

Supervised Learning: Comparison of Five Learning Algorithms

Herbert Ssegane GtID#903755945

1.0 Introduction

This analysis explores differences in the complexity and performance of several supervised learning algorithms on two real world datasets. Supervised learning is a branch of machine learning where a given algorithm (learner) learns to map inputs to an output (a target or label) by providing it with examples of both the inputs and the output. The two common tasks of supervised learning are regression where the output is a continuous and classification where the output is a label or discrete. The objective of this analysis is to compare performance of five learning algorithms on two classification datasets. The five algorithms are decision trees (DT), neural networks (NN), boosting (GBT), support vector machines (SVM), and k-nearest neighbors (KNN).

2.0 Data

2.1 Wisconsin breast cancer diagnostic data (WBCD)

The dataset consists of 569 data points with 30 features computed from each cancer cell nucleus to predict whether the cancer is benign (B) or malignant (M). This is a two-class classification problem, where the column *diagnosis* is the target (Table 1). For a detailed description of each feature, the reader is referred to *Kaggle (2022)*.

Table 1: Sample of the Wisconsin breast cancer data

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
B	12.70	12.17	80.88	495.00	0.09	0.06	0.02
B	13.28	13.72	85.79	541.80	0.08	0.09	0.05
M	14.27	22.55	93.77	629.80	0.10	0.12	0.15
B	13.70	17.64	87.76	571.10	0.10	0.08	0.05
B	9.29	13.90	59.96	257.80	0.14	0.12	0.03

The dataset is relevant because it is a real-world data of the second leading cause of cancer deaths in US women, at a fatal rate of 2.6% or 1 in 39 women [*Cancer, 2022*]. The data provides measurements (features) extracted from images of cancer cells that are predictive of whether the cells are benign or malignant. Thus, it is suited for supervised learning with potential application of early breast cancer detection

2.2 Heart failure prediction data (HFP)

The dataset consists of 918 data points with 11 features to predict heart disease (*HeartDisease* column in Table 2). A value of one (1) for heart disease and a value of zero (0) for no heart disease. This is also a two-class classification problem, where the *HeartDisease* feature is the target. For a detailed description of each feature and data sources, refer to *Fedesoriano (2022)*.

Table 2: Sample of heart failure data


Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0.00	Up	0
49	F	NAP	160	180	0	Normal	156	N	1.00	Flat	1
37	M	ATA	130	283	0	ST	98	N	0.00	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.50	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0.00	Up	0

This data is relevant because it demonstrates application of supervised learning for predictive analytics in the health care industry, knowing that heart failure is the leading cause of deaths, globally.

2.3 Data pre-processing

The supervised learning algorithms expect numeric values. Thus, data preprocessing was implanted on both datasets prior to exploring the learning algorithms. The target feature of the WBCD was the only categorical feature. It was transformed into a binary feature where zero (0) represented benign cancer and one (1) represented malignant cancer. The HFP data has four categorical features (Sex, ChestPainType, RestingECG, ExerciseAngina, and ST_Slope). These features had a maximum cardinality of 4 and thus were transformed into dummy variables to capture the different categorical levels without risk of the curse of dimensionality. Thus, the HFP feature space expanded from 11 to 20 features (refer to figure below).

Sex	ChestPainType						
M	ATA						
F	NAP						
M	ATA						
F	ASY						
M	NAP						



Sex_F	Sex_M	ChestPainType_ASY	ChestPainType_ATA	ChestPainType_NAP	ChestPainType_TA
0	1	0	1	0	0
1	0	0	0	1	0
0	1	0	1	0	0
1	0	1	0	0	0
0	1	0	0	1	0

Figure 1: One hot encoding (OHE) example for categorical features with low cardinality

3.0 Supervised learning

3.1 Compute resources

This analysis uses Azure Databricks compute resources of a standard cluster, 10.2ML runtime, and worker & driver nodes 56GB memory with 16 CPU cores. A standard cluster on Azure Databricks is configured to run workloads in Python, SQL, R, and Scala programming languages. For this analysis, Python was used. The choice of the worker & driver nodes, combined with parallel computing directly affect train and scoring time.

3.2 Train-test data splitting

Each dataset was split into 70% and 30% for train and test datasets, respectively. The splitting was stratified using the target to maintain proportionality within the two data splits. The train data is used to generate the training and validation curves during assessment of the impact of model complexity and impact of train size on model performance and training time. These two curves are generated using a 5-fold cross validation, where the mean of the 5-folds is reported as the training and validation scores for varying levels of model complexity and train sizes. This data was also used to generate training and scoring curves showing the time complexity as the sample size increases. The test data is used to assess each model's ability to generalize instead of just memorization.

