FYS-STK4155 - Project 3

Frida Larsen

The aim of this project was to perform feature selection on the Ahmad et al heart failure data set [3] in order to identify possible causes of death in patients with left ventricular systolic dysfunction. Three supervised machine learning methods were implemented, namely logistic regression, random forests and support vector machines. Statistically, the models had similar performances with accuracies of approximately 0.75. The models identified ejection fraction and serum creatinine as the most important features, however the random forest model also placed significant importance on platelets.

I. INTRODUCTION

Heart failure occurs when the heart is unable to pump an adequate amount of blood. [1] There are several known causes of heart failure, such as cardiovascular diseases, high blood pressure and genetics. Symptoms of heart failure includes shortness of breath, fatigue and accumulation of liquids in the body (edema). Heart failure is associated with an increased risk of death, however the exact mechanisms are not well understood. [2]

The heart failure data set released by Ahmad et al [3][4] contains medical records of 299 patients with heart failure. The aim of this report is to create models for predicting deaths in the heart failure patients based on their medical records. For this purpose we will use the supervised machine learning methods logistic regression, random forests and support vector machines. In addition, we will perform feature selection in order to determine the most important features resulting in death in patients with heart failure.

This report begins with a methods section, which contains three main parts: a description of the data set and its features, a review of the supervised machine learning methods used in this project and finally a description of the statistical methods used to evaluate the models and perform feature selection via recursive feature elimination. Following the methods section, we present our results in the results section and discuss them in the discussion section. Finally, we draw conclusions based on the analysis.

All relevant code may be found in the GitHub repository 'FYS-STK4155'[1] under the Project3 folder. This folder also includes a Figures folder, which holds all the figures presented in this text and produced during the project.

[1] https://github.com/fridalarsen/FYS-STK4155

II. METHODS

A. The data set

The Ahmad et al heart failure data set consists of medical records of 299 patients. All patients were above 40 years of age and suffered from left ventricular systolic dysfunction, a kind of heart failure where the left ventricle is unable to contract properly, meaning that the heart chamber is not completely emptied during contraction.

A summary of the 13 features in the data set are presented in table I along with their units. Note that for the boolean features, 1 and 0 correspond to True and False respectively. In the case of sex, 1 corresponds to male and 0 corresponds to female.

The features were selected because they traditionally have been known as factors of increased risk of heart failure or other cardiovascular issues. For instance, an anaemic patient has a lack of red blood cells or haemoglobin in the blood, meaning that the blood carries oxygen less efficiently. [5]

Creatine and the associated kinase creatinine phosphokinase (CPK) are compounds found in muscle tissue. [6] The creatine partakes in an effective process for local production of ATP when the energy demand is increased. [7] One of the waste products from this process is creatinine, which is normally secreted by the kidneys. Heightened levels of creatinine in the blood (measured as serum creatinine) indicate an issue with the kidney function of the patient. Heightened levels of CPK indicate muscle tissue damage or loss. [8]

In addition to secretion of creatinine, the kidneys are also responsible for maintaining proper levels of sodium in the blood. Heightened levels of sodium is therefore also an indicator of kidney issues. [9]

The time feature reports the time of death, the time of a patient leaving the study or the time when the study was concluded. In this sense it is not truly an explanatory variable, but rather a response in itself. In this article we will therefore discard this feature, since we have chosen to focus solely on whether the patient survived or not.

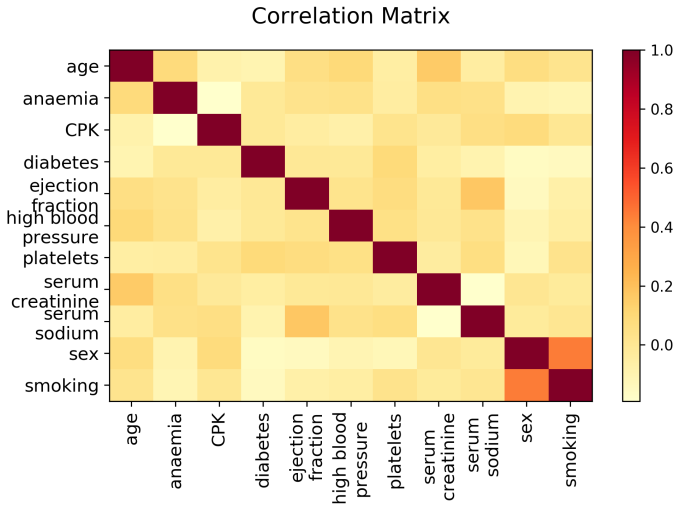


FIG. 1. Correlation matrix of the heart disease dataset.

Figure 1 shows the correlation matrix of the data set (without the time feature). The only pair of features with a noteworthy correlation are smoking and sex.

