A Replication Study: Heart Failure Risk Factors and Survival

Lilian Zaplin

Student # 501209464

CIND 820: Big Data Analytics Project

Supervisor: Professor Tamer Abdou

December 4, 2023

A Replication Study: Heart Failure Risk Factors and Survival

Table of Contents

Abstract

Data mining is becoming a vanguard in the field of biomedical informatics and research. Electronic health records (EHR) are now available that contain patient lifestyle, symptoms, clinical laboratory tests and survival data. From EHR datasets, data mining can be used to discover previously unknown correlations in patient data. Machine learning models can be developed to assess risk factors for chronic disease such as diabetes, and to predict patient survival for critical illnesses such as heart failure and cancer. These models can be used to support medical doctors in patient management.

In my project, I will replicate the analysis of Chicco and Jurman (2020) in developing machine learning models to predict the risk factors and survival of patients with heart failure. (The original analysis was done in R but I will use Python.) Further to duplicating the analysis results, I will address two issues: First, that results could be impacted by the significant imbalance in the dataset towards patients having survived (67.89% majority negative class); this issue was identified but not addressed by the authors. And, second, that other researchers using the same dataset (Ishaq et al., 2021) chose to use more than the top two ranked features in developing their model from the same dataset.

The public open dataset used will be the Heart Failure Clinical Records dataset available in the UC Irvine Machine Learning Repository. The dataset contains the medical records of 299 heart failure patients. The patient demographic is 105 women and 194 men ranging in age from 40 to 95 years old. The dataset contains 13 features relating to patient clinical and lifestyle information. Features are binary, integer and real; for example, anemia, ejection fraction and serum creatinine. The binary classification target is the event of death.

In my analysis to develop the machine learning model to predict the risk factors and survival of patients with heart failure, I have used two tree-based classifiers (Random Forest and Extra Trees); one regression-based algorithm (Logistic Regression); and, one ensemble method (AdaBoost). I have used GridSearchCV to tune the hyperparameters of each algorithm.

I addressed the imbalance problem in the dataset by using SMOTE + ENN, a hybrid technique that results in greater class distinction through generating new instances by interpolation between nearby positive instances, as well as data cleaning where misclassified instances of both classes are removed**.**

The highest ranked risk factors were determined by using select from model with random forest to be: ejection fraction, serum creatinine, age, creatine phosphokinase, serum sodium and platelets. Ishaq et al. (2021) found the same six most relevant features in their study.

The final random forest model fitted to the five key risk factors achieved an MCC of 0.885, and F1 score of 0.917, a balanced accuracy of 0.927, and a recall of 0.865 which exceeded the best performance results of the original study.

Data Description

The public dataset used for this study can be downloaded from the UCI Machine Learning Repository at http://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records.

The dataset (heart_failure_clinical_records_dataset) contains the medical records of 299 patients with left ventricular systolic heart failure who had had previous heart failure and were in classes III or IV of the New York Heart Association classification of the stages of heart failure. The data was collected at the Faisalabad Institute of Cardiology and the Allied Hospital in Faisalabad in Punjab, Pakistan, from April- December 2015 and was used in the study being replicated (Chicco, D., & Jurman, G., 2020).

The patient demographic is 105 women and 194 men ranging in age from 40 to 95 years old. The dataset contains 13 features relating to patient clinical and lifestyle information. Features are binary, integer and real. The binary classification target is the event of death. The data dictionary is given in Table 1.

Table 1 *Heart Failure Dataset Description*

Feature	Description	Measurement	Range
Age	Age of the patient	Years	40- 95
Sex	0=Woman, 1=Man	Binary	0, 1
Anaemia	Decrease in red blood cells or hemoglobin	Boolean	0, 1
Diabetes	If the patient has diabetes	Boolean	0, 1
High Blood Pressure	If the patient has hypertension	Boolean	0, 1
Smoking	If the patient smokes	Boolean	0, 1
Ejection Fraction	Percentage of blood leaving the heart at each contraction	%	14- 80
Creatinine Phosphokinase	Level of the Creatinine phosphokinase enzyme in the blood	mcg/L	23- 7861
Serum Sodium	Level of sodium in the blood	mEq/L	114- 148
Platelets	Platelets in the blood	kiloplatelets/mL	25.01- 850.00
Serum Creatinine	Level of creatinine in the blood.	mg/dL	0.50- 9.40
Time	Follow-up period	Days	4- 285
Death Event (target)	If the patient died during the follow-up period 0=Survived, 1=Death	Boolean	0, 1

Heart failure is one of the pathologies of cardiovascular disease (disorders of the heart and blood vessels) along with heart attacks (coronary heart disease) and strokes (cerebrovascular disease). Heart failure is of two types based on the proportion of blood pumped out of the heart during a contraction known as the ejection fraction. Heart failure due to left ventricular systolic dysfunction (reduced ejection fraction), occurs when the ejection fraction value is less than 40%.

Of the other features, the enzyme creatinine phosphokinase flows into the blood when a muscle becomes damaged; so, high levels in the patient's blood may be indicative of heart failure or injury. Serum creatinine is also produced when a muscle breaks down and high levels in the blood may indicate renal dysfunction. Serum sodium in the blood is routinely tested for because sodium is a mineral required for the correct functioning of muscles and nerves and an abnormally low level of sodium might be caused by heart failure. A patient was considered anaemic with a haematocrit level lower than 36%. No definition was given of high blood pressure.

Initial Analysis

Variables were mapped to the correct data types. There were no missing values or duplicates. A 5 Number summary (see Table 2) and Frequency histograms and Box Plots were generated for each numeric variable (see Figures 1 and 2).

Table 2. *5 Number Summary of Numerical Features*

	age	creatinine_phosphokinase	ejection_fraction	platelets	serum_creatinine	serum_sodium	time
count	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.000000
mean	60.829431	581.839465	38.083612	263.358029	1.39388	136.625418	130.260870
std	11.894997	970.287881	11.834841	97.804237	1.03451	4.412477	77.614208
min	40.000000	23.000000	14.000000	25.100000	0.50000	113.000000	4.000000
25%	51.000000	116.500000	30.000000	212.500000	0.90000	134.000000	73.000000
50%	60.000000	250.000000	38.000000	262.000000	1.10000	137.000000	115.000000
75%	70.000000	582.000000	45.000000	303.500000	1.40000	140.000000	203.000000
max	95.000000	7861.000000	80.000000	850.000000	9.40000	148.000000	285.000000