3.3 Algorithm implementation framework

The *Scikit-learn* version 0.24.1 python library's implementation of the five algorithms is used. This analysis did not focus on exhaustive parameter tuning. However, to assess the impact of model complexity on performance metrics, one or two hyperparameters were varied. The optimal hyperparameter was chosen based on train and validation F1-score. The F1-score is a harmonic mean of the precision and recall scores. The precision score measures the confidence of the model (e.g., how often is the model correct when it predicts that the cancer is benign or malignant), while the recall score measures the robustness of the model (e.g., how many of the actual benign or actual malignant cancer cases, was the model able to accurately predict). Refer to equations 1-3 for the basic mathematical formulations of the three metrics. To avoid overfitting, the hyperparameter whose F1-scores for train and validation data were both high and close in value, was chosen as the optimal hyperparameter for each of the two datasets. Table 3 shows the hyperparameter used to capture model complexity for each algorithm. Other hyperparameters were set to default

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (3)$$

Where TP, FP, and FN are true positive, false positive, and false negative, respectively (https://en.wikipedia.org/wiki/Precision_and_recall)

Table 3: Algorithms, hyperparameter, and search space for model complexity

Algorithm	Scikit-learn module	Hyperparameter (s)	Comment	Search space
Decision Trees (DT)	DecisionTree Classifier	<i>max_depth</i>	Controls the depth of the tree from the root node to the leaf node. Gini used to assess quality of tree splitting	[1, 10]
Neural Networks (NN)	MLP Classifier	<i>hidden_layer_sizes</i>	Controls the number of hidden layers and the number of nodes in each layer. adam optimizer and relu activation are used	[40, 275]
Boosting (GBT)	GradientBoosting Classifier	<i>n_estimators</i>	controls the number of boosting stages to implement. max_depth set to 2 based on optimal from DT	[20, 180]
Support Vector Machines (SVM)	SVC	<i>kernel, C</i>	tested 4 kernels, linear kernel was chosen. Then focused on C regularization parameter	[0.0004, 0.1]
k-Nearest Neighbors (KNN)	Kneighbors Classifier	<i>distance, n_neighbors</i>	tested 6 distance metrics, euclidean distance was chosen. Then focused on varying number of nearest neighbors	[1, 20]