B. Logistic regression

Logistic regression is a standard classification method. In this project, the logistic regression method will only be presented for binary problems with classes $y = 0$ and $y = 1$. The basic idea is to perform linear regression on a logistic transformation of the probability p of observing $y = 1$:

$$\ln \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k, \tag{1}$$

where k is the total number of input features. If we write $\mathbf{x} = (x_1, \ldots, x_k)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)$ we can solve for p and get

$$p = \frac{e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x}}}{1 + e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x}}} \tag{2}$$

For a particular observation $\mathbf{x_i}$ we predict $y_i = 1$ if $p > \frac{1}{2}$ and $y_i = 0$ otherwise.

In order to determine the regression coefficients β_0 and $\boldsymbol{\beta}$ we aim to minimise the cross entropy cost function[10]:

$$C(\beta_0, \boldsymbol{\beta}) = -\sum_{i=1}^{N} \left(y_i (\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}) - \ln\left(1 + e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}}\right) \right) \tag{3}$$

One may derive cross entropy from the maximum likelihood method by considering the log likelihood (specifically, minimising the negative log likelihood). The cost function is typically minimised using gradient descent methods, which requires the derivative of cross entropy:

$$\frac{\partial C(\beta_0, \boldsymbol{\beta})}{\partial \beta_0} = -\sum_{i=1}^{N} \left(y_i - \frac{e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}}}{1 + e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}}} \right), \tag{4}$$

$$\frac{\partial C(\beta_0, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\sum_{i=1}^{N} \left(y_i \mathbf{x_i} - \mathbf{x_i} \frac{e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}}}{1 + e^{\beta_0 + \boldsymbol{\beta}^T \mathbf{x_i}}} \right) \tag{5}$$

The gradient descent methods are handled internally by `sklearn`. A more detailed description was presented in Project 2. [11]

In addition to cross entropy, it is common to add an L^2-penalty term in order to regularise the size of the coefficients. This is implemented by adding the term $\lambda \|\boldsymbol{\beta}\|^2$ to the cost function. The penalty, λ, must be determined experimentally. One option is to cross validate the performance of models of different penalties (see section II E).

C. Decision trees and random forests

1. *Decision trees*

In a decision tree, a data set is divided and arranged into regions by recursive binary splitting. [12] Decision trees get their name from their tree-like structure consisting of nodes connected by branches. Each node represents a binary split in a single feature in the data set, and each branch represents the result of the split. The initial node is called the root node and the final nodes, where we find the prediction, are known as leaves.

For a node m that represents a region R_m containing N_m observations, the proportion of observations from class k is given by [13]

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \tag{6}$$

A decision tree can be trained using for instance the classification and regression tree (CART) algorithm. At each node m the CART algorithm assigns the following cost function to each feature and threshold pair (k, t_k) [14]:

$$C(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}} \tag{7}$$

where $G_{\text{left/right}}$ is a measure of the impurity of the left and right subsets respectively, while $m_{\text{left/right}}$ is the number of points in each subset. The CART algorithm minimises the total cost function (sum of cost from each node) by minimising $C(k, t_k)$ at each node.

Feature	Explanation	Unit
Age	Age of patient	Years
Anaemia	Whether the patient is anaemic	Boolean
Creatinine phosphokinase	Amount of CPK in the blood	mcg/l
Diabetes	Whether the patient is diabetic	Boolean
Ejection fraction	Amount of blood leaving the heart during each contraction	Percentage
High blood pressure	Whether the patient is hypertensive	Boolean
Platelets	Amount of platelets in the blood	kiloplatelets/ml
Serum creatinine	Amount of serum creatinine in the blood	mg/dl
Serum sodium	Amount of serum sodium in the blood	mEq/l
Sex	Gender of patient	Boolean
Smoking	Whether the patient is a smoker	Boolean
Time	Duration of follow-up period	Days
Death event	Whether the patient perished during follow-up period	Boolean

TABLE I. Explanation of features in the heart failure dataset.

The impurity of a subset is a measure of the quality of the split. There are several ways of measuring node impurity, we have opted to use the gini index, given by [13]

$$G_m = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}) \tag{8}$$

In this project we employ the `sklearn` implementation [15] of decision trees via their implementation of random forest.

2. *Random forests*

The random forest is an ensemble of decision trees. Each tree is an independent classifier working on a bootstrap sample of the data and the forest prediction is the average prediction of all the trees based on a majority vote. [16] In contrast to ordinary decision trees, the trees in a random forest are trained such that each split is based on a random subset of the input variables. [13]

The most important parameter to consider when creating a random forest is the number of trees, which should be sufficiently large (increasing the number of trees beyond this threshold does not improve the model). In addition, the maximum depth of each tree and the number of features used in each split must be predetermined.

