ML Report Group 50

Shivam Badal (2830263), Rose Al-Qarwani (2692218), Amla Malltezi (2757647)
Bedirhan Salbacak (2778875), Kareem Mardini (2783070)

March 2024

Abstract

In this paper, four classification models are being compared to identify heart disease using a dataset from Kaggle. This dataset combines data from five separate datasets which have eleven features in common. The four models that will be compared are: Logistic Regression, Support Vector Machines, Neural Network and K-nearest Neighbours. These models were trained and tested using Python with their associated libraries. Neural network performed the best in terms of accuracy, while the kNN model displayed the highest recall value.

1 Introduction

According to the World Health Organization, heart disease stands as the leading cause of death worldwide, responsible for an estimated 16% of the world's total deaths in 2019. This paper leverages a dataset from Kaggle, which is a platform for competitions regarding predictive modeling.[1]. To determine which machine learning model is the most effective in predicting heart disease, this paper seeks to answer the following research question:

What is the comparative performance of various machine learning algorithms in predicting mortality by heart disease on this dataset?

The results will help enhance predictive healthcare. By means of this comparative analysis, the research aims to improve the quality of care for individuals who are at risk, as well as the early detection of cardiac disease, with potential to save lives.

2 Methodology

2.1 Data inspection & preparation

2.1.1 The dataset

The dataset available from Kaggle is a combination of five previously independent datasets which were not combined before[1]. After removal of the duplicated observations, the final dataset consisted of 918 observations, which contained eleven features.

2.1.2 Preparation of the dataset

While analysing the dataset, it was observed that the *Fasting Blood Sugar* feature contained a high number of zero values, with 704 out of the 918 observations reporting a value of zero. A normal value for fasting blood sugar would be around 3.9 to 5.4 mmols/l (70 to 99 mg/dl), the value being zero in 704 out of the 918 observations is implausible and has a high likelihood of being an error in this dataset. Since it might affect the performance of the several machine learning algorithms in a negative way, it was decided to exclude the *Fasting Blood Sugar* feature from the dataset.

It was also observed that the *Oldpeak* feature reported thirteen negative values. These are potential data entry errors, because negative observations for *Oldpeak* are not typical. These values were transformed to a positive value by applying the absolute value operation.

Furthermore, to prepare the data for machine learning analysis it was needed to transform some categorical features into numerical values. Using one-hot encoding, each categorical value was converted into a new binary variable (0 or 1). As a result this expanded our feature set.

2.1.3 Statistical analysis

Using the describe function from the Pandas library in Python, the data was analyzed, focusing on key features that are most relevant to our study of heart disease. After using one-hot encoding, the feature set expanded by a lot. The following highlights represent only a selection of the most critical features from Table 1:

1. **Age:** has a broad range from 28 - 77 years of age with a mean of approximately 53.5 years. This means that the dataset contains primarily middle-aged patients, which shows a correlation between age and having heart disease.
2. **Chest Pain Type:** the most common chest pain type in this dataset was reported to be asymptomatic chest pain. This is a type of chest pain which can be indicative of silent cardiovascular conditions, where a patient doesn't exhibit classic symptoms of heart disease[2].
3. **Resting Blood Pressure:** the mean resting blood pressure is 132.54 mmHg. This feature could be significant, because a high level of resting blood pressure is a known risk factor for heart disease[3].
4. **Cholesterol:** a high level of cholesterol is a known critical heart disease. This dataset offers a great range of cholesterol values from 85 to 603 mg/dL, which could indicate the presence of a patient in significant risk[4].
5. **Max Heart Rate:** the maximum heart rate achieved can be an indicator of heart disease problems. With a mean of 136.81 BPM and a range from 60 to 202 BPM. This feature might provide insights into a patients potential risk for heart disease.
6. **Heart Disease:** This feature reveals that approximately 55% of the dataset's observations are cases with heart disease. This represents a relatively balanced distribution between instances of heart disease and not having a heart disease. The dataset is still leaning towards more cases of heart disease.

