A Comparison of Multilayer Perceptron and Support Vector Machine Applied to the Heart Disease Dataset

Chikaze Mori

1. Brief description and motivation of the problem

Cardiovascular disease is the leading cause of death in the United States and is responsible for 17% of national health expenditures [1]. Some computational techniques were proposed in the field of medical imaging to detect, prognosticate and diagnose the disease [2] [3]. In this work, we will apply Multilayer Perceptron (MLP) and Support Vector Machine (SVM) to the UCI Cleveland heart disease dataset [4] and compare the performances in the binary classification problem where we predict if the patients have a heart disease.

2. Description of the dataset including data types

The dataset used in this work is the Cleveland heart disease dataset from UCI [4], which contains 303 samples with 13 variables including 5 continuous and 8 categorical variables. Table 1 shows that the dataset does not contain any missing values and the number of samples in the classes are only slightly imbalance, which concludes that we do not need to apply techniques such as imputation, omission or SMOTE, to handle missing data or to balance the number of samples [5].

Table 1- Summary of Basic Statistics of the Dataset

	Heart disease					No disease				
Variables	**count**	**mean**	**std**	**min**	**max**	**count**	**mean**	**std**	**min**	**max**
age	138.0	56.6	8.0	35.0	77.0	165.0	52.5	9.6	29.0	76.0
sex	138.0	0.8	0.4	0.0	1.0	165.0	0.6	0.5	0.0	1.0
cp	138.0	0.5	0.9	0.0	3.0	165.0	1.4	1.0	0.0	3.0
trestbps	138.0	134.4	18.7	100.0	200.0	165.0	129.3	16.2	94.0	180.0
chol	138.0	251.1	49.5	131.0	409.0	165.0	242.2	53.6	126.0	564.0
fbs	138.0	0.2	0.4	0.0	1.0	165.0	0.1	0.3	0.0	1.0
restecg	138.0	0.4	0.5	0.0	2.0	165.0	0.6	0.5	0.0	2.0
thalach	138.0	139.1	22.6	71.0	195.0	165.0	158.5	19.2	96.0	202.0
exang	138.0	0.6	0.5	0.0	1.0	165.0	0.1	0.3	0.0	1.0
oldpeak	138.0	1.6	1.3	0.0	6.2	165.0	0.6	0.8	0.0	4.2
slope	138.0	1.2	0.6	0.0	2.0	165.0	1.6	0.6	0.0	2.0
ca	138.0	1.2	1.0	0.0	4.0	165.0	0.4	0.8	0.0	4.0
thal	138.0	2.5	0.7	0.0	3.0	165.0	2.1	0.5	0.0	3.0

Based on the boxplot of the non-categorical variables, figure 1, the number of outliers is relatively small, and it is not necessary to remove them. As multicollinearity effects machine learning algorithms such as SVM [6], we visualised correlation matrix to investigate if there is high correlation between any variables. In the correlation matrix, figure 2, it is shown that there is no such high correlation between each variable as over 0.7 or under -0.7, which indicates that it is not necessary to conduct principal component analysis.

For data preparation, as the range of values in each variable is different, we normalise the values in the range of 0 and 1 to increase consistency. In addition to this, 5 variables among the categorical variables are to be dummy encoded.

The target variable is a binary variable that represents if the patient has a heart disease or not, class 0 or 1, accordingly.

Figure 1- Boxplot of non-categorical variables

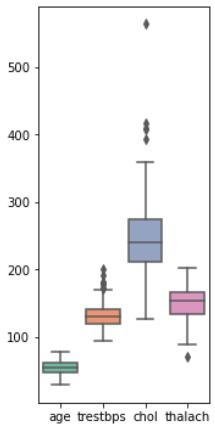
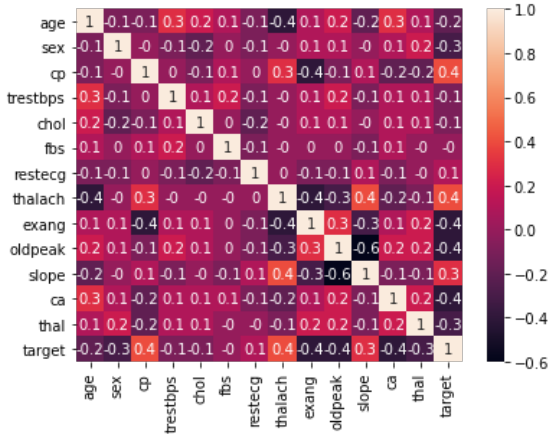