Figure 1. *Histograms of Numerical Features*

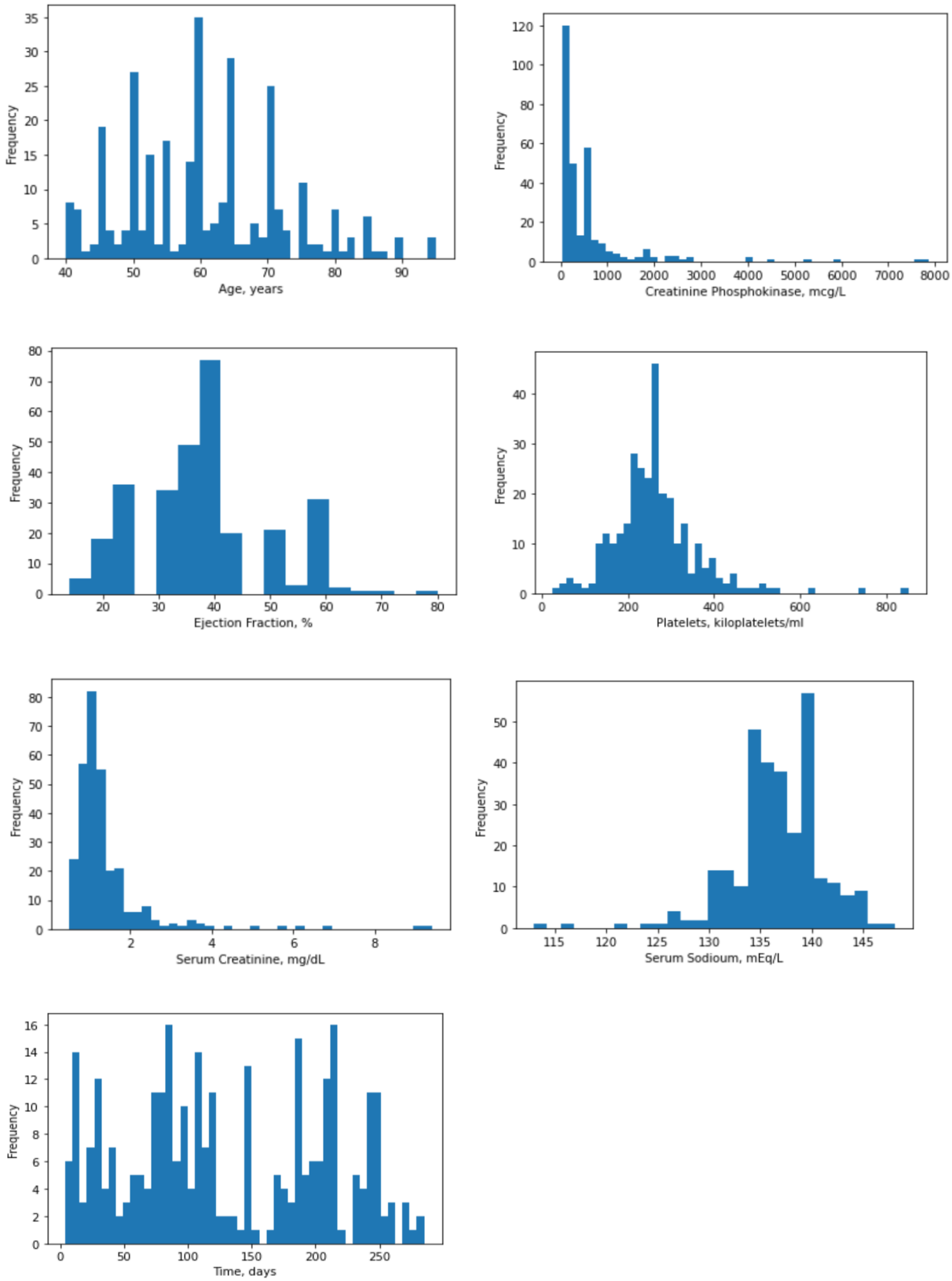
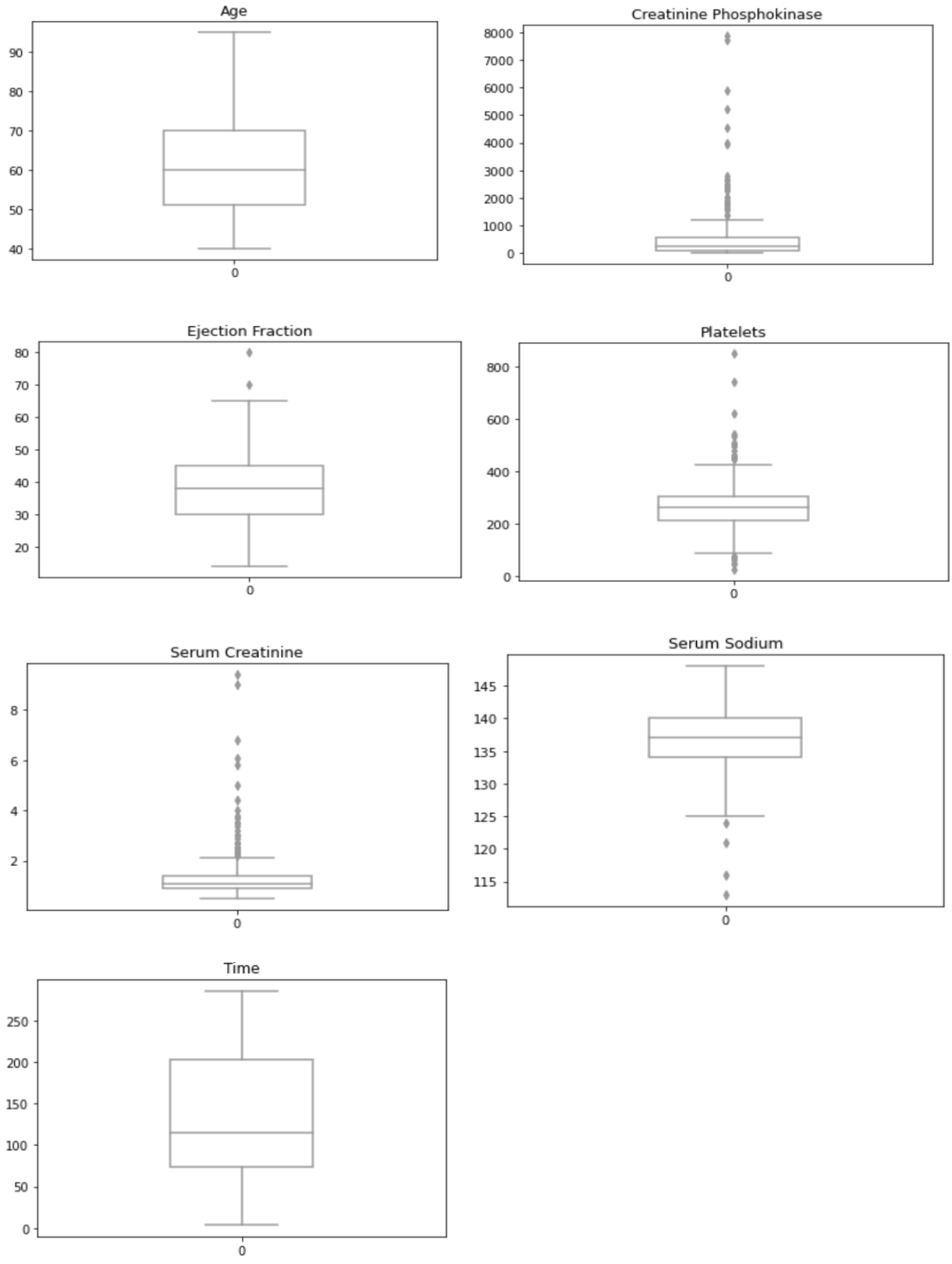


Figure 2. *Box Plots of Numerical Features*



From Figure 1 it can be seen that Age, Ejection Fraction, Platelets, and Serum Sodium appear to have relatively symmetric distributions. Creatinine Phosphokinase and Serum Creatinine are skewed right and Serum Sodium is skewed left.

As seen in Figure 2, outliers occurred in all the numeric variables (except age and time) to varying degrees. With such a small dataset, all outlier values should be retained because many are associated with a patient death.

Normalization is required for distance-based classifiers such as K-Nearest Neighbors. Features with a large range will have a large influence in computing the distance. For that reason, normalization based on max-min, (x – min(x)) / (max(x) – min(x)), doesn't work well with outliers. Instead, standardization or z-score normalization,

(x – mean(x)) / standard deviation (x)

will be used to rescale the features. Standardization will be applied to the training set only.

A heat map plot of the Pearson correlation matrix for the heart failure dataset is shown in Figure 3 and the actual values are shown in Table 3. No significant correlation between pairs of independent variables was observed and so no variables will be removed on this basis.