2.2 Model selection

When selecting the machine learning models for predicting heart disease, the aim was to look at the unique strengths and challenges of each option while still aiming for a diverse selection. The chosen machine learning models are:

1. **Logistic regression:** is an efficient and powerful way to analyze the effect of a group of independent variables on a binary outcome by quantifying each independent variable's unique contribution. Regression techniques are versatile in their application and use in medical research, since they measure associations and predict outcomes. This algorithm might be highly effective for predicting heart disease[5].
2. **K-nearest neighbors:** which is known as one of the simplest non parametric classifiers. It assigns a new observation to the class of majority of the k nearest neighbors. This algorithm can be highly effective when not in a high dimensional setting[6].
3. **(Artificial) Neural network:** aims to simulate the structure and function of the human brain by using multiple layers of interconnected neurons. This is highly effective for processing and analysis of large or complex datasets. They are able to capture complex patterns which makes them desirable for heart disease prediction[7].
4. **Support vector machine:** is known to be a high performing algorithm in many biomedical fields, especially in bio informatics. Contrary to logistic regression, which predicts the occurrence of a binary event fitting data to a logistic curve, SVM discriminates between two classes by generating a hyperplane that separates classes after the input data has been transformed. This makes it a highly effective and diverse algorithm for predicting heart disease [8]

2.3 Training and testing approach

Given the dataset's limited and small size of 918 observations, implementing a conventional 80/20 train-test split might not be the most appropriate approach. As it can lead to a training set which could be too small, which is not optimal.

For this paper cross-validation will be used, specifically (stratified) k-fold cross-validation. This allows us to use all parts of the data as testing data by dividing our dataset in k folds, allowing us to better evaluate the performance of our model. K-fold cross validation ensures that each fold is a good representative of the dataset, it being stratified means that the percentage of patients with and without heart disease stays the same within each fold. By iterating through every fold as a test set, this method gives us a good insight into the model's performance, which will mitigate the risk of over fitting or under fitting.

The choice of the k value in (stratified) k-fold cross validation is pivotal. Gravitating towards a value of k=5 or k=10 is considered common practice. For this experiment a k-value of six will be used.

2.4 Performance metrics

The choice of performance metrics is important in assessing the effectiveness of machine learning algorithms, especially when there are several algorithms with specific characteristics. Different machine learning algorithms may excel in different areas. Comparing performances using a single metric might not fully capture the performance of some machine learning models. To address this, different performance metrics will be used:

1. **Roc auc:** this metric provides a measure of the model's performance across all classification thresholds. It is particularly useful for evaluating the true positive rate against the false positive rate.
2. **Precision:** this metric measures the ratio of true positives to the sum of true and false positives. This metric is crucial where the cost of a false positive is high.
3. **Recall:** this metric measures the model's ability to recall and identify all relevant instances within the dataset. For heart disease prediction this might be particularly important, since a missing positive case might have serious implications.
4. **F1 Score:** this metric provides a balance between precision and recall, offering a single metric that summarizes the model's accuracy in identifying positive instances while minimizing false positive.

Each metric will check different aspects of the model's predictive capabilities, which allows for a better comparison of the various machine learning algorithms which will be applied to our dataset.

3 Experiment Setup

3.1 Model 1: K-nearest Neighbours

K-nearest neighbours (KNN) is known for its straightforward yet effective approach in classification scenarios. It operates by identifying the k nearest data points or neighbors to a new instance, using majority voting for classification tasks or average value calculation for regression. The selection of k, representing the number of nearest neighbors, is pivotal for KNN's performance. For this study, 'k' is empirically set to seven, as suggested by an iterative process for maximizing classification accuracy, a methodology which is mentioned in "KNN Model-Based Approach in Classification" by Guo, Gongde et al[9]. By running the model and enumerating in the k-values from $k = 1, 2, 3, ...20$, reflecting the methodology where k is varied for distinct data to identify an optimal value in terms of classification precision. Mathematically, KNN's functionality is depicted as follows [9], given a data point x, KNN locates the k closest points in the training set, symbolized as $\{x_1, x_2, \ldots, x_k\}$. For classification, the predicted label $\hat{y}$ is determined by the mode of the k nearest points' labels, expressed as $\hat{y} = \text{mode}\{y_1, y_2, \ldots, y_k\}$, where y_i represents the label of x_i.