D. Support vector machines

The following is heavily based on Hastie et al's book [13] and James et al's book [12].

1. *The formalism of support vector machines*

The main idea of support vector machines (SVMs) is to create a boundary, known as a hyperplane, in feature space that separates data points of different classes. A hyperplane in a p-dimensional space is an affine subspace of dimension $p-1$, given by

$$\mathbf{x}^T\mathbf{w} + b = 0, \tag{9}$$

where $\mathbf{x}$ is a position vector in the space, $\mathbf{w}$ is a parameter vector with $\|\mathbf{w}\| = 1$ describing the hyperplane and b is the intercept term. Points $\mathbf{x}$ in the data set can be classified into two distinct regions based on wether they are above or below the hyperplane, which is determined by the sign of $\mathbf{x}^T\mathbf{w} + b$. Thus for an observation $\mathbf{x_i}$ we classify the data point as $y_i = \pm 1$. The aim of the SVM is to maximise the distance between the hyperplane and the points belonging to each class such that the binary classes are separated by the hyperplane.

For convenience, we define $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + b$ such that the hyperplane is given by $f(\mathbf{x}) = 0$. Since $y_i = 1$ whenever $f(\mathbf{x_i} > 0$ and $y_i = -1$ whenever $f(\mathbf{x_i} < 0$ we must have

$$y_i f(\mathbf{x_i}) > 0 \quad \forall i \tag{10}$$

The idea of SVM is to find the greatest margin $M > 0$ such that

$$y_i f(\mathbf{x_i}) \geq M, \quad i = 1, \ldots, N. \tag{11}$$

It is, however, unreasonable to assume that the points of each class are well separated by a hyperplane with no overlap. This can be handled by introducing a slack variable $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)$ that allows for some points to be on the wrong side of the boundary. The constraint of equation 11 can now be modified to

$$y_i f(\mathbf{x_i}) \geq M(1 - \xi_i) \quad \forall i, \tag{12}$$

with $\xi_i \geq 0$ and $\sum_{i=1}^{N} \xi_i \leq K$, where K is some constant. When $\xi_i > 1$ we have a misclassification. Therefore, by ensuring $\sum_{i=1}^{N} \xi_i \leq K$, the number of allowed misclassifications is automatically limited by K. If we let $M = 1/\|\mathbf{w}\|$ and let $\|\mathbf{w}\|$ vary, we may rephrase the constraint problem as

$$\min \|\mathbf{w}\| \quad \text{subject to} \quad \begin{cases} y_i f(\mathbf{x_i}) \geq 1 - \xi_i \ \forall i, \\ \xi_i \geq 0, \sum \xi_i \leq K \end{cases} \tag{13}$$

In order to solve the constraint problem we must formulate it as a Lagrangian constraint problem. We begin by rephrasing the above condition to

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{subject to } \xi_i \geq 0, y_i f(\mathbf{x_i}) \geq 1 - \xi_i \ \forall i \end{aligned} \tag{14}$$

The corresponding Lagrange function to this constraint problem is

$$\begin{aligned} L_P =& \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \\ &- \sum_{i=1}^{N} \alpha_i [y_i f(\mathbf{x_i}) - (1 - \xi_i)] - \sum_{i=1}^{N} \mu_i \xi_i \end{aligned} \tag{15}$$

where α_i and μ_i are the Lagrange multipliers. Computing the derivatives of $[y_i f(\mathbf{x_i}) - (1 - \xi_i)]$ and $[\xi_i]$ with respect to $\mathbf{w}$, b and each ξ_i yields for all i:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} \tag{16}$$

$$0 = \sum_{i=1}^{N} \alpha_i y_i \tag{17}$$

$$\alpha_i = C - \mu_i \tag{18}$$

Substituting these into equation 15 we get

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j} \tag{19}$$

This equation is maximised for $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$, in addition to the Karush-Kuhn-Tucker conditions:

$$\alpha_i [y_i f(\mathbf{x_i}) - (1 - \xi_i)] = 0, \tag{20}$$

$$\mu_i \xi_i = 0, \tag{21}$$

$$y_i f(\mathbf{x_i}) - (1 - \xi_i) \geq 0, \tag{22}$$

for $i = 1, \ldots, N$. This constraint problem can be solved using convex quadratic programming methods, which in this project is handled internally by `sklearn`.