Figure 3. *Heat Map of Pearson Correlation Matrix for Heart Failure Features*

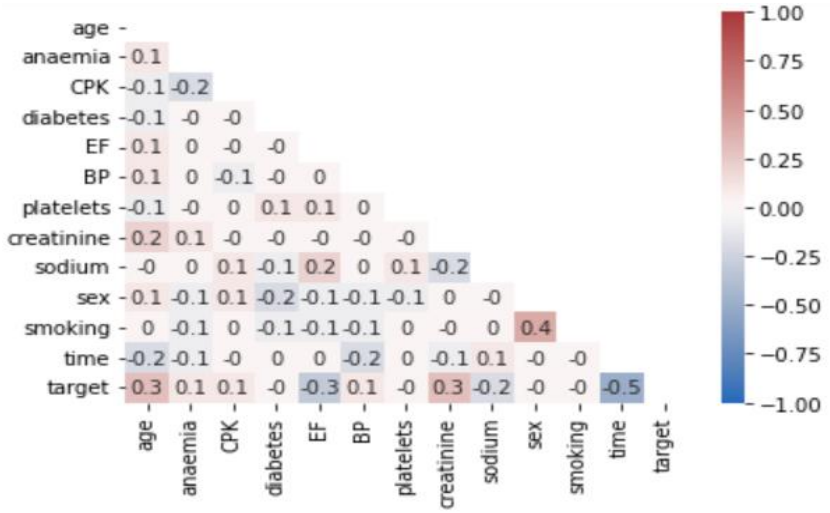
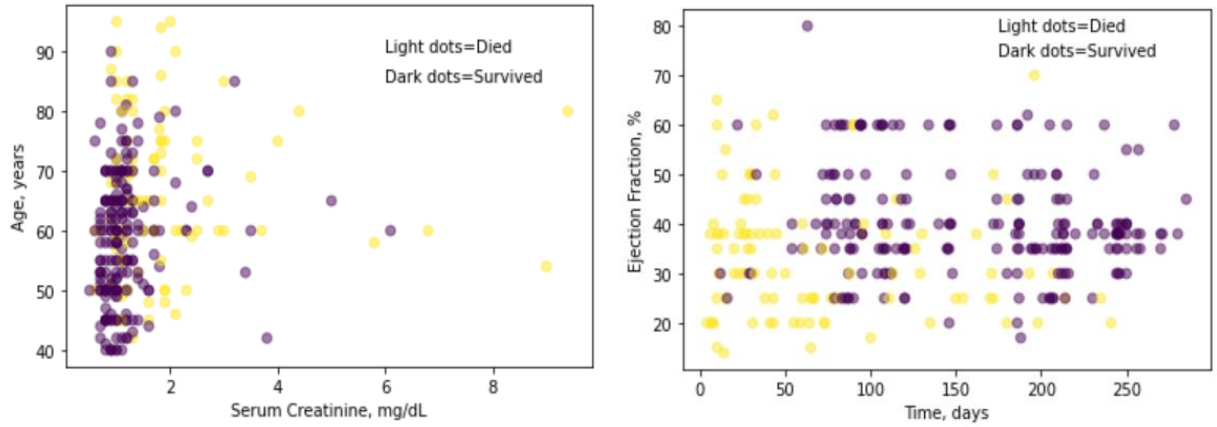


Table 3. *Pearson Correlation Values for Heart Failure Features*

	age	anaemia	CPK	diabetes	EF	BP	platelets	creatinine	sodium	sex	smoking	time	target
age	1.000	0.088	-0.081	-0.101	0.060	0.093	-0.052	0.159	-0.046	0.066	0.019	-0.224	0.254
anaemia	0.088	1.000	-0.191	-0.013	0.032	0.038	-0.044	0.052	0.042	-0.095	-0.107	-0.141	0.066
CPK	-0.081	-0.191	1.000	-0.010	-0.044	-0.071	0.024	-0.016	0.060	0.080	0.002	-0.009	0.063
diabetes	-0.101	-0.013	-0.010	1.000	-0.005	-0.013	0.092	-0.047	-0.090	-0.158	-0.147	0.034	-0.002
EF	0.060	0.032	-0.044	-0.005	1.000	0.024	0.072	-0.011	0.176	-0.148	-0.067	0.042	-0.269
BP	0.093	0.038	-0.071	-0.013	0.024	1.000	0.050	-0.005	0.037	-0.105	-0.056	-0.196	0.079
platelets	-0.052	-0.044	0.024	0.092	0.072	0.050	1.000	-0.041	0.062	-0.125	0.028	0.011	-0.049
creatinine	0.159	0.052	-0.016	-0.047	-0.011	-0.005	-0.041	1.000	-0.189	0.007	-0.027	-0.149	0.294
sodium	-0.046	0.042	0.060	-0.090	0.176	0.037	0.062	-0.189	1.000	-0.028	0.005	0.088	-0.195
sex	0.066	-0.095	0.080	-0.158	-0.148	-0.105	-0.125	0.007	-0.028	1.000	0.446	-0.016	-0.004
smoking	0.019	-0.107	0.002	-0.147	-0.067	-0.056	0.028	-0.027	0.005	0.446	1.000	-0.023	-0.013
time	-0.224	-0.141	-0.009	0.034	0.042	-0.196	0.011	-0.149	0.088	-0.016	-0.023	1.000	-0.527

From Table 3 it can be observed that the strongest positive relationship to the target variable is with Serum Creatinine followed by Age while the strongest negative relationship to the target variable is with Time followed by Ejection Fraction. Figure 4 shows scatter plots of these features and the target variable.

Figure 4. *Scatter Plots of Features with the Strongest Correlation to the Target (Death) Variable*



There exists a significant imbalance in the dataset towards patients having survived (67.89% majority negative class).

	count	percentage
False	203	67.9
True	96	32.1

Literature Review

A summary review of current studies in multimodal machine learning in precision health gave one aim: the support of clinical diagnoses made by medical doctors. Precision health is an approach intended to customize a patient's healthcare (decisions, treatments, practices, etc.) based on their individual phenotype (the interaction of their genotype with their environment). A clinical diagnosis can be based on symptoms, single or multiple lab values, and imaging. However, a clinical diagnosis made by medical doctors may fail to consider the relative weighting and relationships of the disparate data sources, resulting in a one-size-fits-all treatment per disorder. Multimodal machine learning is the field of machine learning that brings together and harmonizes disparate data sources to perform algorithmic modeling. The data sources can include genomics, clinical notes, time series, demographics, lab values, and imaging. The algorithmic models can be used, for example, to provide efficient and evidence-based drugs and treatments to diverse sub-populations or provide insight into clinical trials and drug discovery (Kline et al., 2022).

According to the World Health Organization (WHO), cardiovascular disease (CVD)—diseases of the heart or blood vessels which can lead to heart attacks, stroke, and heart failure—is responsible for 31% of deaths globally. Heart failure occurs when not enough blood can be pumped to the body because the heart muscles have enlarged and the heart ventricles have stiffened, causing fatigue and difficulty breathing. Pakistan is included in the WHO list of countries where CVD is increasing significantly with a consequent increase in heart failure in the population (Ahmad et al., 2017).

The rest of the literature review highlights that the modeling algorithms applied to binary classification problems of heart health data are becoming increasingly sophisticated beginning

with biostatistical analysis, proceeding to conventional machine learning, ensemble modelling and through to deep learning. The protocol of all the studies was the same: First, to use the complete set of features in the classification algorithm, and measure the model performance using all the features. Second, to apply feature selection techniques and measure the model performance using only the selected features. For each author, I will summarize their methodology and results (see Table 4).

Ultimately, all the researchers aimed to develop an efficient, accurate and reproducible methodology for doctors to use in a clinical setting to assist diagnoses. Another goal was to achieve optimized models in terms of stability, scalability, memory usage and processing time.

The study by Ahmad et al. (2017) was the original analysis of the survival of heart failure patients using the data that was collected at a cardiology institute and hospital in Faisalabad (the third most populus city in Pakistan) from April to December 2015. The dataset was released to *UCI* as the Heart Failure Clinical Records dataset. The main objective of the original study was to investigate death rates due to heart failure and the associated risk factors relating to a Pakistani demographic because the available studies were largely in relation to the West.

Statistical regression modeling was applied using the Cox proportional hazards model to investigate the association between the predictor variables and patient survival time. It was concluded that increasing age, a low ejection fraction, high serum creatinine (associated with renal dysfunction), and a high level of anemia were the key factors contributing to death among heart failure patients, while increased levels of serum sodium can reduce the odds of death. Smoking, diabetes, and gender were found to be non-significant.

In contrast, the main objective of the study by Zahid et al. (2019) using the *UCI* heart failure clinical records dataset, was to build and assess the performance of separate survival

prediction models for men and women using gender-specific risk factors. The authors noted, for example, differences in the normal levels of some clinical factors between men and women, and that the left ventricular systolic dysfunction, the subject of the study, is more common in males than females.

The authors found a significant difference in the heart failure survival prediction models of male and female patients and in the risk factors. Using the Cox proportional hazards model and group lasso for feature selection, it was determined that the informative variables for men are blood pressure, age, ejection fraction, serum sodium, serum creatinine, platelets, and creatinine phosphokinase; and that the informative variables for women are smoking, diabetes, blood pressure, anemia, age, serum creatinine, and creatinine phosphokinase.