Figure 2 - Correlation matrix



3. Brief summary of the two neural network models with their pros and cons

3.1. Multilayer Perceptron (MLP)

MLP is a class of artificial neural network that consists of three or more fully connected layers of nodes, an input layer, one or more hidden layers and an output layer. Hidden layers are to apply weights to the inputs and direct them through an activation function as the output. Based on the amount of error in the output compared to the expected result, the weights in the perceptron are to be modified thorough the process of backpropagation. MLP is a preferred technique for gesture and pattern recognition [7] [8], which is proven to be a universal approximator [9] that does not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration in comparison to other probability-based models [7]. However, MLP is a black box, and we cannot know influences of each independent variable to the dependent variables.

3.2. Support Vector Machine

SVM is a supervised learning model and one of the classical machine learning techniques that can still help solve big data classification problems [10]. Conceptionally, the implement is that input vectors are non-linearly mapped to a very high-dimension feature space where a linear decision surface is constructed [11]. It assigns samples to one of two categories and maximise the width of the gap between the two categories. It can help the multidomain applications in a big data environment, however, SVM is mathematically complex [9] and it is computationally expensive to reach the SVM solution by stochastic gradient descent, mainly due to the regularization term [12].

4. Hypothesis statement

We expect both algorithms to perform well since both are considered as methods that are well suited for classification, regression and prediction tasks [13]. However, the previous studies show that SVM is unbeatable in classification while MLP shows better ability of generalisation in regression [13]. As we conduct a binary classification task, we expect SVM to perform better than MLP. In terms of learning speed, we expect SVM to be quicker in training than MLP, as SVM is based on the local approximation strategy while MLP is based on the global approximation strategy [13].

According to the previous empirical study [14], both SVM and MLP can improve its performances by optimising hyperparameters, except in the cases where default hyperparameters already yield a perfect result. Based on their study, we expect SVM to take less time to beat default hyperparameter performance by hyperparameters tuning than MLP.

5. Description of choice of training and evaluation methodology

The dataset is to be split into a train set (80%) and a test set (20%). We train our models on the train set that contains 242 samples. For each model, we tune its hyperparameters to find optimum values using grid search with cross validation. We use 10-fold cross validation to prevent overfitting and improve the accuracy of error estimation [15]. The error measure used is classification error.

To inspect the results more closely, we will calculate precision, recall and F-score, and visualise confusion matrix and ROC curve. Along with ROC, we will also calculate AUC. Our two models are to be compared thorough these measurements.

6. Choice of parameters and experimental results

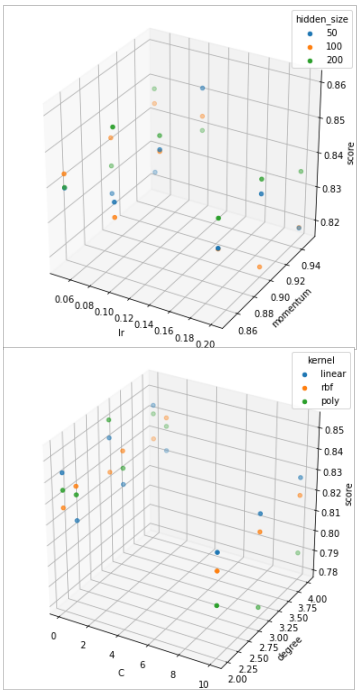
The MLP model that we built contains an input layer, a hidden layer and an output layer. The activate function ReLU was applied to the hidden layer and softmax function was applied to the output layer. The loss function used was negative log likelihood loss. The model was trained through back propagation. Dropout regularisation is considered as a technique used to avoid overfitting, however, it is not applied to our model as our dataset is small.
In grid search, the hyperparameters tuned were learning rate, momentum of the optimiser and size of the hidden layer for MLP, and regularisation parameter C, degree of polynomial kernel function and kernel for SVM.