4.0 Results

4.1 Decision Trees (DT)

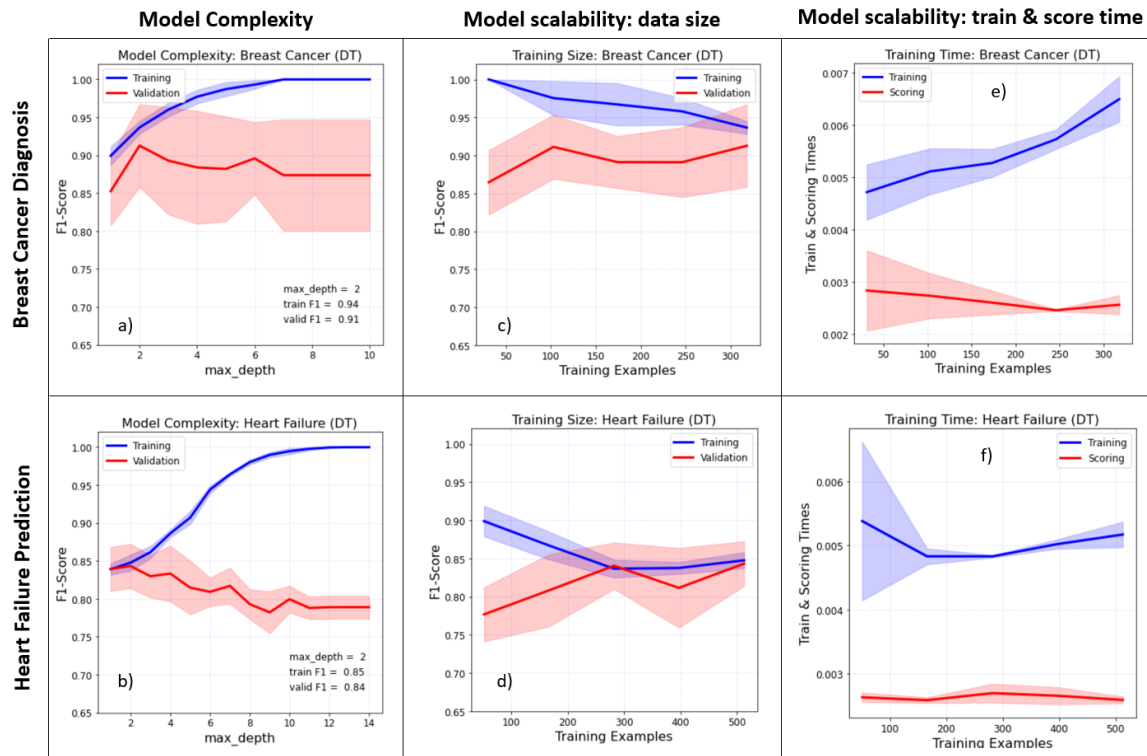


Figure 2: comparison of model complexity and data size on Decision Tree (DT) performance during training and validation (5-fold cross validation) on two data. Performance is measured by F1 score.

For both datasets (Figures 2: a & b), as the model complexity increased (*max_depth*), the DT overfit the data. This is demonstrated by the widening gap between training and validation curves as the *max_depth* increased. To minimize overfitting, a *max_depth* of two (2) was chosen for proceeding analyses. The corresponding F1-scores were high and relatively close (0.94 vs. 0.91 for WBCD and 0.85 vs. 0.84 for HFD). The impact of training data size on DT overfitting is depicted by Figures 2 c & d. Both figures show a narrowing of the two curves as the data size increases. Thus, one approach to minimize model overfitting is to increase data size if possible. Figures 3 e & f demonstrate computation time complexity during DT training and scoring as a function of data size. Both datasets depict a linear trend (more pronounced in Figure 1e) for DT training while a constant time during DT scoring.

The trained DT(s) on both datasets (Figure 3) have high predictive performance on the respective test data. Most of the performance metrics of AUC, precision, recall, F1, and accuracy are ≥ 0.8 . The corresponding confusion matrices are normalized to the predictions and thus represent class-level precision. For example, the precision for positive breast cancer diagnosis is 0.98 while for a positive heart failure prediction is 0.83. Although, the WBCD is less in data size and has less features than the HFD, the DT better captured the underlying causal relationship. The overall training and scoring times were similar for both data (≤ 0.03 s).

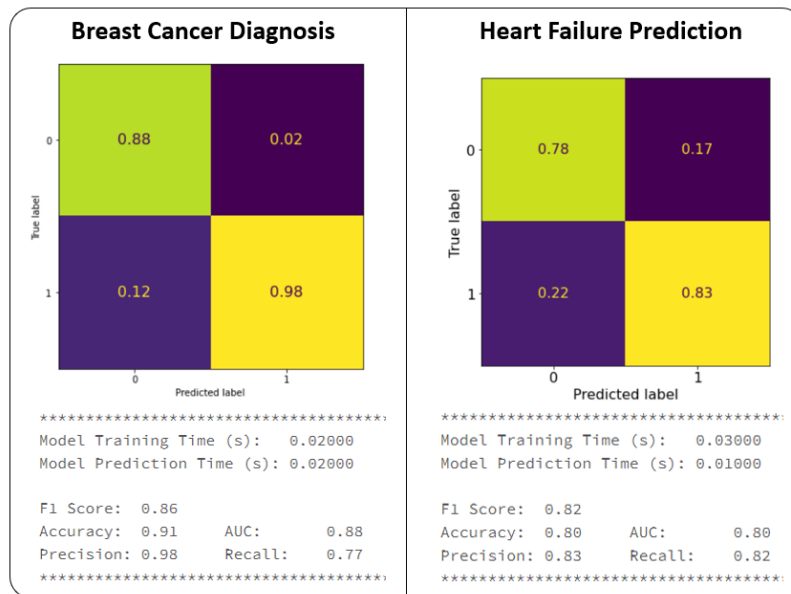


Figure 3: Confusion matrices depicting DT's generalization performance on the two test data not seen during the training and validation.