I am replicating the study by Chicco and Jurman (2020) that used the heart failure clinical records dataset. Preceding studies by Ahmad et al. (2017), and Zahid et al. (2019), applied biostatistical analyses, whereas the researchers here intended to further develop the solution by applying machine learning approaches.

An extra step that was unique to these researchers regarding the objectives was to, first predict patient survival and rank the features without including patient follow-up time in the event of death, and second, repeat the analysis including the time feature. The analysis (without time and with time) was then repeated using only the top two ranked features.

Ten different machine learning methods were applied to the complete dataset: decision tree, random forest, logistic regression, naïve Bayes, one rule, artificial neural network, support vector machines linear, support vector machines with Gaussian radial kernel, k-nearest neighbors and, gradient boosting.

Random forest was used to determine the top two ranked features: serum creatinine and ejection fraction. Using only these two top features and excluding time, the survival prediction classification was repeated using random forest, gradient boosting and support vector machines with Gaussian radial kernel. There was no definitive improvement in performance when only the two features were used in comparison to the corresponding results when using the complete dataset excluding time.

Stratified logistic regression was then applied to the complete dataset, including a derived follow-up month from the time feature, to predict patient survival and feature ranking. Serum creatinine and ejection fraction were again identified as the top two clinical features. Stratified logistic regression using only serum creatinine, ejection fraction and the follow-up month feature, resulted in a model with the highest accuracy value of 0.838 that outperformed all other methods attempted.

The study by Ishaq et al. (2021) used the heart failure clinical records dataset and cited the preceding works by Ahmad et al. (2017), Zahid et al. (2019), and Chicco and Jurman (2020). The study had the same objective and followed the same analysis approach as the study being replicated, but with the important difference that the Synthetic Minority Oversampling Technique (SMOTE) was used to balance the dataset, and that the time variable was always included. The machine learning models were trained on both the imbalanced and balanced datasets, and the prediction results were evaluated.

Nine classification modeling techniques were used. Tree-based ensemble techniques were decision tree, random forest, and Extra Trees classifier. Tree-based boosting techniques were adaptive boosting and gradient boosting. Regression-based techniques were logistic

regression and stochastic gradient. And statistical-based techniques were Gaussian naïve Bayes and support vector machine.

The use of SMOTE had a varied impact on the performance of the machine learning classifiers. The performance of the tree-based ensemble and boosting classifiers improved, except for gradient boosting, which showed no change, while the performance of regression-based and statistical-based classifiers decreased.

The highest-ranked features were identified using random forest with SMOTE to be time, serum creatinine, ejection fraction, age, platelets, creatinine phosphokinase, and serum sodium. The prediction results using the Extra Trees classifier with SMOTE, for the complete set of features and for only the significant features, both achieved the highest accuracy of 0.9262.

A different *UCI* dataset—the heart disease (Cleveland) dataset—was used by Javid et al. (2020) in a study of heart disease prediction modeling which compared the performance of various machine learning and deep learning classifiers. The authors also investigated improving the accuracy of heart disease prediction modeling by creating a voting-based model that combined all five of the machine learning and deep learning classifiers from the study.

The machine learning methods were random forest, support vector machine and k-nearest neighbor; and the deep learning methods were long-short-term memory and gated-recurrent unit neural network. The ensemble voting-based model combined all five methods and used all thirteen features of the heart disease (Cleveland) dataset. The ensemble voting-based model using the complete set of features, proved the most accurate for heart disease prediction, with an accuracy value of 0.857.

Ghosh et al. (2021) sought to improve the performance of heart disease prediction modeling achieved by Javid et al. (2020) by using feature selection on the complete *UCI* heart disease dataset (Cleveland, Hungary, Switzerland, VA Long Beach).

Two feature selection techniques were used: relief and least absolute shrinkage and selection operator (LASSO). The authors used the ensemble learning techniques of bagging and boosting to create ensemble modeling methods: bagging decision tree, bagging random forest, bagging k-nearest neighbors, boosting adaptive boosting, and boosting gradient boosting. Bagging random forest using 10 features selected by the relief method from the combined heart disease dataset achieved the highest accuracy of 0.991.

Table 4 summarizes the different datasets, classification and feature selection techniques, and accuracy results that have been referred to in the literature review.

Table 4. *Comparison Table of Accuracy Between Proposed Models*

Author	*UCI* Dataset	Model	Accuracy All Features	Feature Selection	Ranked Selected Features (Number of)	Accuracy Selected Features
Ahmad et al. (2017)	Heart failure clinical records	Cox regression			Age, ejection fraction, serum creatinine, serum sodium, anaemia, (5)	
Zahid et al. (2019)	Heart failure clinical records	Cox regression		Group lasso	Men: blood pressure, age, ejection fraction, serum sodium, serum creatinine, platelets, creatinine phosphokinase (7) Women: smoking, diabetes, blood pressure, anemia, age, serum creatinine, creatinine phosphokinase (7)	
Chicco and Jurman (2020)	Heart failure clinical records	Stratified logistic regression	0.833	Random forest	Serum creatinine, ejection fraction (2)	0.838
Ishaq et al. (2021)	Heart failure clinical records	Extra Trees with SMOTE	0.926	Random forest	Time, serum creatinine, ejection fraction, age, platelets, creatinine phosphokinase, serum sodium (7)	0.926
Javid et al. (2020)	Heart disease (Cleveland)	Ensemble voting-based	0.857	Na	All (13) for heart disease dataset (Cleveland)	Na
Ghosh et al. (2021)	Heart disease (4 combined)	Bagging random forest	0.927	Relief	(10) from heart disease dataset (4 combined)	0.991

The replication study, using the two features selected by random forest (serum creatinine and ejection fraction) in a stratified logistic regression model, was outperformed by all the other classification models.

Generalizing the results of any study using the heart failure clinical records dataset should be carefully considered in light of three limitations: first, the small size of the dataset; second, the bias in the dataset that specifically represents a Pakistani demographic; and third, the male and female subsets in the data.

A further limitation to the replication study noted by the authors (Chicco & Jurman, 2020) was that all methods in all scenarios predicted better scores on the true negative rate than on the true positive rate because the dataset was significantly imbalanced with a 67.89% negative class majority. As mentioned previously, Ishaq et al. (2021) proposed handling this imbalance by using SMOTE; however, they observed that the performance of regression-based logistic regression models—the type put forward by the replication study as the best performing model when the time variable was included—actually decreased with SMOTE.

Predicting patient survival from heart failure is a challenging clinical data analytics problem with no fully reproducible methodology developed yet.

Methodology

Replication Study

In the original study (Chicco & Jurman, 2020) to be replicated, the performances of various classifiers were evaluated for all features in the dataset with the exclusion of the time variable, and the features were ranked. It was concluded that Random Forest outperformed all other classifiers tested. Second, the authors selected only the top two ranked features identified and created a model using only these two features. The top two ranked features (identified by aggregating the rankings determined by various classifiers into a Borda count score) were determined to be serum creatinine and ejection fraction.

Best Results Techniques from Subsequent Study

In the subsequent study by Ishaq et al. (2021), the author's objective was to increase the accuracy of prediction of patient survival by balancing the dataset using synthetic minority oversampling technique (SMOTE). Extra Trees Classifier (ETC) was found to give the highest performance metrics. The authors again used random forest feature selection as had Chicco and Jurman (2020); but, chose instead to use the top six features scoring greater than 0.05 mean decrease in impurity: serum creatinine, ejection fraction, age, platelets, creatinine phosphokinase, and serum sodium.

Project Study Design

Four test conditions (TC) will be evaluated to address my research questions ("Abstract" section) based on the replication (Chicco & Jurman, 2020) and subsequent study (Ishaq et al, 2021).

TC1: The replication study identified random forest classifier (RF) and logistic regression (LR) as the best performing machine learning algorithms for the imbalanced dataset.

The subsequent study identified extra trees classifier (ETC) as the best performing in conjunction with using SMOTE to balance the dataset.

In the first test condition, I will fit RF, ETC, and LR with default parameter settings to the complete, imbalanced dataset.

TC2: The dataset will be balanced using synthetic minority oversampling technique plus edited nearest neighbors (SMOTE+ENN). The objective of using SMOTE+ENN is not only to balance the dataset but also to further improve the prediction accuracy compared to the use of SMOTE alone as in the subsequent study by Ishaq et al. (2021).