3.2 Model 2: Logistic regression

A logistic regression model was chosen primarily due to its wide range of applicability in classification tasks. The nature of this issue requires it to be viewed from many angles, thus a regression model serves as an approach both easy to implement, and very accurate in its results, particularly for a task with features so strongly correlating the target labels. For this reason, a regression model is useful not only in its own predictions, but also to serve as a comparison to other models, offering a clue into whether a model might be overfitting the data. For this domain, a logistic regression model was chosen over a linear one, considering the target labels are binary. A custom model was built for the purpose of this experiment, as it enables closer monitoring of the performance aspects, especially during training. In other words, it allows for easier debugging. Moreover, it simplifies the calculations of the loss and statistical testing, considering values such as log likelihoods are stored as class attributes, and therefore updated during training and testing.

Training Regression models generally adhere to the following relationship [10]:

$$Y = wX + b + \epsilon \tag{1}$$

Where X and Y represent the instances and result respectively, w represents the weights added to each input, b is a bias term, and ϵ represents some factor of random noise. In this case, the random noise was omitted for simplicity, leading to the following relationship [10]:

$$\sum y = \sum wx + b \tag{2}$$

In logistic regression, a sigmoid function is typically used to ensure the output values lie between 0 and 1 [11]:

$$F(x) = \frac{e^x}{1 + e^x} \tag{3}$$

This was implemented using a custom class in python, with the training procedure as follows. Initially, the weights and bias terms were all set to zero. During the training, matrix multiplication was used to define part of the model – a term corresponding to wx – which was then passed through a sigmoid function to ensure the values lie within an acceptable range. The gradient loss was then calculated for both the weights and bias, and each value was updated accordingly. Specifically, for the weights, the incorrect classifications were matrix-multiplied by each instance and averaged over the dataset, whereas the bias term neglected the matrix multiplication:

$$GradientforWeights = \sum \frac{1}{n} x * (y - t) \tag{4}$$

$$GradientforBias = \sum \frac{1}{n} (y - t) \tag{5}$$

3.3 Model 3: Neural network

The initial hyperparameter under consideration is the number of neurons in the hidden layer. The neurons explored will include: 1, 10, 64, 128 and 256. Using Tanh as the activation function in the hidden layer, without incorporating dropout, reveals that 256 neurons in the hidden layer gives the best performance, as detailed in Table 2.

Moving on to the next hyperparameter, the focus will be on the optimal activation function. The three most common activation functions are: Relu, Sigmoid and Tanh. The use of 256 neurons in the hidden layer (identified before as the optimal number of neurons) gives the result of Tanh performing the best, which can be seen in Table 3.

The final hyperparameter for this experiment will be the ideal number of epochs for the neural network. Looking at the plotted loss curve in Figure 1 reveals that the loss doesn't decrease that much after the third epoch, leading to the selection of an epoch amount of three.

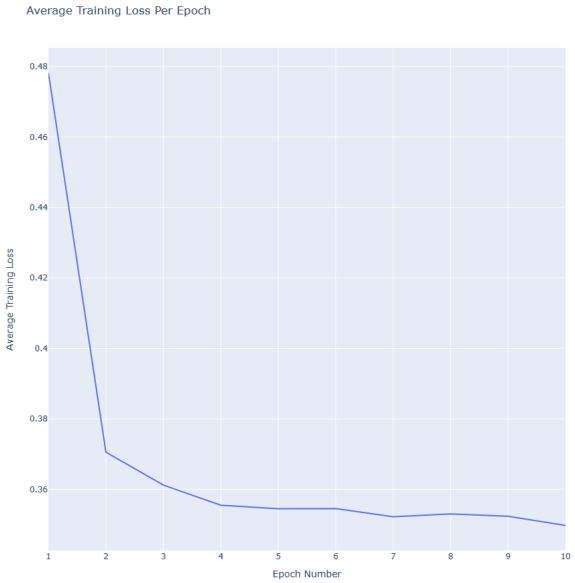


Figure 1: Loss curve for neural network

3.4 Model 4: Support vector machine

The support vector machine model is selected for being efficient in handling high-dimensional feature spaces, contrary to KNN, effectively finding the clearest division between different classes by maximizing the margin, which is crucial for its predictive accuracy later.[12]