4.2 Neural Networks (NN)

For the NN the increase in model complexity ($n_hidden_layer_sizes$) does not result in drastic differences for the train and validation curves (Figures 4 a & b) compared to the DT (Figures 2 a & b) for the predefined hyper-parameter search space. To minimize overfitting, $n_hidden_layer_sizes$ of 250 and 200 were chosen for proceeding analyses. The corresponding F1-scores were high and relatively close (0.90 vs. 0.90 for WBCD and 0.88 vs. 0.86 for HFD). The behavior of the impact of training data size on NN overfitting is similar to that of DT (Figures 4 c & d versus Figures 2 c & d). The figures show a narrowing of the two curves as the data size increases. Figures 4 e & f better demonstrate the linear and constant computation time complexity during NN training and scoring as a function of data size.

The trained NN(s) on both datasets (Figure 5) have higher predictive performance on the respective test data than the trained DT. Most of the performance metrics of AUC, precision, recall, F1, and accuracy are ≥ 0.85 . The corresponding confusion matrices depict precision for positive breast cancer diagnosis of 0.96 while for a positive heart failure prediction is 0.86. Also, the NN better captured the underlying causal relationship for the WBCD than HFP data. There is a 10-fold difference in overall training times ($\leq 0.61s$) compared to the scoring times ($\leq 0.03s$) for both data.

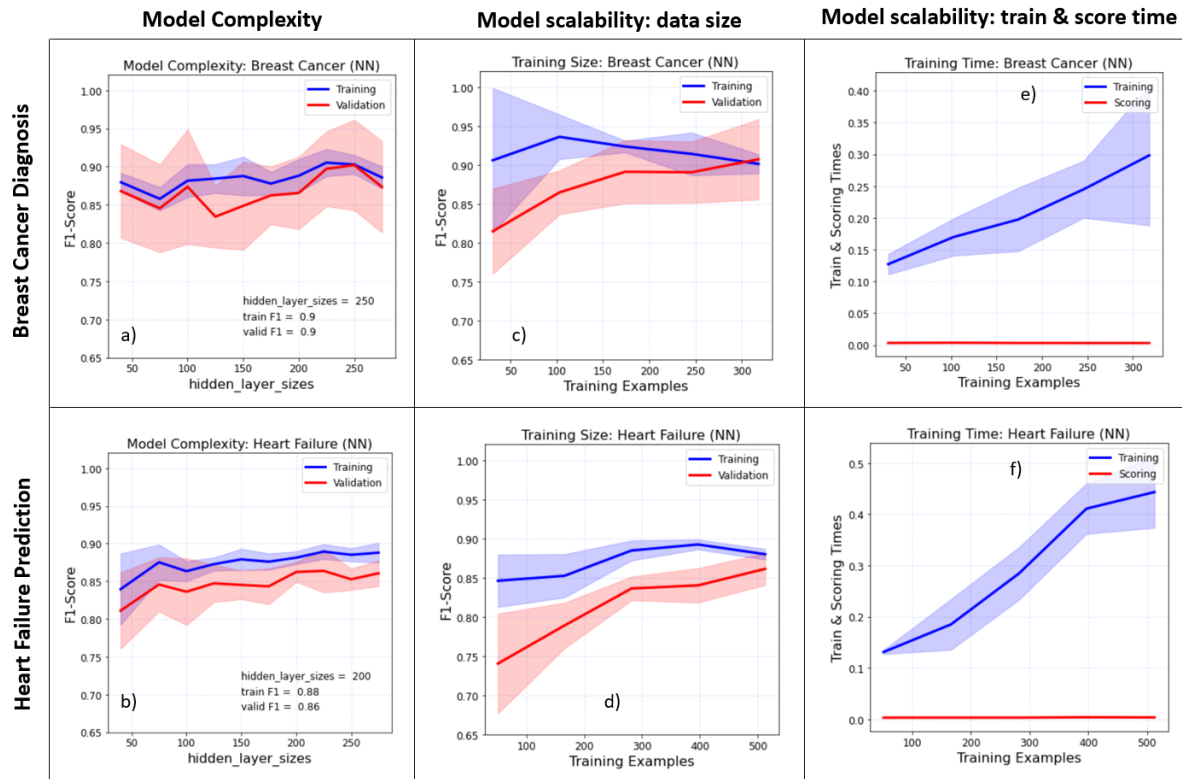


Figure 4: comparison of model complexity and data size on Multi-Layer Perceptron Neural Network Classifier (NN) performance during training and validation (5-fold cross validation) on two data. Performance is measured by F1 score.