In the second test condition, I will fit RF, ETC, and LR with default parameter settings to the complete, now balanced dataset.

TC3: To further improve the accuracy of the heart failure prediction model of the project study, in addition to balancing the dataset, hyperparameter tuning will be performed on the classifiers which will then be fitted on the balanced dataset.

In the third test condition, I will fit RF, ETC, LR and AdaBoost (with LR as the base estimator) now with tuned hyperparameters, to the complete, balanced dataset.

TC4: Both the original and subsequent studies selected the most significant features based on the feature importance from random forest feature selection. The project study will also use "SelectFromModel" with RF as the base estimator and classifier, to achieve automatic selection of the most significant features from the balanced dataset.

The prediction results of all models will be measured through the confusion matrix metrics: Matthews Correlation Coefficient (MCC), F1 score, accuracy, balanced accuracy, true positive rate (recall), and the receiver operating characteristic area under the curve (ROC), where they are available from the studies. The models created in my project study also will be

evaluated and compared for their performance in terms of efficiency (runtime and memory usage), and stability.

The test conditions investigated in the methodology are laid out in the workflow steps in Table 5. The python script for my analysis was developed from this workflow.

Table 5. *Workflow Steps*

Workflow	Steps
Data loading	- heart_failure_clinical_records_dataset - Drop time feature from data - Map variables to correct datatype and power
Data feature engineering	- Transform numeric data using Robust Scaler - Passthrough all binary data and features without outliers - Apply SMOTEENN
Data splitting	- Test size of 20% as per original study - No seed set - Stratify parameter set to y
Classifier hyperparameter tuning	- Defined hyperparameter grid to search - Grid Search Cross Validation using AUC-ROC scoring metric a. Tree-based (random forest and extra trees) b. Logistic regression c. AdaBoost
Model training & fitting	- Define the classifier: - random forest, extra trees, logistic regression, AdaBoost a. Default b. Tuned - Define model pipeline for preprocessing, classifier and feature selection 1. Imbalanced, complete dataset; classifier with default settings 2. Balanced, complete dataset; classifier with default settings 3. Balanced, complete dataset; tuned classifier 4. Balanced dataset; select from model features using random forest as the base estimator; tuned random forest classifier
Model evaluation	- MCC, F1 Score, Accuracy, Balanced accuracy, Recall, Precision, AUC-ROC
Feature importance ranking	- Mean decrease in impurity method (tree classifiers) - Permutation importance method
Hypothesis statistical significance testing	- Shapiro-Wilk test to assess normality of data - One sample t test for normally distributed data - Wilcoxon signed-rank test for non-parametric data

Data Loading

The heart_failure_clinical_records_dataset is a public dataset used for this study that can be downloaded from the UCI Machine Learning Repository at http://archive.ics.uci.edu/dataset/519/heart+failure+clinical+records. The mapping of variables to datatypes, and the exclusion of the time variable initially, is as per the original study.

It is important to note that I decided to exclude the time feature in my final analysis because the objective of the original study by Chicco and Jurman (2020) was to develop a model that doctors could use in a hospital setting to predict patient survival from the electronic health records available at that time. The time feature was a measurement of days to a later follow-up visit and did not directly relate to the data collected during the patient's initial stay in hospital. Furthermore, the time to follow-up would not strongly suggest any causality with the death event because it was set to a maximum value if follow-up had not yet occurred.

Data Feature Engineering

All the numeric variables, except age, had many outliers. Therefore, RobustScaler was used to standardize the numeric variables rather than StandardScaler. RobustScaler is less sensitive to outliers than StandardScaler because it uses the median and interquartile range (IQR) to scale the data and not the mean and standard deviation in Z-score normalization.

Age and the remaining Boolean and binary variables were passed through without transformation. The Random Forest algorithm can work with Boolean and binary data and can use them as features for making decisions at splits of decision tree nodes like other feature data types.

Data Splitting

No seed was set in the train-test-split so that different selections of data instances for the dataset splits would always occur with each execution leading to slightly different results. The stratify parameter in the train-test-split was set to yes. This is an important setting in modeling imbalanced datasets because it ensures that the class distribution of the target variable is maintained in both the training and test sets in the same proportion as in the dataset.

Classifiers Selected for Analysis

In a study by Olson et al. (2017) they stated that there is no silver bullet algorithm and that a suite of algorithms must be tested on a given dataset to see what works best.

The prediction results of the replication study (Chicco & Jurman, 2020) showed that Random Forest outperformed all the other methods they had tested. In a subsequent study utilizing the same dataset (Ishaq et al., 2021) but addressing the imbalance in the dataset using SMOTE, the best performing algorithm was found to be Extra Trees. Also, in another study relating to cardiovascular disease prediction (Ghosh et al., 2020) the ensemble methods of bagging and boosting were found to increase performance.

Based on these results, in my project study I will use four algorithms: two tree-based algorithms—Random Forest Classifier (RF) and Extra Trees Classifier (ETC)—one regression-based algorithm—Logistic Regression (LR)—and one ensemble method—AdaBoost with LR as the base estimator.

Classifier Hyperparameter Tuning

The study by Olson et al. (2017) also found that hyperparameter tuning of classifiers on datasets with no feature engineering other than standardization, would achieve improvements in the performance metrics of from 3% to 50%.

GridSearchCV was chosen for hyperparamter tuning in the project study instead of RandomizedSearchCV. RandomizedSearchCV randomly selects only a subset of hyperparameters from the possibilities specified with a much lower computational cost but GridSearchCV conducts an exhaustive search of all possible hyperparameter combinations.

The goal of classifier hyperparameter tuning is to increase model accuracy along with decreasing both model overfitting and computational complexity. In the case of the tree-based classifiers (Random Forest and Extra Trees) increased accuracy can be achieved by increasing the number of decision trees (n estimators) in the ensemble. Overfitting and computational complexity can be decreased by shallower trees (max depth) with smaller number of leaf nodes (max leaf nodes) in any tree; and, larger numbers of samples required (min samples split) with smaller numbers of features (max features) at a split node. Also, class weight parameters of balanced or balanced subsample are criteria relating to the imbalanced dataset.

In the case of Logarithmic Regression, the "C" parameter inversely controls the regularization strength. Regularization helps prevent the model from becoming too complex and overfitting the training data in the following way. Smaller values of "C" correspond to increased regularization that will cause the model to have coefficients with smaller absolute values (simpler coefficients), potentially reaching zero, thereby excluding features from the model. Penalty is another parameter that affects regularization. Lasso (L1) regularization penalty is

more likely to achieve simpler coefficients than Ridge (L2) regularization penalty and thereby reducing model complexity. Specific to this dataset is the use of class weight parameter to address class imbalance. Also, in relation to this dataset the 'liblinear' solver optimization algorithm which is suitable for small datasets, is used in conjunction with L1 regularization (lasso) penalty to achieve the optimal coefficients for the model features. Table 6 is a table of algorithms and the hyperparameters evaluated using GridSearchCV.

Table 6. *Algorithms and the Hyperparameters Evaluated*

Algorithm	Hyperparameters
Random Forest (RF) & Extra Trees Classifier (ETC)	**n estimators:** Number of decision trees in the ensemble. **max depth:** Length of the longest path from the root node to a leaf node. Controls the depth of the trees in the forest. **max leaf nodes:** Controls the maximum number of leaf nodes of any individual decision tree in the ensemble. **min samples split:** Threshold on the number of samples required in a node for the algorithm to consider making a further split. If the number of samples in a node is less than the specified value, the node won't be split, and it will become a leaf node. **max features:** Number of features to consider when computing the best node split. **class weight:** Addresses dataset imbalance by assigning higher weights to the minority class and lower weights to the majority class. **criterion:** Function used to measure the quality of a split.
Logistic Regression (LR)	**C:** Regularization strength. **penalty:** Lasso or Ridge regularization. **class weight:** Addresses class imbalance. **solver:** Optimization algorithm used to find the optimal set of weights (coefficients) for the input features, allowing the model to make accurate predictions of the probability of belonging to a particular class. **max iter:** Specifies the maximum number of iterations that the optimization algorithm is allowed to run when fitting the model to the training data. Prevents the model from running indefinitely if convergence is not reached.
AdaBoost	**n estimators:** Number of decision trees in the ensemble. **learning rate:** Shrinks the contribution of each successive decision tree in the ensemble.