2. Kernel methods

So far we have only considered linear boundaries in the feature space. However, the SVM can be made more flexible by allowing different sets of basis functions. By introducing basis transformations $h(\mathbf{x_i}) = (h_1(\mathbf{x_i}), h_2\mathbf{x_i}, \ldots, h_M(\mathbf{x_i}))$ for $i = 1, \ldots, N$, we may follow the same procedures as before using $h(\mathbf{x_i})$ as points in the feature space, i.e. replace $\mathbf{x_i}$ with $h(\mathbf{x_i})$. Going through the process yields the following Lagrange function:

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle h(\mathbf{x_i}), h(\mathbf{x_j}) \rangle \tag{23}$$

Because this equation only depends on the inner product of $h(\mathbf{x_i})$ and $h(\mathbf{x_j})$, its suffices to specify the kernel function

$$K(x, x') = \langle h(x), h(x') \rangle. \tag{24}$$

For this project, the following kernels are implemented:

$$\text{Polynomial kernel: } K(x, x') = (1 + \langle x, x' \rangle)^d \tag{25}$$

$$\text{Radial kernel: } K(x, x') = \exp\Big(-\gamma \|x, x'\|^2\Big), \tag{26}$$

where γ is a parameter. We use the `sklearn` default value.

E. Evaluating the models

The standard way of evaluating classifiers (and any supervised machine learning algorithm in general) is to divide the data set at random into a a training set and testing set. The training set is used to prepare the classifiers, which are then evaluated using the testing set. Typically, most of the data is used for training. In order to compare the classifiers, the accuracy classification metric is used. Accuracy is defined as the number of correct predictions normalised by number of samples:

$$\text{Accuracy}(y, \hat{y}) = \frac{1}{N}\sum_{i=0}^{N-1} 1(\hat{y}_i = y_i) \tag{27}$$

where N is the number of samples, y is the observed response and $\hat{y}$ is the predicted.

The data set used in this project contains only 299 points, which means that the testing set is very sensitive to the random selection. In order to get a good estimate of the accuracy we implement K-fold cross validation. The main idea is to split the data set into K folds (subsets) which are then iteratively used as testing sets (using the remaining points as training data). The algorithm is as follows:

Algorithm 1 K-fold Cross Validation

1: Split data into K folds at random
2: **for** $k = 1, 2, \ldots, K$ **do**
3: The kth set is used as testing data and the remaining points are used as training data
4: The model is fitted and validated, the accuracy is calculated
5: **end for**
6: The average and standard deviation of the calculated accuracies are computed

The average accuracy is used as an estimate of the true accuracy, with the standard deviation as an estimate of the uncertainty. In the data set used in this project, there are 203 patients that survived and 96 patients that died. This means that a randomly selected fold will typically contain more surviving patients than dead patients. We might even have a situation where a fold contains no dead patients at all. In order to ensure that both classes are represented in every fold, we use a stratified version of K-fold cross validation that splits the data such that each fold contains roughly the same percentage of dead patients.

In this project, we use cross validation as a tool for optimising the models' parameters. In the logistic regression case, we use cross validation to determine the optimal penalty λ. For random forests we use cross validation to determine a sufficient number of trees, and the best combination of tree depth and number of features per split. Finally, for support vector machines, we use cross validation to determine the optimal cost C for each kernel.

F. Feature selection

The basic idea of feature selection is to determine what features have the greatest impact on a classification model. This will provide an indication of wether the input features are related to the outcome. In our case, we want to determine what medical records are most correlated with survival in order to gain a better understanding of the underlying mechanisms of heart failure deaths.

As opposed to logistic regression and support vector machines, random forests have an inherent feature selection ability based on averaging the importance of the features in each tree. In particular, the feature importance computes the total gain in the gini index of each tree when a feature is removed completely from the forest.

A general-purpose method for feature selection is recursive feature elimination (RFE). RFE is based on a selection criterion that identifies the least important feature in the model. The idea is to measure the performance of the full model and recursively remove the feature that is determined to be least important via the selection criterion. For random forests, the selection criterion is based on the inherent feature importance measure. There is no equivalent natural way of determining the selection criterion for logistic regression and support vector machines. A common choice is to rank the features according to the square of their coefficients. In this project, we will average the performance at each step over 100 repetitions. RFE selects the most important features by choosing the smallest set of features for which the average performance is within the confidence interval of the performance of the full model. In this project, the confidence interval is taken as within one standard deviation of the average accuracy of the full model.

III. RESULTS

A. Model optimisation

Figure 2 shows the 5-fold cross validated accuracies for different penalties in the logistic regression model. Overall, there are no significantly large improvements to be made by tuning the penalty. However, there is a local maximum at $\lambda = 0.1$, which is the value we will use in the feature selection process. While tuning the parameter does not significantly improve the model, a large penalty ($\lambda \geq 10^2$) limits the accuracy to approximately 0.7. Figure 3 shows the confusion matrix of the logistic regression model when using $\lambda = 0.1$. We see that the largest contribution to the accuracy comes from surviving patients (0) that are correctly labeled. From the lower row, we see that most dead patients are predicted to survive. Using the optimal penalty $\lambda = 0.1$, the logistic regression model achieves an accuracy of 0.746 ± 0.041 (5fold cross validated).