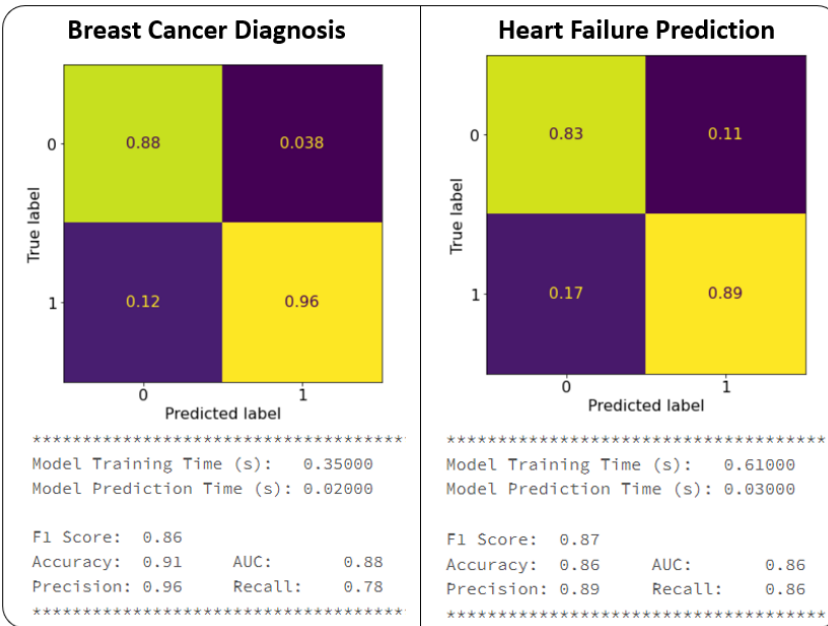


Figure 5: Confusion matrices depicting NN's generalization performance on the two test data not seen during the training and validation.