Table 2- Results of grid search

MLP			
lr	momentum	hidden_size	score
0.05	0.85	50	0.839333
0.05	0.9	50	0.827
0.05	0.95	50	0.822833
0.05	0.85	100	0.843333
0.05	0.9	100	0.843333
0.05	0.95	100	0.8435
0.05	0.85	200	0.8395
0.05	0.9	200	0.835167
0.05	0.95	200	0.847833
0.1	0.85	50	0.8395
0.1	0.9	50	0.843833
0.1	0.95	50	0.851667
0.1	0.85	100	0.835167
0.1	0.9	100	0.843333
0.1	0.95	100	0.8435
0.1	0.85	200	0.860167
0.1	0.9	200	0.847833
0.1	0.95	200	0.839333
0.2	0.85	50	0.835167
0.2	0.9	50	0.839333
0.2	0.95	50	0.818333
0.2	0.85	100	0.835
0.2	0.9	100	0.818167
0.2	0.95	100	0.8185
0.2	0.85	200	0.8435
0.2	0.9	200	0.8435
0.2	0.95	200	0.835167

SVM			
C	degree	kernel	score
0.1	2	linear	0.843667
0.1	2	rbf	0.827167
0.1	2	poly	0.835500
0.1	3	linear	0.843667
0.1	3	rbf	0.827167
0.1	3	poly	0.852167
0.1	4	linear	0.843667
0.1	4	rbf	0.827167
0.1	4	poly	0.839500
1	2	linear	0.823000
1	2	rbf	0.839167
1	2	poly	0.835167
1	3	linear	0.823000
1	3	rbf	0.839167
1	3	poly	0.830833
1	4	linear	0.823000
1	4	rbf	0.839167
1	4	poly	0.835000
10	2	linear	0.827167
10	2	rbf	0.818500
10	2	poly	0.802167
10	3	linear	0.827167
10	3	rbf	0.818500
10	3	poly	0.781500
10	4	linear	0.827167
10	4	rbf	0.818500
10	4	poly	0.789833

Figure 3 - Results of grid search (MLP on top, SVM on bottom)



As the results of grid search, we obtained the best hyperparameters shown highlighted in table 2. The hyperparameters that achieved the best classification accuracy were 0.1 of learning rate, 0.85 of momentum and 200 of hidden size for MLP, and 0.1 of C, 3 of degree and kernel type of poly for SVM, whose accuracy were 0.860 and 0.852 respectively. The

results, figure 3 along with table 2, show that bigger hidden size tended to lead higher accuracy in MLP, while combination of hyperparameters is more important to achieve high accuracy rather than the hyperparameters themselves in SVM.

7. Analysis and critical evaluation of results

Figure 4, the confusion matrices of each model on the test set, shows that both models performed quite well, which is consistent with the previous study [13]. However, MLP tended to predict class 0 more often than SVM, while SVM predicted class 1 more often. Table 3 shows the measurements of the models on the test set for each class, 0 and 1, accordingly. On table 3, the biggest difference is that MLP has much higher recall 0.85 for class 0, while SVM has higher recall 0.91 for class 1. This means that SVM performed better at successfully classify class 1 when the true class was 1, which is crucial when detecting a heart disease.