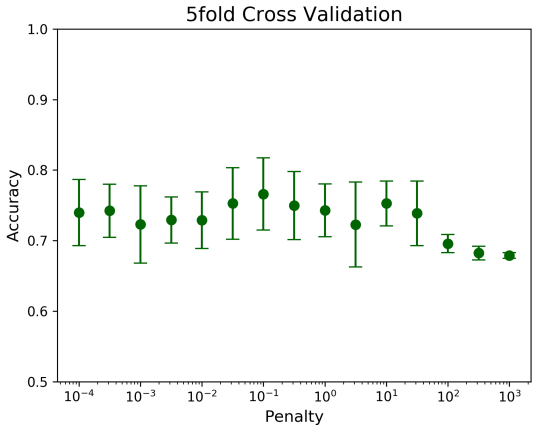


FIG. 2. Cross validated accuracies of different penalties for logistic regression.

Figure 4 shows the 5-fold cross validated accuracies for different number of trees in a random forest model. There is no significant improvement to the accuracy for a number of trees higher than 50, and this is the number of trees we will take as being sufficient. Figure 5 shows the 5-fold cross validated accuracies of different combinations of maximum tree depth and number of features at each split for a random forest model with 50 trees. There is no clear pattern for determining the optimal combination, although the best result was achieved with a maximum depth of 15 and maximally 7 features per tree. Figure 6 shows the confusion matrix for the best random forest model. Similar to the logistic regression confusion matrix, the random forest confusion matrix shows that the model is worse at predicting patient deaths compared to survivals. Using the optimal choice of a maximum tree depth of 15 and a maximum of 7 features per tree, the random forest achieves an accuracy of 0.756 ± 0.022.

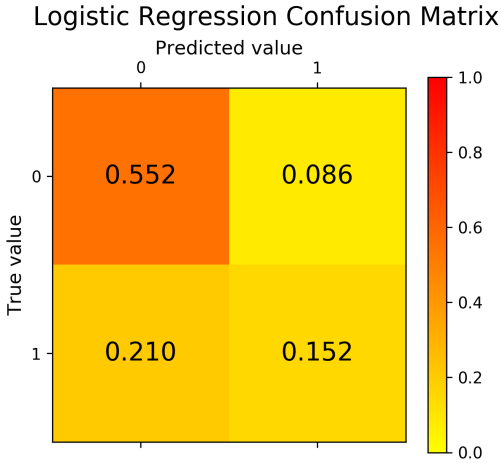


FIG. 3. Confusion matrix for best logistic regression model.

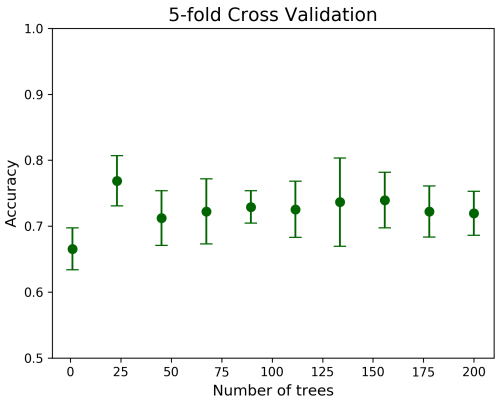


FIG. 4. Cross validated accuracies for different number of trees in a random forest.

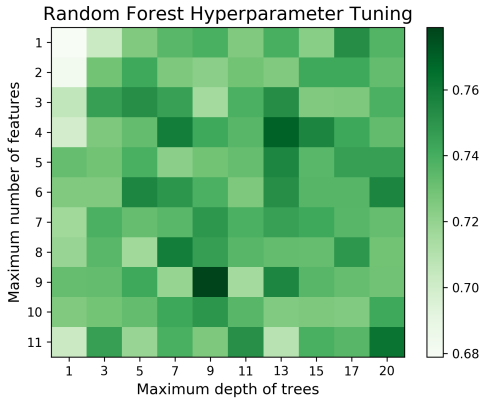


FIG. 5. Cross validated accuracies for combinations of hyperparameters for random forest.