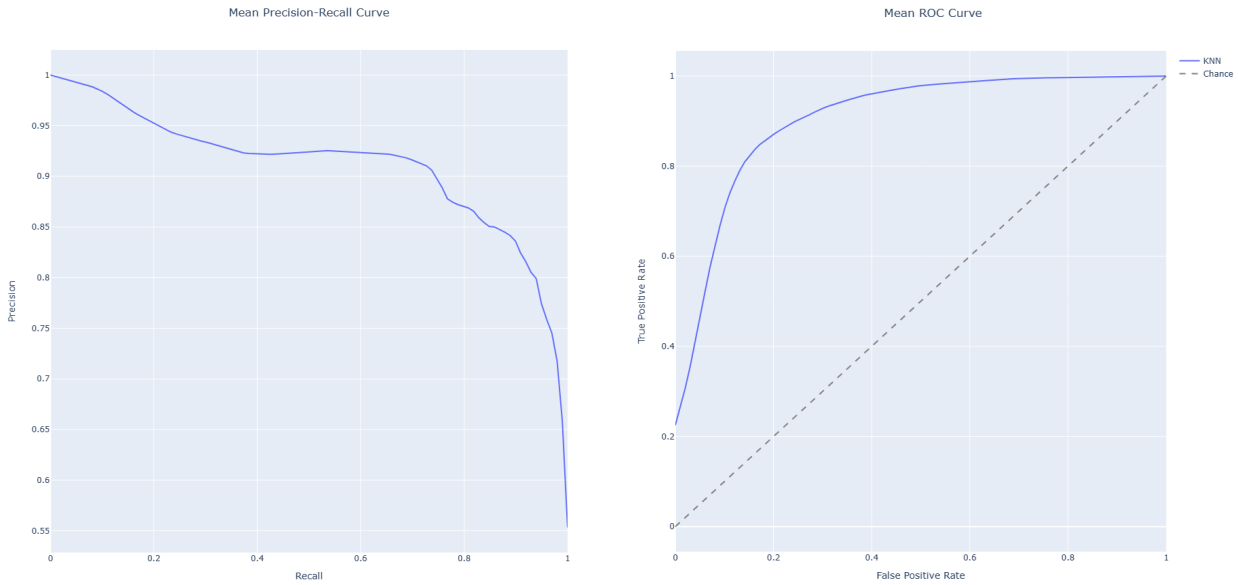
Fine-tuning the hyperparameters, particularly C and gamma in this case, is crucial in SVM's configuration. The C parameter is known to balance model complexity and generalization capability, aiding with preventing overfitting. The choice of the kernel, notably the RBF, influences the transformation of data into a higher-dimensional space, facilitating class separability.[13]

4 Results

4.1 Model 1: K-nearest neighbours

Since the KNN model was ran for each cross validation fold (six times), the scores were saved and the mean accuracy was computed. The results can be seen in Table 4, while a visual representation of the results can be seen in Figure 2

- **Roc auc:** The mean accuracy is 90.7%, which is quite high. This score showcases KNN model's robust performance.
- **Precision:** This score indicates the accuracy of positive predictions, which has a mean accuracy of 83.3%. This highlights the model's performance in accurately predicting positive predictions.
- **Recall:** The mean accuracy is 89.9%, which means it can identify all the actual positives with a high accuracy.
- **F1 Score:** The mean accuracy is 86%, which means that it has a fairly high accuracy depicting the balance between precision and recall.