Model Training, Fitting, and Evaluating

Table 7 defines the binary classification metrics used to evaluate the model performance.

Table 7. *Binary Classification Metrics*

Metric	Definition	Value Range
Matthews Correlation Coefficient (MCC)	Good scoring measure for imbalanced datasets because it doesn't favor either the positive or negative class. It is calculated from all four confusion matrix scores—TP, TN, FP, and FN.	-1 to 1 where: 1 indicates a perfect prediction 0 indicates randomness -1 indicates total disagreement
F1 Score	Good scoring measure for imbalanced datasets that provides a balanced score between precision and recall. It is referred to as the harmonic mean. It gives more weight to lower, and penalizes more extreme values.	0 to 1 where: 1 indicates perfect precision and recall 0 means either precision or recall is 0
Accuracy	Not a good scoring measure for imbalanced datasets because it may overpredict majority class. It is the ratio of the number of correct predictions (where the model's prediction matches the actual class label) to the total number of predictions.	0 to 1 where: 1 means all predictions are correct 0 means no predictions are correct
Balanced Accuracy	Good scoring measure for imbalanced datasets calculated as the mean of sensitivity (true positive rate) and specificity (true negative rate	0 to 1 where: 1 means all predictions are correct 0 indicates randomness
Recall	Recall is the ratio of TP to the sum of TP plus FN. It measures how well the model correctly identifies all relevant instances in a dataset. Recall is also known as the TP rate.	0 to 1 where: 0 means no TP were captured, only FN 1 means that the model is capturing all TP and no FN.
Precision	Precision is the ratio of TP to the sum of TP plus FP. It measures how well the model correctly identifies positive predictions.	0 to 1 where: 0 means no TP were captured, only FP. 1 means that the model is capturing all TP and no FP.
AUC-ROC	(Area Under the Receiver Operating Characteristic) AUC-ROC is is less sensitive to class imbalance. The ROC curve plots the TP rate against the FP rate at various threshold settings. AUC-ROC calculates the area under the ROC curve. A higher AUC-ROC value indicates better identification of positives and negatives.	0 to 1 where: 0.5 indicates randomness 1 indicates a perfect model

Feature Selection

One of the goals of the replication (Chicco & Jurman, 2020) and subsequent study (Ishaq et al., 2021), was to find the significant features to achieve accurate predictions of the death event so that the complete dataset would not be required to be used as input in a clinical setting.

In both studies, random forest feature selection was used to determine the significant features based on feature importance ranking. I chose to use “SelectFromModel” with Random Forest as the base estimator to have the top features selected automatically.

Random Forest Feature Selection

Random forest feature selection is an embedded method that combines filter and wrapper methods (Dubey, 2023). A wrapper method selects features through forward elimination, backward elimination, and bi-directional selection to find the best performing combination. In the embedded method, the model is trained and the best features can be selected based on how much the feature decreases the impurity (Aman, 2022). Dubey (2023) goes on to say that each of the hundreds of decision trees that make a random forest model, is built from a random sample of dataset instances and features ensuring that the trees are de-correlated and less prone to over-fitting.

“SelectFromModel” Feature Selection

The "SelectFromModel" feature selection method can provide automatic feature selection of the most informative features of a model based on the feature importance generated by the classifier specified as the base estimator.

Feature Importance Ranking

Two methods are used for feature selection and ranking: The Mean Decrease in Impurity method and the Permutation Importance Method.

The Mean Decrease in Impurity Method is typically used to assess feature importance in tree-based models such as Random Forest. This method assumes that features contributing to the reduction of Gini impurity in classification models at each split in the tree are likely to be more important with higher values meaning higher feature importance.

The Permutation Importance method can be applied to a variety of machine learning models. It assesses the importance of each feature by evaluating the change in model performance in terms of accuracy with higher values meaning higher feature importance.

Algorithms

In this section certain algorithms are further explained.

SMOTE + ENN

Imbalanced datasets often occur in health data binary classification problems because the event being studied is significantly the rarer of the two. Analysis of imbalanced datasets can lead to bias in the results. A 1:1 balance can be achieved by either oversampling the minority class or under sampling the majority class. Oversampling is preferred, especially in a smaller dataset, because under sampling can remove potentially important information.

SMOTE (Synthetic Minority Oversampling Technique) is a method of oversampling that generates new instances by interpolation between nearby positive instances.

ENN (edited nearest neighbors) is a method of under sampling that considers the three nearest neighbors of an instance. ENN removes a majority instance when it is misclassified based on its three nearest neighbors. If the instance is of the minority class, and it is misclassified based on its three nearest neighbors, then its nearest neighbors in the majority class are removed.

SMOTE + ENN is a hybrid technique that results in greater class distinction through data cleaning where misclassified instances of both classes are removed (Satpathy, 2023: *What is Undersampling?,* 2022).

Extra Trees

Extra Trees (short for extremely randomized trees) is an ensemble machine learning algorithm with similarities to bagging and random forest. Every one of the unpruned decision trees in Extra Trees are grown from the whole training set, whereas bagging and random forest use bootstrap samples of the training set. The Extra Trees algorithm selects features and cut points completely at random, whereas bagging and random forest use a greedy algorithm (a locally optimal choice at each stage) to select an optimal split point. The resulting prediction from the decision trees, in the case of classification, is determined by majority voting (Brownlee, 2021).

Bagging and Boosting

Bagging and boosting are ensemble learning techniques that combine the results of multiple homogeneous models, referred to as base learners, into a final prediction that has increased performance and accuracy. The final prediction is determined by majority voting for classification. In bagging, a training set is randomly divided into samples, with replacement, which is called bootstrapping. A separate base model is built for each of these samples of the training set and the combined final prediction is determined. Each model in bagging is given the same weight. In boosting, a random set of training data is used to train a base learner. If an error occurs after testing, another base learner is trained for the error. The process continues sequentially with each new model given a weight (*Bagging and Boosting—a Method of Ensemble Learning Using Python | smartyR*, n.d.).

Results

In my project study, I replicated the analysis of Chicco and Jurman (2020) to develop a machine learning model to predict the risk factors and survival of patients with heart failure. The authors approached their analysis in two steps. First, the performances of various classifiers were evaluated for all features in the dataset with the exclusion of the time variable, and the features were ranked. Second, the authors selected only the top two ranked features—serum creatinine and ejection fraction—and developed their model using random forest.

Further to duplicating the analysis results, I addressed two issues with the replication study that were considered in a subsequent study (Ishaq et al., 2021) using the same dataset. First, that the results could be impacted by the significant imbalance in the dataset towards patients having survived (67.89% majority negative class); and, second, that selecting more than just the top two risk factors, along with balancing the dataset using SMOTE could achieve significant improvements in model performance.

There are four research questions in my project. First, regarding the original study (Chicco & Jurman, 2020), to replicate the prediction results of patient survival from heart failure. Second, to replicate the risk factor ranking that may lead to heart failure. Third, to repeat the analysis using SMOTE + ENN to oversample the minority class in combination with random under-sampling of the majority class before modelling. And, fourth, to repeat the analysis using classifiers after hyperparameter tuning rather than default settings.

To address my research questions, four test conditions were evaluated ("Methodology" section). First, random forest, extra trees, and logistic regression algorithms with default parameter settings were fitted to the standardized, imbalanced complete test dataset. Second, the dataset was balanced by applying SMOTE+ENN. And, random forest, extra trees, and logistic

regression algorithms still with default parameter settings were fitted to the standardized, and now balanced complete test dataset. Third, hyperparameter tuning was performed on the random forest, extra trees, logistic regression, and AdaBoost machine learners. Then the tuned random forest, extra trees, logistic regression and AdaBoost (with logistic regression as the base estimator) were fitted to the standardized, balanced complete test dataset. And fourth, random forest feature selection and select from model with random forest as the base estimator techniques were fitted to extract the most significant risk factor features in patient survival.

Models created were evaluated and compared for their confusion matrix performance metrics as well as efficiency and stability.

The final project model was developed and fitted using only the most significant risk factor features identified. The final model's predictive performance, efficiency and stability were evaluated.