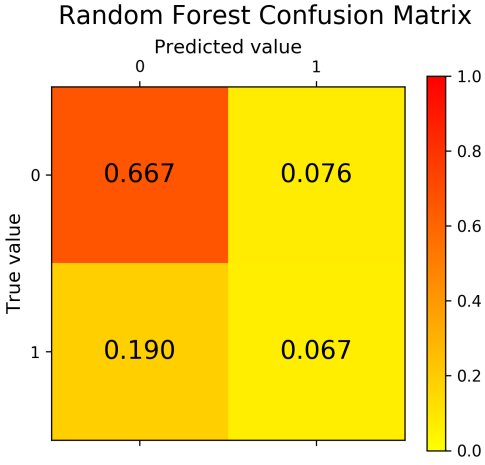
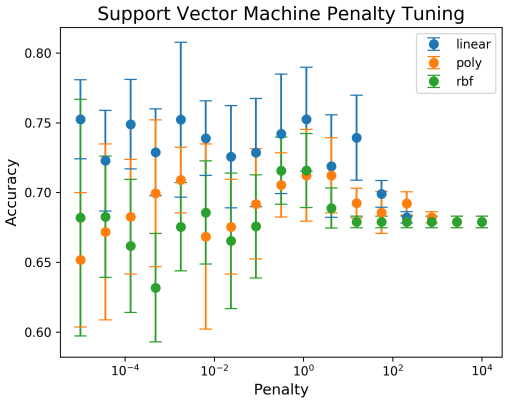


FIG. 6. Confusion matrix for best random forest model.



t

FIG. 7. Cross validated accuracies for different penalties of the support vector machine.

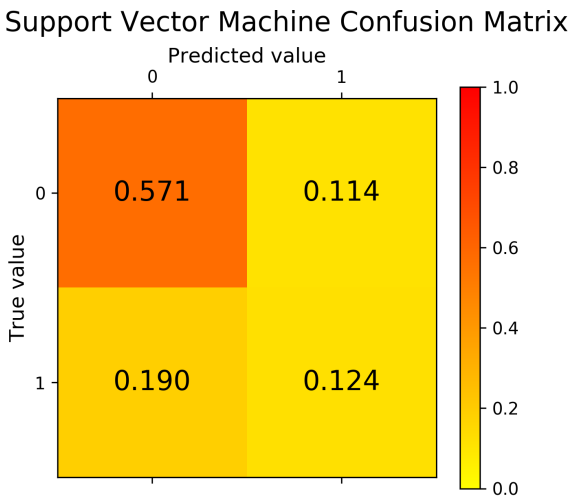


FIG. 8. Confusion matrix for best support vector machine model.

Figure 7 shows the 5-fold cross validated accuracies for different penalties for a support vector machine model using three different kernels. The linear kernel consistently performs better than the other kernels for all penalty choices. Therefore the linear kernel was chosen for performing feature selection. For the linear kernel there are, as for the logistic regression model, no significant improvements to be made by changing the penalty. However, we see that in this case also a large penalty limits the accuracy to just below 0.7. Figure 8 shows the confusion matrix of a support vector machine model using the linear kernel and the best penalty determined from figure 7. Similarly to the other models, the support vector machine has the largest contribution to the accuracy from correct prediction of surviving patients, whilst most dead patients are labeled incorrectly. The support vector machine achieves an accuracy of 0.752 ± 0.050 using the linear kernel and the corresponding optimal penalty $\lambda = 1$. In contrast with logistic regression and random forest, the support vector machine required considerably more time to train.

B. Feature selection

Figure 9 shows the feature importance measures from the random forest model. The measures show that there are some features of little importance and some features of high importance. The less important features are smoking, diabetes, anaemia, high blood pressure and sex, while the most important features are serum sodium, platelets, CPK, age, ejection fraction and serum creatinine. Notably, the most important features all have comparable importances (meaning that they are in the same order of magnitude).

The results of the recursive feature elimination procedures for all the models are presented in figures 10, 11 and 12. All three models agree that using 2-3 features can reproduce the results of the complete models, although they don't necessarily agree on the order of feature removal. Notably, the random forest model requires 3 features in order to reproduce the full model accuracy whilst the linear regression and support vector machine models only require 2.

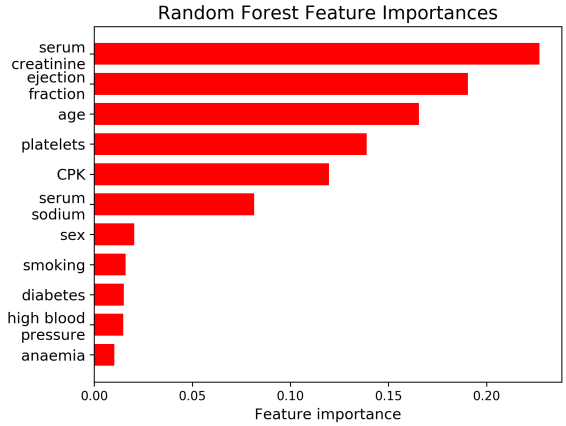


FIG. 9. Feature importances in the random forest model.