(a) Mean precision recall curve

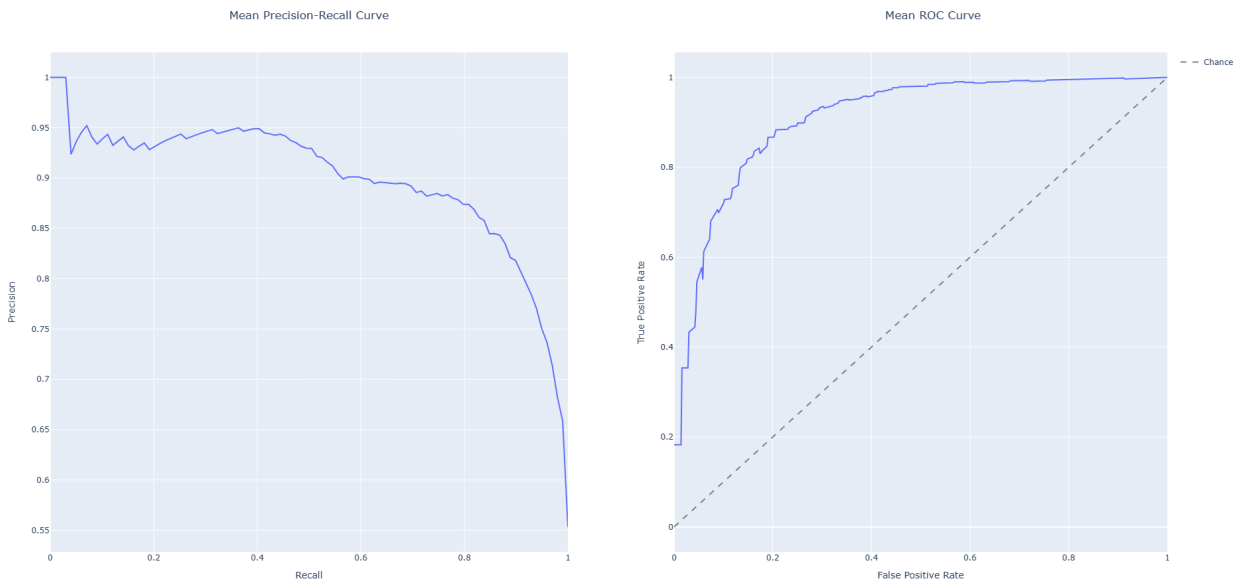
(b) Mean roc curve

Figure 2: Mean of Precision Recall and Roc curve depicting performance across 6 folds

4.2 Model 2: Logistic regression

Figure 3 depicts the precision-recall and Roc curve over six folds. These plots demonstrate the performance of a custom-built logistic regression model. Unlike advanced implementations found in libraries like scikit-learn, this model is constructed from scratch, leading to noticeably jagged curves. More sophisticated libraries employ various techniques to smooth these curves out, but for the purpose of comparison and analysis the current level of detail in the curves will suffice. As with all machine learning algorithms discussed in this paper, the scores were saved and the mean was computed for each metric, which can be seen in Table 5:

- **Roc auc:** The mean accuracy is 90%, which is quite high. This means that the model can distinguish postive and negative classes well between each other.
- **Precision:** This score indicates the accuracy of positive predictions, which has a mean accuracy of 83%.
- **Recall:** The mean accuracy is 89%, which means it can identify all the actual positives with a high accuracy.
- **F1 Score:** The mean accuracy is 86%, which means that it has a fairly high accuracy depicting the balance between precision and recall.



(a) Mean precision recall curve

(b) Mean roc curve

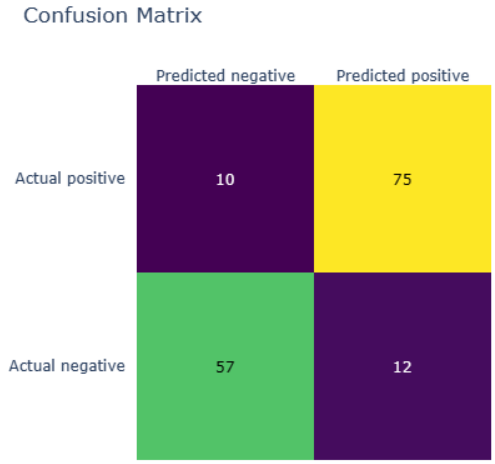
Figure 3: Mean of Precision Recall and Roc curve depicting performance across 6 folds

4.3 Model 3: Neural network

Figure 4 shows us the confusion matrix and a table depicting the performance of the trained neural network. To provide more clarification:

- **Roc auc:** This score indicates that it can distinguish well between the positive and negative classes. The mean accuracy is 91%.
- **Precision:** This score indicates the accuracy of positive predictions, which has a mean accuracy of 86.7%
- **Recall:** This score indicates the ability the different models abilities to correctly identify all the actual positives, which has a mean accuracy of 88.6%
- **F1 Score:** This score indicates a balance between precision and recall, suggesting that the different models are reliable in terms of accuracy and completeness in the positive predictions. It has a mean accuracy of 87.5%

In general the neural network models show good performance across all different metrics. Recall and F1 Score might both be import metrics for predicting heart disease, since an incorrect prediction can be dangerous. Patients might not get identified with the disease resulting in potential heart disease issues.