Discussion

The test conditions have been previously outlined in the "Methodology" section, and again in the preceding "Results" section, and included classifiers with default or tuned hyperparameters applied to imbalanced or balanced datasets. Feature selection was then applied to the tuned random forest model fitted to the balanced dataset to develop the final model. Model efficiency was measured by the runtime in seconds, and the memory usage in MB, to generate the model (using "tracemalloc"). Table 8, presents efficiency and performances scores for the various conditions tested.

Table 8. *Model Efficiency and Performance Metrics*

Classifier	Tuned	Dataset	Feature Selection	Efficiency		Performance Score, mean / sd						
				Runtime, secs	Memory, MB Current / Peak	MCC	F1 Score	Accuracy	Balanced Accuracy	Recall	Precision	AUC-ROC
Replication Study (Chicco & Jurman, 2020)												
RF	No	Imbalanced	none			0.384	0.547	0.74		0.491		0.8
			RF			0.418	0.754	0.585		0.541		0.698
Project Study												
RF	No	Imbalanced	none	117.8	0.863 0.918	0.501 0.032	0.656 0.023	0.786 0.014	0.748 0.016	0.644 0.031	0.669 0.027	0.792 0.013
		Balanced		124.4	0.390 0.531	0.519 0.045	0.673 0.032	0.790 0.020	0.761 0.023	0.680 0.043	0.667 0.035	0.788 0.012
	Yes	**Balanced**	**none**	661.5	0.725 0.736	**0.548** **0.033**	**0.693** **0.025**	**0.802** **0.013**	**0.776** **0.019**	**0.705** **0.037**	**0.682** **0.018**	**0.789** **0.006**
			SFM(RF)	1,410.6	1.089 1.323	0.519 0.037	0.680 0.025	0.780 0.016	0.769 0.020	0.739 0.037	0.631 0.021	0.777 0.007
ETC	No	Imbalanced	none	80.7	0.613 0.762	0.324 0.068	0.511 0.058	0.722 0.025	0.652 0.035	0.463 0.069	0.574 0.048	0.750 0.016
		Balanced		86.3	0.424 0.547	0.408 0.071	0.553 [1] 0.059	0.759 0.027	0.682 0.035	0.472 0.066	0.671 0.056	0.783 [2] 0.013
	Yes	Balanced	none	268	0.748 0.877	0.402 0.045	0.551 [1] 0.036	0.757 0.018	0.681 0.023	0.474 0.044	0.663 0.039	0.782 [2] 0.006
LR	No	Imbalanced	none	20	0.705 0.785	0.511 0	0.6 0	0.8 0	0.712 0	0.474 0	0.818 0	0.786 0
		Balanced		17.2	0.585 0.712	0.4 0	0.571 0	0.75 0	0.69 0	0.526 0	0.625 0	0.792 0
	Yes	Balanced	none	11.9	0.380 0.586	0.414 0	0.595 0	0.750 0	0.704 0	0.579 0	0.611 0	0.761 0
AdaBoost (estimator = Logistic Regression(class_weight= 'balanced'))	Yes	Balanced	none	1,089.1	1.386 2.618	0.384 0	0.579 0	0.733 0	0.692 0	0.579 0	0.579 0	0.782 0

Notes.

1 & 2. Data is not significantly different from mean value.

Bolded values represent the best performing model and conditions used along with feature selection to develop the final model.

Performance Metrics

The available metrics for the replication study by Chicco and Jurman (2020) are included in Table 8. With respect to the replication study, the first record shows random forest results evaluated for all features in the dataset and in the second record, for only the top two ranked features identified by random forest feature selection—serum creatinine and ejection fraction—fitted to the random forest model.

For the first test condition of the project study, random forest (RF), extra trees (ETC), and logistic regression (LR), with default settings, were fitted to the standardized, imbalanced, complete test dataset. The project results compared to the replication study, were on average:

RF 17.2% higher, ETC 7.4% lower, and LR 9.2% higher.

For the second test condition, the dataset was balanced by applying SMOTE + ENN. And, random forest, extra trees, and logistic regression algorithms still with default parameter settings were fitted to the standardized, and now balanced complete test dataset. For this test condition, the project results compared to the replication study, were on average:

RF 20.6% higher, ETC no change, and LR 3% higher.

It can be observed from these results that SMOTE+ENN increases the performance of tree-based classifiers (RF and ETC) but decreases the performance of regression-based LR. This observation is in agreement with the findings of Ishaq et al. (2021).

For the third test condition, the hyperparameter tuned RF, ETC, LR and AdaBoost (with LR as the base estimator) were fitted to the standardized, balanced complete test dataset. Project results compared to the original study averaged:

RF 24.2% higher, ETC no change, LR 6.2% increase, AdaBoost(LR) 4.2% increase

And fourth, random forest feature selection and "SelectFromModel" with RF as the base estimator, techniques were applied using a tuned RF classifier on the balanced dataset. Six features were selected: ejection fraction, serum creatinine, age, creatinine phosphokinase, and serum sodium. The project results using the six features selected were 19% higher on average than the replication study using only ejection fraction and serum creatinine.

It can be seen in Table 8, that the project model with the best performance metrics was the tuned RF classifier applied to a standardized, balanced dataset. The expected superior performance metrics from ETC and AdaBoost were not observed. Furthermore, using AdaBoost with LR as the base estimator degraded the performance metrics of LR alone.

Stability

The measure of model stability was the standard deviation of the performance metrics derived from the folds of the stratified K-fold cross validation which were repeated 10 times with 10 folds each repeat. ETC was the least stable algorithm with an average standard deviation of 0.038. Logistic Regression is the most stable with zero-value standard deviation. Significantly, tuning the tree-based classifiers increased their stability by decreasing the standard deviation 34% on avg.

Efficiency

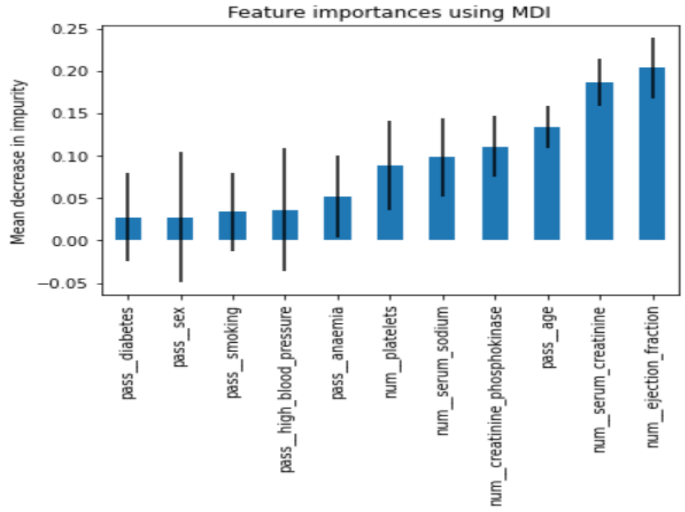
It can be seen in Table 8, that the worst model in terms of efficiency was the tuned AdaBoost (with Logistic Regression as the base estimator) when applied to the standardized, balanced complete dataset. For this condition, the model required a runtime of 1,089 seconds (18 min.), and memory usage of 1.4 MB with a peak of 2.6 MB, to complete. The most efficient algorithm was Logistic Regression completing in 12 secs, and using 0.4 MB memory with a peak of 0.6 MB for the same condition.

Final Project Model

The final project model was developed by applying "SelectFromModel" (SFM) feature with a tuned RF as the base estimator and classifier on a standardized, balanced dataset.

Figure 5 shows the feature importance ranking plot for the complete, balanced dataset using the tuned RF classifier, and for the six best performing features selected by SFM: Ejection fraction, serum creatinine, age, creatinine phosphokinase, serum sodium, and platelets.