Figure 4-Confusion matrices

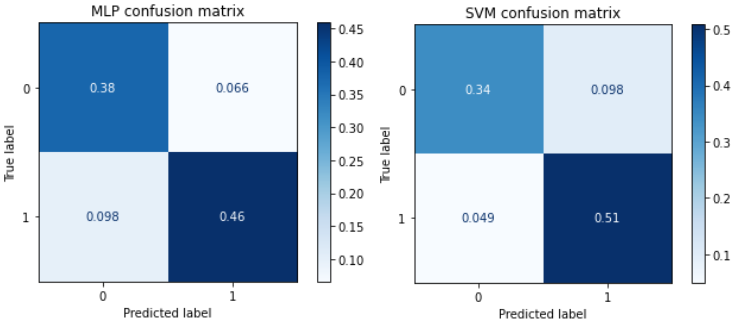


Table 3 - Measurements of the models

	MLP	SVM
Precision	0.79, 0.88	0.88, 0.84
Recall	0.85, 0.82	0.78, 0.91
F1	0.82, 0.85	0.82, 0.87

Table 4 displays the classification accuracies of the models on the train set, cross validation and the test set, along with the computing time that consumed during hyperparameter tuning by grid search with cross validation and training. Overall, SVM achieved higher accuracy than MLP, which is consistent with our hypothesis. MLP required much more time to be tuned than SVM, especially with bigger number of epochs and hidden size, which indicates that MLP is capable of being trained more computationally, while SVM is simpler.

Table 4 -Classification accuracy of the models on each phase and computing time

	MLP	SVM
Train	0.868	0.884
Validation	0.860	0.852
Test	0.836	0.852
Time (sec)	21.920	1.401

Both models showed moderate bias and variance which are indicated by the small difference of accuracies between the train and test sets. However, the worse performance of MLP and the slightly bigger drop of accuracy on the test set suggest our MLP model's inability of reducing bias and that it overfitted more than SVM, considering the fact that we did not apply dropout regularisation to the model even though our dataset is small.

Figure 5 - ROC curves and AUC on class 1 (with heart disease)

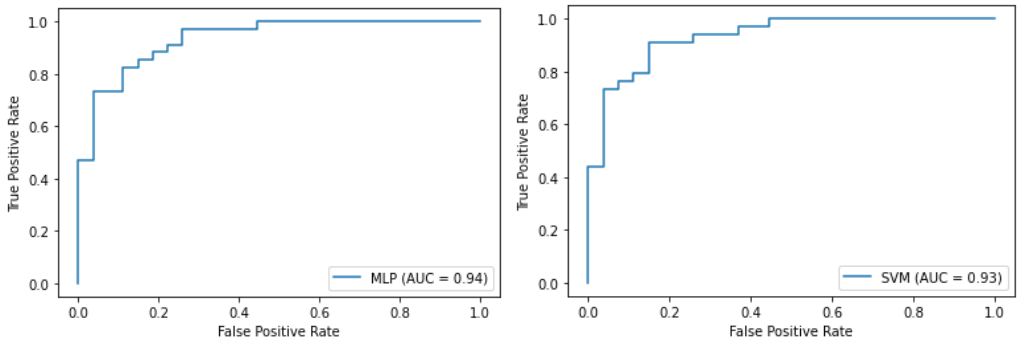


Figure 5 shows ROC curves and AUC of each model. We do not see big differences in the ROC curves and AUC, despite of the difference in their classification accuracies. However, in our analysis of detecting a heart disease, it is more important to classify class 1, a presence of a heart disease, rather than class 0 as cardiovascular disease is a life-threatening illness [1]. As we saw on table 3, SVM had higher recall 0.91 on class 1 than MLP's recall 0.82, which indicates that SVM detects a heart disease more likely than MLP, even though the classification might not be correct. In addition to this, it is noticeable that SVM achieved higher F-score 0.87 on class 1 than MLP's F-score 0.85, while the two models are commensurable and achieved the same F-score 0.82 on class 0. Thus, along with its higher accuracy, we conclude that the preferred model for our disease detection task is SVM.

8. Conclusions, lessons learned, references and future work

8.1. Conclusions, lessons learned and future work

Our study showed how accurately the two trained models were able to predict whether there is a presence of a heart disease. We conclude that MLP and SVM are comparable, which is consistent with the previous study [13] and are able to achieve higher accuracy by tuning its hyperparameters. However, SVM performs better than MLP in a classification task and is almost unbeatable [16].