(a) Confusion matrix neural network

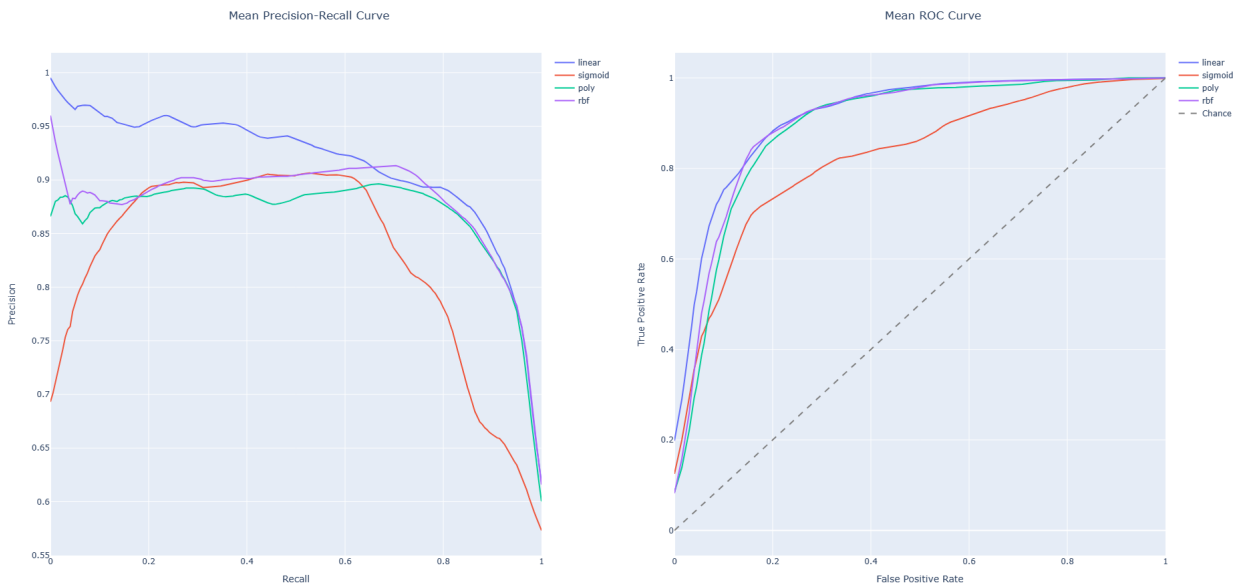
Metric	Mean accuracy
Roc auc	0.91
Precision	0.867
Recall	0.886
F1	0.875

(b) Performance neural network

Figure 4: Confusion matrix and a table with different metrics depicting performance.

4.4 Model 4: Support vector machine

The tables depicted in Figure 6 contain the mean accuracy for the performance of different kernel models. The linear and RBF kernels display strong performances, with Roc scores of 0.84, indicating their effective class separation abilities. The polynomial kernel follows closely, with a slightly lower Roc score of 0.83, while the sigmoid kernel stays behind with a Roc score of 0.76, suggesting challenges in class differentiation. Precision and recall metrics across the kernels mirror the Roc score trends, with the linear and RBF kernels leading in balanced performance, as reflected in their F1 scores of 0.87 and 0.86. The sigmoid kernel's lower score shows its limitations for this machine learning problem. A visual representation of the results can be seen in Figure 5:



(a) Mean precision recall curve

(b) Mean roc curve

Figure 5: Mean of Precision Recall and Roc curve depicting performance across six folds

5 Discussion

Acknowledging the ethical and social considerations in the development and application of machine learning models, to guarantee responsible use of the analytics in healthcare, this paper prioritizes transparency. Efforts were made to describe pre-processing methods, data sources and maintain methodology transparency.