Figure 5. *Feature Importance Ranking for the Complete Dataset and Selected Features.*



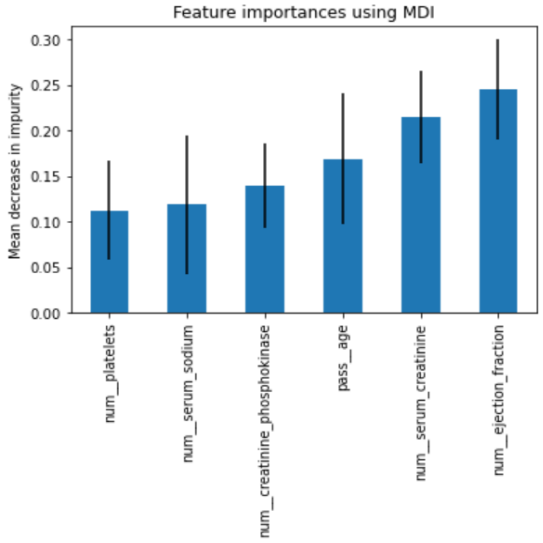


Figure 6 shows the fitting and evaluation of the final model using only the six selected features.

Figure 6. *Final Model Fitting and Evaluating using only Selected Features*

```
Pipeline(steps=[('pre',
                 ColumnTransformer(transformers=[('num', RobustScaler(),
                                                  ['creatinine_phosphokinase',
                                                   'ejection_fraction',
                                                   'platelets',
                                                   'serum_creatinine',
                                                   'serum_sodium']),
                                                 ('pass', 'passthrough',
                                                  ['age', 'anaemia',
                                                   'high_blood_pressure',
                                                   'diabetes', 'sex',
                                                   'smoking'])])),
                ('smote',
                 SMOTEENN(enn=EditedNearestNeighbours(kind_sel='mode'),
                          random_state=0)),
                ('fs',
                 SelectFromModel(estimator=RandomForestClassifier(class_weight='balanced_subsample',
                                                                  criterion='entropy',
                                                                  max_depth=10,
                                                                  max_features='log2',
                                                                  max_leaf_nodes=50,
                                                                  min_samples_split=5,
                                                                  n_estimators=500))),
                ('classifier',
                 RandomForestClassifier(class_weight='balanced_subsample',
                                        criterion='entropy', max_depth=10,
                                        max_features='log2', max_leaf_nodes=50,
                                        min_samples_split=5,
                                        n_estimators=500))])
MCC: 0.877
F1 Score: 0.912
Accuracy: 0.946
Balanced Accuracy: 0.925
Recall: 0.865
Precision: 0.965
AUC-ROC: 0.968
```

Limitations

Generalizing the results of this or any study using the heart failure clinical records dataset should be carefully considered in light of three limitations: first, the small size of the dataset; second, the bias in the dataset that specifically represents a Pakistani demographic; and third, the

male and female subsets in the data. Also there could be an impact from other, yet unconsidered variables.

For my own research, I was limited by my system and how long models could take to complete which kept me from investigating other feature selection techniques such as Relief and LASSO. I would also have benefited from knowing the parameter tuning used by the researchers for the classifiers whether in python or in R, as in the case of the original study.

Ethical Considerations

There are many ethical considerations and accompanying legislation concerning the use of electronic health records (EHRs) in data analysis: patient privacy and confidentiality; informed consent; data security, integrity, and exchange; and, long-term data retention and de-identification.

Conclusion

Table 9 shows the final model results for the replication study (Chicco & Jurman, 2020) compared to my project study.

Table 9. *Replication and Project Study Results for Final Models*

Study	Classifier	Selected Features	MCC	F1 Score	Accuracy	Recall	AUC-ROC
Replication	Random Forest	Serum creatinine, Ejection fraction	0.418	0.754	0.585	0.541	0.698
Project	Random Forest	Ejection fraction, Serum creatinine, Age, Creatinine phosphokinase, Serum sodium, platelets	0.877	0.912	0.946	0.865	0.968

The significant improvements in the metric scores are due to several design features in my project study. First, the application of SMOTE+ENN to balance the dataset, Second,

investigation of the parameters available for tuning of the random forest classifier. And, third not limiting the best performing features for use in the final model to only the best two.

One of the goals of the replication and subsequent studies was to limit the number of major risk factors physicians would have to focus on in predicting survival of heart failure in a clinical setting. The subsequent study by Ishaq et al. (2021) selected the same six features as in my study, to be the most significant based on random forest feature ranking of feature importance. However, all three studies (replication, subsequent, and project) concur that serum creatinine and ejection fraction are the two highest ranked features in determining the event of death.

Furthermore, in the study by Ishaq et al. (2021), the best performance accuracy using select features with SMOTE and ETC was 0.926 which is comparable to my model accuracy of 0.946 using SMOTE+ENN and RF.

In the course of this small study, it was observed how the factors predicting the survival among heart failure patients and the performance of the predictive models have developed and improved using electronic health records starting with statistical reviews in early studies and continuing on to machine learning and deep learning. Furthermore, multimodal machine learning is bringing together and harmonizing disparate data sources to perform algorithmic modeling. The data sources can include genomics, clinical notes, time series, demographics, lab values, and imaging. As these algorithms increase in sophistication and multimodal health care data becomes more prevalent and accessible, it can be expected that precision health will become a viable tool for diagnostic use by healthcare practitioners in a clinical setting.

References

Ahmad, T., Munir, A., Bhatti, S.H., Aftab, M., & Raza, M.A. (2017 July 20). Survival analysis of heart failure patients: A case study. *PLoS ONE 12*(7):e0181001. https://doi.org/10.1371/journal.pone.0181001

Aman. [Unfold Data Science]. (2022 March 29). *Feature Selection Wrapper and Embedded Techniques* [Video]. YouTube https://www.youtube.com/watch?v=za1aA9U4kbI

Brownlee, J. (2021). How to Develop an Extra Trees Ensemble with Python. *MachineLearningMastery.com*. https://machinelearningmastery.com/extra-trees-ensemble-with-python

Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making (2020)*20:16. https://doi.org/10.1186/s12911-020-1023-5

Dubey, A. (2023, April 4). Feature Selection Using Random forest - Towards Data Science. *Medium.* https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f

Ghosh, P., Azam, S., Jonkman, M., Karim, A., Shamrat, F. M. J. M., Ignatious, E., Shultana, S., Beeravolu, A. R., & De Boer, F. (2021). Efficient prediction of cardiovascular disease using machine learning algorithms with Relief and LASSO feature selection techniques. *IEEE Access*, *9*, 19304–19326. https://doi.org/10.1109/access.2021.3053759

Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V., & Nappi, M. (2021). Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques. *IEEE Access (9)*, 39707-39716. https://doi.org/10.1109/access.2021.3064084

Javid, I., Alsaedi, A. K. Z., & Ghazali, R. (2020). Enhanced Accuracy of Heart Disease Prediction using Machine Learning and Recurrent Neural Networks Ensemble Majority Voting Method. *International Journal of Advanced Computer Science and Applications*, *11*(3). https://doi.org/10.14569/ijacsa.2020.0110369

Kline, A., Wang, H., Li, Y., Dennis, S., Hutch, M., Xu, Z., Wang, F., Cheng, F., & Luo, Y. (2022). Multimodal machine learning in precision health: A scoping review. *npj Digital Medicine (2022)*5:171. https://doi.org/10.1038/s41746-022-00712-8

Olson, R. S., La Cava, W., Mustahsan, Z., Varik, A., & Moore, J. H. (2017). Data-driven advice for applying machine learning to bioinformatics problems. *arXiv (Cornell University)*. https://arxiv.org/pdf/1708.05070

OpenAI. (2023). *ChatGPT* (Nov version) [Large language model]. https://chat.openai.com

Satpathy, S. (2023). SMOTE for Imbalanced Classification with Python. *Analytics Vidhya*. https://analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Bagging and Boosting—a method of ensemble learning using Python | smartyR. (n.d.). https://coderspacket.com/bagging-and-boosting-a-method-of-ensemble-learning-using-python

Urbanowicz, R. J., Meeker, M., La Cava, W., Olson, R. S., & Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, *85*, 189–203. https://doi.org/10.1016/j.jbi.2018.07.014

What is undersampling? (2022, August 10). CORP-MIDS1 (MDS). https://www.mastersindatascience.org/learning/statistics-data-science/undersampling

Zahid, F.M., Ramzan, S., Faisal, S., & Hussain, I. (2019 February 19). Gender based survival prediction models for heart failure patients: A case study in Pakistan. *PLoS ONE 14*(2):e0210602. https://doi.org/10.1371/journal.pone.0210602