We also learned that performance of MLP heavily relies on the size of its hidden layers. With more neurons in hidden layers, MLP can achieve higher accuracy, however, bigger hidden layers require longer computing time for hyperparameter tuning and training. SVM is a model that could be built much faster than MLP, which indicates that SVM does not require much of hyperparameter tuning, while MLP does. Thus, it is suggested that MLP is more capable of computational training than SVM and has potential to outperform SVM when it is trained with bigger size of hidden layers.

For future work, we recommend training the models thorough different methods, such as boosting, which is a well-known ensemble learning algorithm that is very effective in improving generalisation performance [17]. Also, applying feature extraction methods such as PCA and Wavelets to the dataset could potentially improve performance for both models [18]. Also, for MLP models, it is interesting to see if dropput regularisation function can be applied to avoid overfitting and improve accuracy even when the dataset is small.

8.2. References

[1] Heidenreich Paul A. *et al.*, 'Forecasting the Future of Cardiovascular Disease in the United States', *Circulation*, vol. 123, no. 8, pp. 933–944, Mar. 2011, doi: [1].

[2] K. Uyar and A. İlhan, 'Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks', *Procedia Computer Science*, vol. 120, pp. 588–593, Jan. 2017, doi: 10.1016/j.procs.2017.11.283.
[3] A. Gupta, R. Kumar, H. S. Arora, and B. Raman, 'MIFH: A Machine Intelligence Framework for Heart Disease Diagnosis', *IEEE Access*, vol. 8, pp. 14659–14674, 2020, doi: 10.1109/ACCESS.2019.2962755.
[4] 'UCI Machine Learning Repository: Heart Disease Data Set'. https://archive.ics.uci.edu/ml/datasets/Heart+Disease (accessed Mar. 17, 2021).
[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, 'SMOTE: Synthetic Minority Over-sampling Technique', *jair*, vol. 16, pp. 321–357, Jun. 2002, doi: [5].
[6]C. F. Dormann *et al.*, 'Collinearity: a review of methods to deal with it and a simulation study evaluating their performance', *Ecography*, vol. 36, no. 1, pp. 27–46, 2013, doi: [6].
[7] Mu-Chun Su, Woung-Fei Jean, and Hsiao-Te Chang, 'A static hand gesture recognition system using a composite neural network', in *Proceedings of IEEE 5th International Fuzzy Systems*, Sep. 1996, vol. 2, pp. 786–792 vol.2, doi: 10.1109/FUZZY.1996.552280.
[8] C. M. Bishop and P. of N. C. C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.
[9] K. Hornik, M. Stinchcombe, and H. White, 'Multilayer feedforward networks are universal approximators', *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989, doi: [9].
[10] C. F. Dormann *et al.*, 'Collinearity: a review of methods to deal with it and a simulation study evaluating their performance', *Ecography*, vol. 36, no. 1, pp. 27–46, 2013, doi: [10].
[11] S. Suthaharan, 'Support Vector Machine', in *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, S. Suthaharan, Ed. Boston, MA: Springer US, 2016, pp. 207–235.
[12] C. Cortes and V. Vapnik, 'Support-vector networks', *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: [12].
[13] R. Collobert and S. Bengio, 'Links between perceptrons, MLPs and SVMs', in *Twenty-first international conference on Machine learning - ICML '04*, Banff, Alberta, Canada, 2004, p. 23, doi: 10.1145/1015330.1015415.
[14] S. Sanders and C. Giraud-Carrier, 'Informing the Use of Hyperparameter Optimization Through Metalearning', in *2017 IEEE International Conference on Data Mining (ICDM)*, Nov. 2017, pp. 1051–1056, doi: 10.1109/ICDM.2017.137.
[15] G. Jiang and W. Wang, 'Error estimation based on variance analysis of k-fold cross-validation', *Pattern Recognition*, vol. 69, pp. 94–106, Sep. 2017, doi: [15].
[16] S. Osowski, K. Siwek, and T. Markiewicz, *MLP and SVM networks - a comparative study*, vol. 46. 2004, p. 40.
[17] N. C. Oza and S. Russell, 'Experimental comparisons of online and batch versions of bagging and boosting', in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*, San Francisco, California, 2001, pp. 359–364, doi: 10.1145/502512.502565.
[18] E. Gumus, N. Kilic, A. Sertbas, and O. N. Ucan, 'Evaluation of face recognition techniques using PCA, wavelets and SVM', *Expert Systems with Applications*, vol. 37, no. 9, pp. 6404–6408, Sep. 2010, doi: 10.1016/j.eswa.2010.02.079.