6 Conclusion

The models performed generally well over all aspects. Of the models tested, the neural network seems to outperform the rest in terms of its predictive accuracy, as reflected by its ROC-AUC score. Nevertheless, the differences in performance are rather slim. The neural network likely benefits from its relatively more complex architecture in learning from the dataset, however the undeniable relevance of the features on the target labels allow for simpler models to achieve comparable performance. In practice, this makes the application of such technology more widely available, offering the potential for implementation of such predictive measures even to smaller hospitals which cannot afford the cost of maintaining a complex neural network.

Moreover, while the neural network outperformed other models with its accuracy score, there is a cost imbalance between false positives and negatives. While false positives might cause panic, false negatives could cost a life. Therefore, the recall is a very significant measure in the performance of these models. In this regard, the KNN classifier performed best, with the regression model as a close second. The similar recall values between the models, in addition to their cross-validation and testing procedures, suggest that it is unlikely that such a high value is due to overfitting. The support vector machine seemed to suffer the most in this classification task.

Nevertheless, in a production setting, the ideal solution would involve an ensemble. The result would combine each model's strongest points, such as the aforementioned accuracy of the neural network and recall of the KNN model, leading to higher overall performance. Although each of these models individually displayed satisfactory performance, and an ensemble would likely be exemplary, the field of medicine is not an insignificant one. Therefore, the application of these models would serve exclusively as a preliminary indication of the patient's health to doctors, rather than a genuine test for heart disease.

References

[1] Federico Soriano Palacios. Heart Failure Prediction Dataset. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction/data, 2021. Accessed: 2024-03-28.

[2] F. Loskot and P. Novotny. Die asymptomatische myokardischämie [asymptomatic myocardial ischemia]. *Zeitschrift fur die gesamte innere Medizin und ihre Grenzgebiete*, 45(13):370–373, 1990.

[3] F. D. Fuchs and P. K. Whelton. High blood pressure and cardiovascular disease. *Hypertension*, 75 (2):285–292, 2020. doi: 10.1161/HYPERTENSIONAHA.119.14240. URL https://doi.org/10.1161/HYPERTENSIONAHA.119.14240. Dallas, Tex. : 1979.

[4] S. A. Peters, Y. Singhateh, D. Mackay, R. R. Huxley, and M. Woodward. Total cholesterol as a risk factor for coronary heart disease and stroke in women compared with men: A systematic review and meta-analysis. *Atherosclerosis*, 2016. doi: 10.1016/j.atherosclerosis.2016.03.016. URL https://doi.org/10.1016/j.atherosclerosis.2016.03.016. Accessed: 2024-03-28.

[5] J. C. Stoltzfus. Logistic regression: a brief primer. *Academic Emergency Medicine*, 18(10):1099–1104, 2011. doi: 10.1111/j.1553-2712.2011.01185.x. Official journal of the Society for Academic Emergency Medicine.

[6] H Raeisi Shahraki, S Pourahmad, and N Zare. K important neighbors: A novel approach to binary classification in high dimensional data. *Biomed Res Int*, 2017:7560807, 2017. doi: 10.1155/2017/7560807. Epub 2017 Dec 11. PMID: 29376076; PMCID: PMC5742505.

[7] J Kufel, K Bargieł-Łączek, S Kocot, M Koźlik, W Bartnikowska, M Janik, Ł Czogalik, P Dudek, M Magiera, A Lis, I Paszkiewicz, Z Nawrat, M Cebula, and K Gruszczyńska. What is machine learning, artificial neural networks and deep learning? - examples of practical applications in medicine. *Diagnostics*, 13(15):2582, 2023. doi: 10.3390/diagnostics13152582. PMID: 37568945; PMCID: PMC10417718.

[8] Wei Yu, Tiebin Liu, Rodolfo Valdez, Marta Gwinn, and Muin J. Khoury. Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes. *BMC Medical Informatics and Decision Making*, 10(1):16, 2010. ISSN 1472-6947. doi: 10.1186/1472-6947-10-16. URL https://doi.org/10.1186/1472-6947-10-16.