4.3 Boosting (GBT)

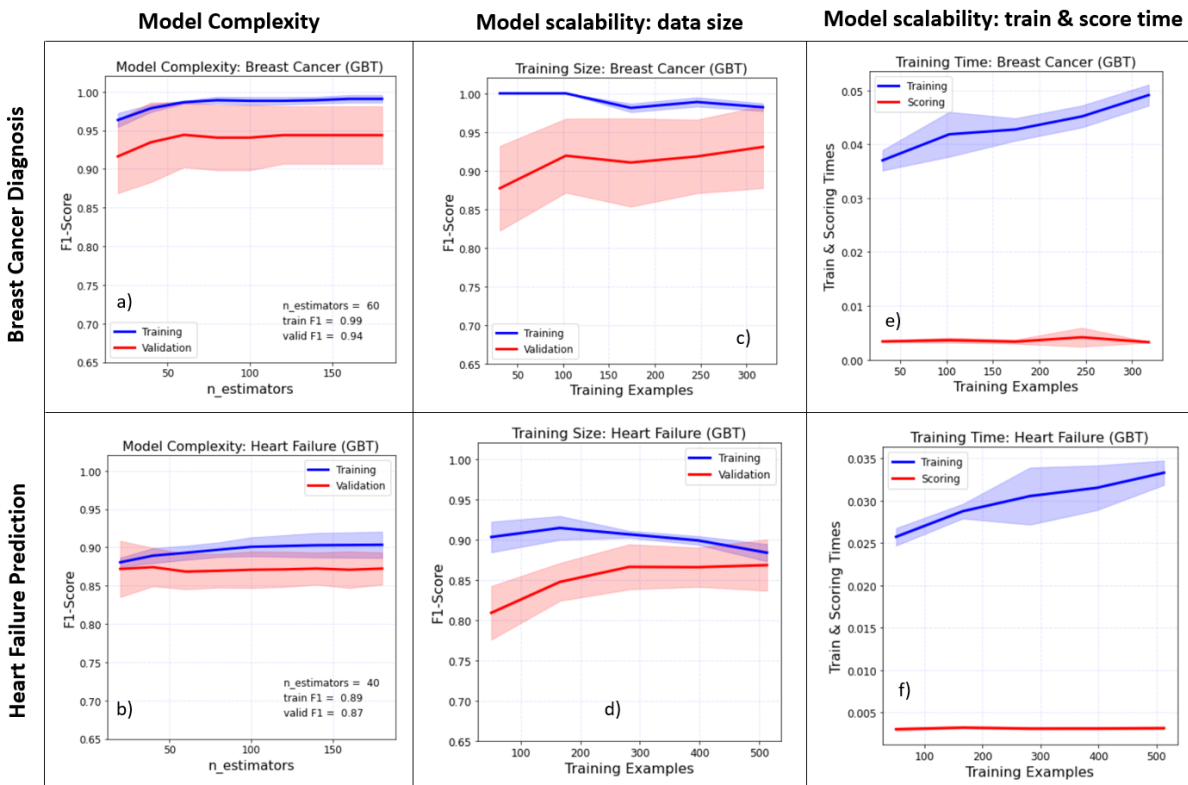


Figure 6: comparison of model complexity and data size on Gradient Boosting Classifier (GBT) performance during training and validation (5-fold cross validation) on two data. Performance is measured by F1 score.

Figures 6: a & b show the impact of model complexity ($n_estimators$) on model overfitting of the GBT. The GBT overfits for both data when the number of boosting stages ($n_estimators$) exceeds 60. To minimize overfitting, $n_estimators$ of 40 and 60 were chosen for proceeding analyses. The corresponding F1-scores were high and relatively close (0.99 vs. 0.94 for WBCD and 0.89 vs. 0.87 for HFD). The impact of training data size on GBT overfitting is depicted by Figures 6 c & d. the behavior is similar to DT and NN, demonstrated by the narrowing of the two curves as the data size increases. Figures 6 e & f demonstrate computation time complexity during GBT training and scoring as a function of data size. The behavior is similar to DT and NN.

The trained GBT(s) on both datasets (Figure 7) have the highest predictive performance compared to DT and NN. Most of the performance metrics of AUC, precision, recall, F1, and accuracy are ≥ 0.88 . The corresponding confusion matrices precision for positive breast cancer diagnosis is 1.0 while for a positive heart failure prediction is 0.88. The overall training and scoring times were similar for both data ($\leq 0.1s$).

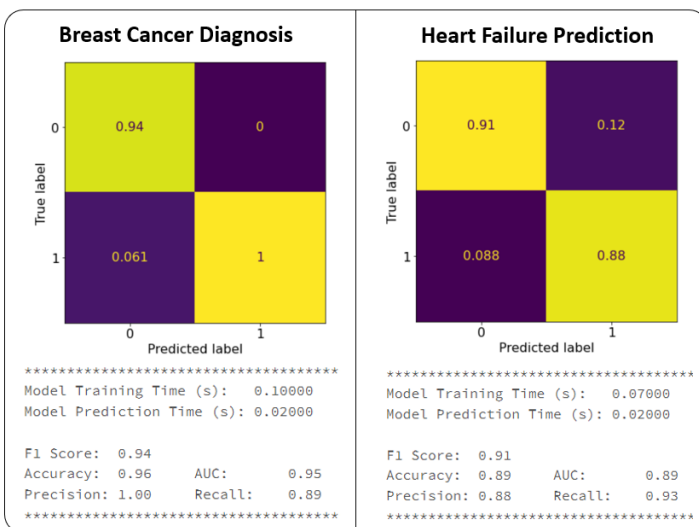


Figure 7: Confusion matrices depicting NN's generalization performance on the two test data not seen during the training and validation.

4.4 Support Vector Machines (SVC)

To minimize overfitting, the C regularization parameter was set to of 0.01 and 0.1 for the two data (Figures 8 a & b). The corresponding F1-scores were high and relatively close (0.94 vs. 0.93 for WBCD and 0.87 vs. 0.86 for HFD). The impact of training data size on SVC overfitting is similar to DT, NN, and GBT (Figures 8 c & d). Computation time complexity during training is exponential while constant during scoring as a function of data size. Thus, SVC are better suited for small data sizes. The trained SVC(s) on both datasets (Figure 9) have high predictive performance compared to DT and NN. All 0.98 while for a positive heart failure prediction is 0.88. The overall training and scoring times are comparable to NN times.