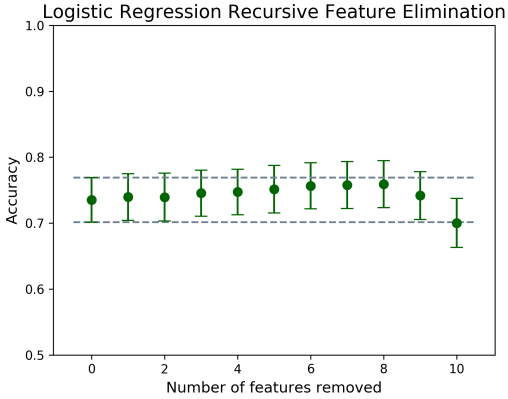


FIG. 10. Results of recursive feature elimination for logistic regression. Order of removal: platelets, diabetes, serum sodium, creatinine phosphokinase, smoking, anaemia, sex, high blood pressure, age, ejection fraction and serum creatinine.

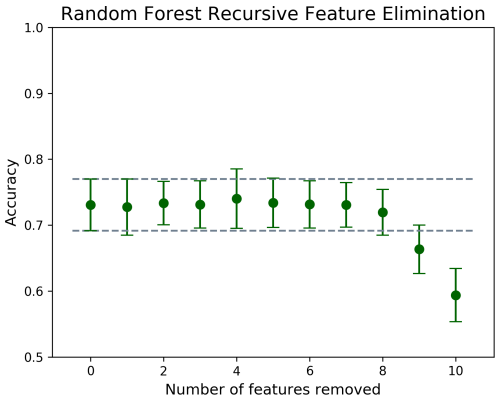


FIG. 11. Results of recursive feature elimination for random forests. Order of removal: smoking, diabetes, anaemia, sex, high blood pressure, serum sodium, creatinine phosphokinase, age, ejection fraction, serum creatinine and platelets.

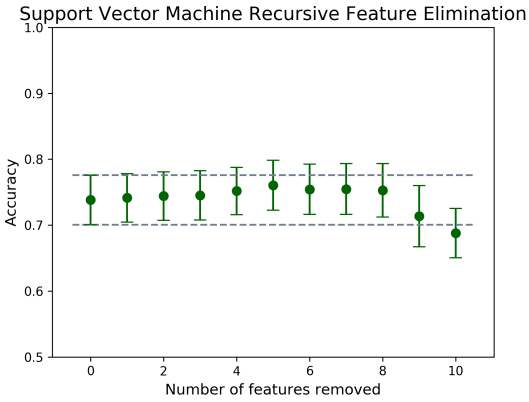


FIG. 12. Results of recursive feature elimination for support vector machines. Order of removal: platelets, diabetes, creatinine phosphokinase, anaemia, serum sodium, smoking, high blood pressure, sex, age, ejection fraction and serum creatinine.

IV. DISCUSSION

For both the logistic regression model and the support vector machine, figures 2 and 7 show that tuning the penalty parameter had little effect on the overall performance of the models. In both cases, the accuracy stabilises for higher penalties and varies for lower penalties, although the variation is much greater for the support vector machine. This is unexpected from the point of view of a bias-variance tradeoff analysis as one would expect there to be an optimal region for the penalty. Interestingly, the linear kernel outperformed the other kernels in the support vector machine. This is also unexpected, as it is less flexible than the other kernels. It is also worth mentioning that the time required to train a support vector machine was considerably higher than that of logistic regression and random forest.

For random forests, figure 4 shows that increasing the number of trees beyond 50 led to no significant improvement in the model accuracy. This is lower than typical values which are of magnitude 10^2. [13] Interestingly, tuning the hyperparameters of the decision trees in the forest showed no clear patterns of improvement; figure 5 shows that using the optimal choice of hyperparameters does not outperform the default parameters used in figure 4.

Overall, all the models achieved a similar accuracy (≈ 0.75). In addition, the confusion matrices (figures 3, 6 and 8) show that the largest contribution to the accuracy came from correctly predicting surviving patients. Out of the dead patients, a greater fraction were predicted to survive in all the models. The models' failure to accurately predict patient deaths likely stems from the imbalance in the data set (out of the 299 patients, only 96 patients died).

There is an interesting difference between the feature importance measures of the random forest (figure 9) and the results of the recursive feature elimination procedures (figures 10, 11 and 12). Judging by the feature importances, a larger amount of features should be important to the model compared to what we found using recursive feature elimination. Also, the importance measures fall off exponentially whereas the recursive feature elimination shows a discontinuous distinction between the important and the unimportant features. The main difference between the two methods is that the feature importance measures were determined based on all the features, whilst the recursive feature elimination considers a decreasing set of features. The discrepancy between the two results indicates that some of the features contain some of the same information without being correlated. For instance, high levels of serum sodium or serum creatinine both indicate kidney issues, which can increase the likelihood of mortality. However, the discrepancy may also be explained by how the random forest treats the data differently from logistic regression and support vector machines. Whereas random forests use hard cut-offs to make binary splits in the feature space, both logistic regression and support vector machines consider continuous transformations of the data. The results of this can be seen in the recursive feature elimination procedure for the random forest (figure 11), which is unlike the other two in the sense that it requires a greater number of minimum features to achieve an accuracy similar to the complete model.