[9] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. volume 2888, pages 986–996, 01 2003. ISBN 978-3-540-20498-5. doi: 10.1007/978-3-540-39964-3_62.

[10] Alan O. Sykes. An introduction to regression analysis. Technical report, University of Chicago, 1993.

[11] David R. Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20:215–242, 03 1958. ISSN 1467-9868.

[12] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. doi: 10.1007/BF00994018. Received May 15, 1993; Accepted February 20, 1995; Issue Date September 1995.

[13] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A practical guide to support vector classification. Online, 2016. Department of Computer Science, National Taiwan University, Taipei 106, Taiwan. Available online at http://www.csie.ntu.edu.tw/ cjlin.

7 Information sheet

Please include this page in your report either at the start or at the end, before the appendix. Do not change the formatting.

Group number
50

Authors

name	student number
Shivam Badal	2830263
Rose Al-Qarwani	2692218
Amla Malltezi	2757647
Bedirhan Salbacak	2778875
Kareem Mardini	2783070

Software used *Describe briefly which software and libraries you relied on. If you built a particular algorithm from scratch, you can explain that here.*

For this paper we have used Python with packages: numpy, pandas, plotly.express, scikit-learn and keras. The figures have been plotted with Plotly.express, while we used the other libraries to program the various machine learning algorithms described in the paper.

Use of AI tools *If you used tools like ChatGPT or Github Copilot in any phase of the report, please detail here which tools you used, what parts of the work you used them for and how. Note that if you use an AI tool and don't report it, it's considered fraud, so be detailed. Note also that having an AI tool write for you is not allowed.*

ChatGPT was used assisting in resolving bugs that were encountered writing code.

Link to code (optional) *If you would like to share your code, you can link to your repository here.*

Group disagreements *If there are any disagreements in the group, you may **not** remove a student from the author list without their consent. You should bring disagreements to our attention early. As a last resort, you can describe any grievances here*

No disagreements!

8 Appendix

Table 1: Descriptive Statistics of the Dataset

Feature	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
Age	53.51	9.43	28.00	47.00	54.00	60.00	77.00
RestingBP	132.54	17.99	80.00	120.00	130.00	140.00	200.00
Cholesterol	240.58	53.98	85.00	214.00	223.00	267.00	603.00
MaxHR	136.81	25.46	60.00	120.00	138.00	156.00	202.00
Oldpeak	0.92	1.07	0.00	0.00	0.60	1.50	6.20
HeartDisease	0.55	0.50	0.00	0.00	1.00	1.00	1.00
Binary Feature	**Proportion of '1s' (Mean Value)**						
ChestPainType_ASY	0.54	-	-	-	-	-	-
ChestPainType_ATA	0.19	-	-	-	-	-	-
ChestPainType_NAP	0.22	-	-	-	-	-	-
ChestPainType_TA	0.05	-	-	-	-	-	-

Table 2: Validation accuracy for 1, 10, 64, 128 and 256 neurons.

Neurons	1	10	64	128	256
Mean training accuracy	0.6604	0.675	0.804	0.833	0.839

Table 3: Validation accuracy for Relu, Sigmoid and Tanh.

Activation function	Relu	Sigmoid	Tanh
Mean training accuracy	0.839	0.789	0.842

Table 4: Performance k-nearest neighbours model

Metric	Mean accuracy
Roc auc	0.90
Precision	0.83
Recall	0.89
F1	0.86

Table 5: Performance logistic regression model

Metric	Mean accuracy
Roc auc	0.90
Precision	0.83
Recall	0.89
F1	0.86

Figure 6: Performance of all SVM kernels

(a) Performance with linear kernel

Metric	Mean Accuracy
Roc AUC	0.84
Precision	0.85
Recall	0.89
F1	0.87

(b) Performance with polynomial kernel

Metric	Mean Accuracy
Roc AUC	0.83
Precision	0.83
Recall	0.89
F1	0.86

(c) Performance with rbf kernel

Metric	Mean Accuracy
Roc AUC	0.84
Precision	0.84
Recall	0.88
F1	0.86

(d) Performance with sigmoid kernel

Metric	Mean Accuracy
Roc AUC	0.76
Precision	0.80
Recall	0.74
F1	0.77