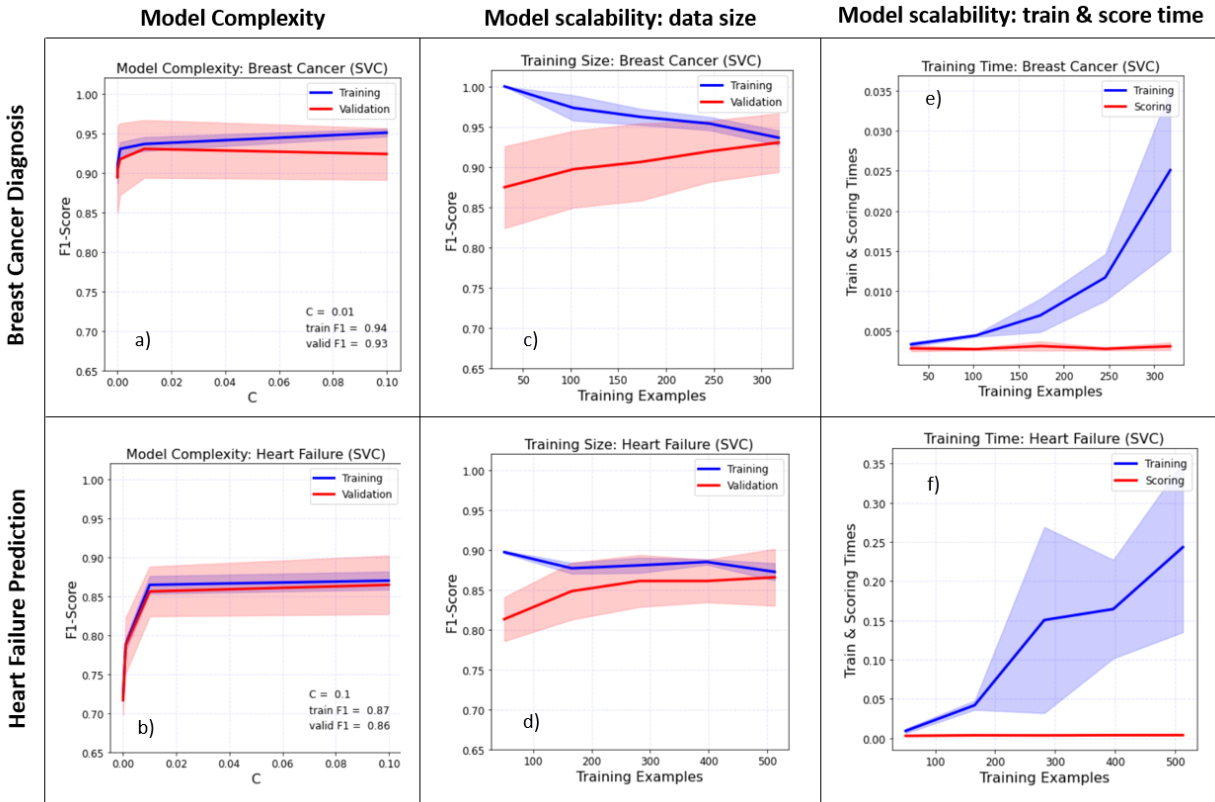


Figure 8: Comparison of model complexity and data size on Support Vector Classifier (SVC) performance during training and validation (5-fold cross validation) on two data. Performance is measured by F1 score.

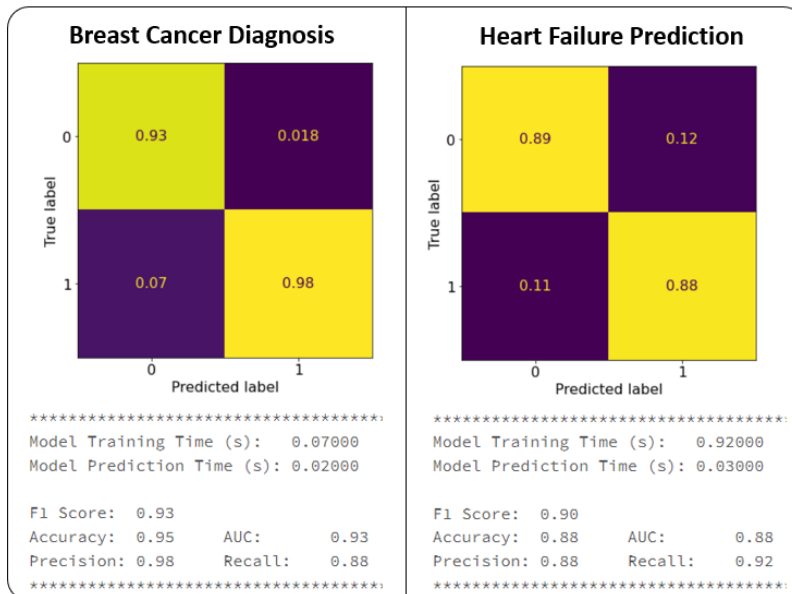


Figure 9: Confusion matrices depicting SVC's generalization performance on the two test data not seen during the training and validation.