In the recursive feature elimination, the order of removal for the logistic regression model and the support vector machine agree almost exactly (see figure 10 and 12). Importantly, the two methods agree on the three most important features: age, ejection fraction and serum creatinine. In addition, the models also agree that ejection fraction and serum creatinine are sufficient for reproduc-

ing the accuracy of the full models. It is reasonable for ejection fraction to be one of the most important features for this patient group, since they all suffer from left ventricular systolic dysfunction. In a sense, the ejection fraction is a measure of the severity of the patients heart failure. It is also reasonable for serum creatinine to be one of the most important features, since the serum creatinine levels are an indicator of kidney health. The random forest recursive feature elimination has ejection fraction, serum creatinine and platelets as the three most important features (see figure 11). This is in contrast to logistic regression and the support vector machine which considered platelets to be the least important feature. However, this is another argument supporting the claim that the features share information other than a direct correlation. Moreover, the importance of platelets is noticeable in the feature importance measures of the random forest (figure 9), which explains how platelets was not removed during the early stages of the recursive feature elimination procedure.

V. CONCLUSION

The aim of this project was to use supervised machine learning methods for identifying the most important features in the survival and death of heart failure patients in order to provide an indication of possible causes of death in this patient group. The features considered stem from medical records of 299 heart failure patients as collected by Ahmad et al. [3] The project considered three supervised machine learning methods: logistic regression, random forests and support vector machines. Using the optimal hyperparameters for each model (as determined by 5fold cross validation), the models achieved an accuracy of 0.746 ± 0.041, 0.756 ± 0.022 and 0.752 ± 0.050, respectively. As these are all within the confidence intervals of each other, there is no reason to prefer a particular model. However, the support vector machine required considerably more time to train, thus from an efficiency point of view it is recommended not to use this method for similar data. Moreover, each model was most successful in predicting survival. Tuning the hyperparameters of the methods had little to no effect on the overall performance. For feature selection, recursive feature elimination was used. Each method found that 2-3 features were sufficient for reproducing the accuracy of the full model. All the models found ejection fraction and serum creatinine to be among the top three most important features. However, random forests placed significant importance on platelets, which logistic regression and the support vector machine considered to be the least important feature.

The models used in this project all had a similar performance and failed to classify the same group of patients. Future work is needed to understand the relationship between this patient group and the models. In particular, whether the models fail to classify the patients for the same reason, e.g. lack of information in the data or outlier effects. In addition, it could be of interest to investigate a larger and more balanced data set, potentially with more features. Since each model achieved similar results, there is reason to believe that another supervised machine learning method would not be able to achieve a higher accuracy.

[1] Kolbjørn Forfang. Hjertesvikt. Store medisinske leksikon, 2016. Accessed 14.12.2020 from Store medisinske leksikon (Norwegian).

[2] Renate Alsén Øvergård. Høy dødelighet ved akutt hjertesvikt. forskning.no, 2013. Accessed 14.12.2020 from forskning.no (Norwegian).

[3] Tanvir Ahmad et al. Survival analysis of heart failure patients: A case study. *PLoS ONE*, 12(7), July 2017. DOI.

[4] Heart failure prediction. Kaggle, 2020. Accessed 30.11.2020 from Kaggle.

[5] Anaemia. WHO Health Topic. Accessed 14.12.2020 from the World Health Organisation.

[6] Kreatin. Store medisinske leksikon, 2020. Accessed 14.12.2020 from Store medisinske leksikon (Norwegian).

[7] Wenche Frølich et al. Assessment of creatine in sports products. Norwegian Scientific Committee for Food Safety (VKM) report, 2010. Accessed 14.12.2020 from VKM.

[8] Davide Chicco and Giuseppe Jurman. Macine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, 20(16), 2020. DOI.

[9] Per Holck. Nyrene. Store medisinske leksikon, 2019. Accessed 15.12.2020 from Store medisinske leksikon (Norwegian).

[10] Morten Hjorth-Jensen. Logistic regression. Lecture slides, Sep 2020.

[11] Frida Larsen. Fys-stk4155 - project 2. Report, 2020. Can be accessed from GitHub.

[12] James et al. *An Introduction to Statistical Learning*. Springer, 2013.

[13] Hastie et al. *The Elements of Statistical Learning*. Springer, 2 edition, 2009.

[14] Morten Hjorth-Jensen. From decision trees to bagging methods. Lecture slides, Nov 2020.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] Morten Hjorth-Jensen. Random forests and boosting. Lecture slides, Nov 2020.