9. Glossary

Accuracy	Percentage of correct predictions made by the model.
Attribute	Synonym for feature. In fairness, attributes often refer to characteristics pertaining to individuals.
Bias	A learner's tendency to consistently learn the same wrong thing.
Boosting	A machine learning technique that iteratively combines a set of simple and not very accurate classifiers into a classifier with high accuracy by up weighting the examples that the model is currently misclassifying.
Categorical Variables	Variables with a discrete set of possible values. Can be ordinal (order matters) or nominal (order doesn't matter).
Class	One of a set of enumerated target values for a label.
Classification	Predicting a categorical output.
Confusion Matrix	Table that describes the performance of a classification model by grouping predictions into 4 categories.
Cross-validation	A mechanism for estimating how well a model will generalize to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.
Dummy Variable	A placeholder for a variable that will be integrated over, summed over, or marginalized.
Feature	With respect to a dataset, a feature represents an attribute and value combination.
Hyperparameters	Higher-level properties of a model such as how fast it can learn (learning rate) or complexity of a model.
Label	In supervised learning, the "answer" or "result" portion of an example.
Layer	A set of neurons in a neural network that process a set of input features, or the output of those neurons.
Model	A data structure that stores a representation of a dataset (weights and biases). Models are created/learned when you train an algorithm on a dataset.
Multi-class classification	Predicts one of multiple possible outcomes.
Neural Networks	A machine learning method that's *very* loosely based on neural connections in the brain.
Neuron	A node in a neural network, typically taking in multiple input values and generating one output value.
Normalisation	Restriction of the values of weights in regression to avoid overfitting and improving computation speed.
Overfitting	Occurs when your model learns the training data too well and incorporates details and noise specific to your dataset.
Oversampling	A technique used to solve imbalance of data. It selects more samples from one class than another.

Parameters	Properties of training data learned by training a machine learning model or classifier. They are adjusted using optimisation algorithms and unique to each experiment.
Principal Component Analysis (PCA)	Simply looks at the direction with the most variance and then determines that as the first principal component.
ReLU function	A piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.
Softmax function	A function that turns a vector of K real values into a vector of K real values that sum to 1.
Test Set	A set of observations used at the end of model training and validation to assess the predictive power of your model.
Training Set	A set of observations used to generate machine learning models.
Variance	How much a list of numbers varies from the mean (average) value.
References	https://developers.google.com/machine-learning/glossary https://ml-cheatsheet.readthedocs.io/en/latest/glossary.html http://www.datascienceglossary.org

10. Implementation details

For convenience, all the data pre-processing process was managed in Python separately from our analysis. The functions used for training the models are listed below:

- torch.nn
- skorch.NeuralNetClassifier
- sklearn.svm.SVC
- skorch.GridSearchCV

For MLP models, the dataset was converted from pandas DataFrame to tensor (torch.tensor) which is the only acceptable format of data in PyTorch. The optimiser used for MLP models was stochastic gradient descent (torch.optim.SGD). To allow us to use softmax function for outputs, the output layer in our MLP models contains two neurons. As saving function that Skorch provides does not store model structure itself, we only saved the learned attributes and parameters of the trained model and read the same model structure again on our testing phase. This procedure is only for convenience of coding and does not interfere our models or results.

In SVM models, the value of gamma, kernel coefficient for rbf and poly kernels, was set to the default parameter scale, which uses 1 / (number of features * variance of the dataset along the specified axis).

Websites that were referred to for our coding are listed below:

https://pytorch.org
https://skorch.readthedocs.io/en/stable/index.html
https://scikit-learn.org/stable/index.html