurred in month 0 (within 28 days). Due to this, there was no clear effect of follow-up visits or readmissions on the survival of the patient. In Chicco and Jurman [1], there was no trend showing an increase or decrease in survival over months in relation to follow-up visits. In this regard, the current study aligns with these findings from previous literature.

VI. CONCLUSION & DISCUSSION

In our project, we investigated the effectiveness of four machine learning models which includes Linear Regression, Logistic Regression, Bayesian network, and an Artificial Neural Network. We used these models for predicting death in heart failure patients using two key features: serum creatinine and left ventricular ejection

fraction (LVEF). Our team's goal was to ultimately see whether a model could still reliably set apart survivors and non survivors within the given dataset. In doing so, this provides a generalizable and interpretable decision support tool. Upon analyzing our evaluation metrics which includes accuracy, precision, recall, and F1 score, the Bayesian network outperformed all of the other machine learning models.. This shows the Bayesian network's ability to correctly identify death cases despite the dataset's imbalance.

Some limitations of our approach call for discussion. One of those limitations is because of the extreme class imbalance in both the original dataset (57 deaths of 2,008 patients) and our filtered subset (10 deaths of 621 patients). These distributions inherently challenge any binary classification model. Technically, it led the ANN and Logistic Regression to adopt trivial strategies, achieving high accuracy but zero clinical utility. Although we partially addressed this imbalance by increasing the training data proportion to 75% and by lowering classification thresholds in the Linear Regression model, these measures alone were unfortunately not enough to bring the non-Bayesian models to acceptable performance levels. Future work should explore advanced resampling techniques, cost sensitive learning, and better ensemble approaches.

Another limitation comes about from our intentionally narrow feature set. While prior literature (Chicco and Jurman [1], Wang et al. [2]) has demonstrated that creatinine and LVEF are among the strongest predictors of mortality in heart failure, focusing solely on these two biomarkers may overlook additional patient characteristics such as age, serum sodium, comorbid conditions, medication adherence, and socioeconomic factors. This could further refine risk stratification. Chicco and Jurman's use of larger, institution-specific electronic health record data sets indeed yielded slightly different feature rankings in some cases. By restricting our model to only two inputs, we are able to prioritize interpretability and generalizability at the potential expense of maximal predictive power. Future research could look into how the inclusion of a broader set of features interacts with model type, specifically in terms of how additional variables may mitigate class imbalance by introducing more nuanced variance between survivors and non survivors.

Our work highlights the importance of rigorous handling of class imbalance in machine learning. A model's accuracy in itself is insufficient. Metrics such as recall, precision, F1, and AUC should be evaluated with emphasis on the minority class. Clinicians are most

concerned with minimizing false negatives, and thus models must be explicitly tuned to optimize for such outcomes. Beyond resampling and cost sensitive methods, threshold tuning can be a simple yet effective strategy, particularly when complemented by clinician input to calibrate acceptable trade offs between false positives and false negatives.

Ultimately, our team recognizes that our machine learning model represents only a piece of the larger puzzle in heart failure management. Predicting high risk patients is useful only if it leads to successful interventions. To successfully help a patient, tests done on top of the machine learning model could include more frequent follow up visits, tailored medication adjustments, or referrals to specialized heart failure clinics for more in-depth analysis. Ultimately, this is what can also contribute to helping alter patient trajectories. In a real world implementation, our machine learning model would need to be integrated into an electronic health record system with all the datasets available. New studies in hospital settings would be necessary and beneficial to evaluate not only predictive accuracy, but also downstream effects on patient outcomes.

In summary, our team's project indeed shows that a Bayesian network trained on two available biomarkers such as serum creatinine and LVEF can outperform commonly used classifiers in death among heart failure patients. This is even when we are confronted with severely imbalanced data. The Bayesian network captures subtle variations between renal function and cardiac performance that other discriminative methods often fail to find. Nevertheless, the Bayesian network model performance is inherently constrained by the given data imbalance and the limited feature set. Our future work could possibly address these limitations by incorporating class balancing strategies, expanding the feature space to include more clinical variables, and constantly validating model performance in prospective clinical settings. At the end of the day, our team's goal is to translate predictive insights into actionable interventions of heart failure that together work to improve patient care and reduce death associated with heart failure.

VII. MEMBER CONTRIBUTION & PROJECT ROADMAP

The link to our Github can be found here:
https://github.com/avgale/ECS_171_Group_Project.

Our project roadmap was the following:

The individual member contributions took the following form: Anna wrote the exploratory data analysis, gen-

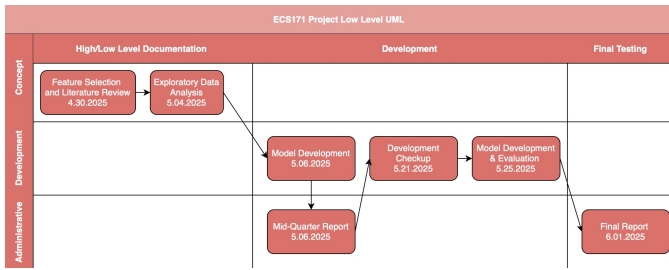


Fig. 8: The project roadmap as defined at the time of the proposal and followed throughout the course of the project.

erated Fig 1 and 3, created the linear regression model, and outlined the video plan. Amabel filtered the dataset, contributed to feature selection statistics, developed the ANN model, and created and styled the demo app. She generated Fig 4 and 5. Sean developed the grid search on the ANN model, proposed methodology, and ANN analysis. Daniel developed the Naive Bayesian Network model and contributed to the proposed methodology and limitations sections of the report. He created the project roadmap graphics, Naive Bayes ROC figures and helped generate some presentation slides. Grace found the dataset and did literature review to supplement the introduction and background section. She also did the model testing and evaluating for the logistic regression model. Lastly, she made the following figures: Fig 2, 6, and 7.

REFERENCES

- [1] 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, Feb. 2020, doi: 10.1186/s12911-020-1023-5.
- [2] J. Wang et al., "Predicting Mortality Risk in Heart Failure Patients: Insights from COX Regression and Machine Learning," *IEEE Conference Publication — IEEE Xplore*, pp. 601–606, Nov. 2024, doi: 10.1109/MedAI62885.2024.00085.
- [3] Z. Zhang et al., "Electronic healthcare records and external outcome data for hospitalized patients with heart failure," *Scientific Data*, vol. 8, no. 1, Feb. 2021, doi: 10.1038/s41597-021-00835-9.
- [4] C. Chen-Scarabelli, L. Saravolatz, B. Hirsh, P. Agrawal, and T. M. Scarabelli, "Dilemmas in end-stage heart failure," *Journal of Geriatric Cardiology*, vol. 12, no. 1, pp. 57–65, 2015, doi: 10.11909/j.issn.1671-5411.2015.01.007.
- [5] B. Bozkurt et al., "HF STATS 2024: Heart failure epidemiology and outcomes statistics—An updated 2024 report from the Heart Failure Society of America," *Journal of Cardiac Failure*, vol. 31, no. 1, pp. 66–116, 2024.
- [b1] . Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Information Sciences*, vol. Volume 513,