4.5 k-Nearest Neighbors (KNN)

To minimize overfitting, $n_neighbors$ were set to of 5 and 17 for the two data (Figures 10 a & b). The corresponding F1-scores were high and relatively close (0.96 vs. 0.96 for WBCD and 0.87 vs. 0.87 for HFD). The impact of training data size on KNN overfitting is similar to DT, NN, GBT, and SVC (Figures 10 c & d). Computation time complexity during training is linear but constant during scoring as a function of data size (Figures 10 e & f). The trained KNN(s) on both datasets (Figure 11) have high predictive performance and comparable to other algorithms. The precision for positive breast cancer diagnosis is 1.0 while for a positive heart failure prediction is 0.87.

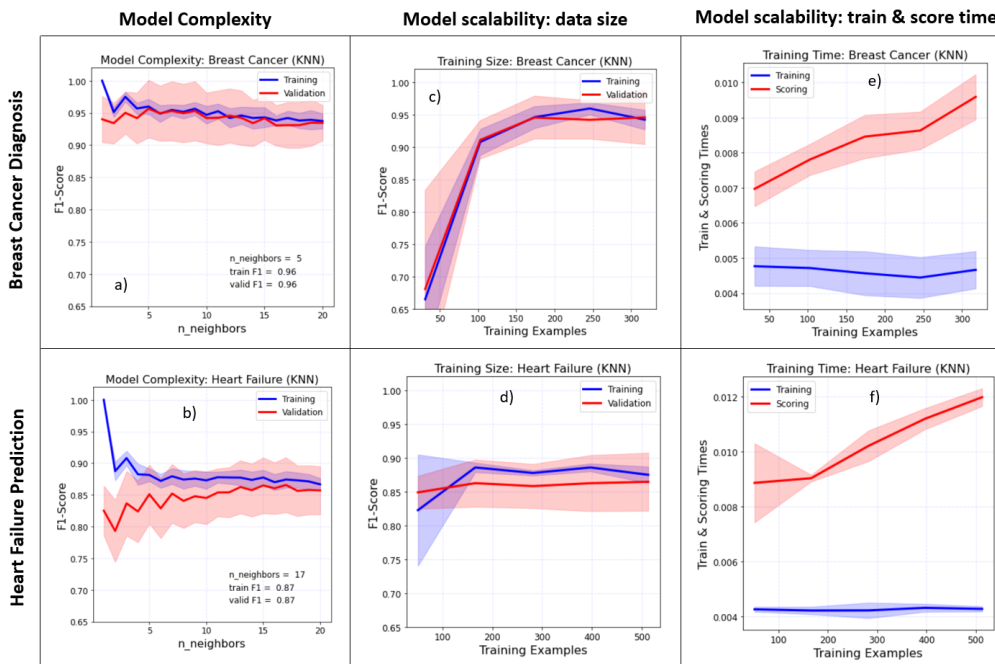


Figure 10: Comparison of model complexity and data size on k-Nearest Neighbors (KNN) performance during training and validation (5-fold cross validation) on two data. Performance is measured by F1 score.

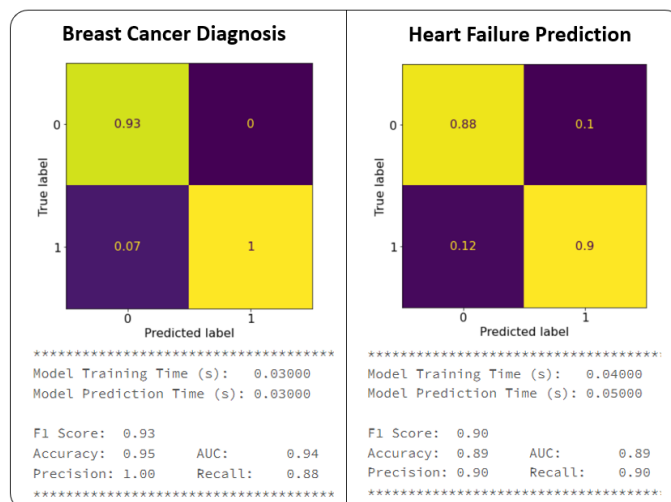


Figure 11: Confusion matrices depicting KNN's generalization performance on the two test data.

4.6 Comparison of models

Table 4 compares performance of the five algorithms on two medical datasets using seven performance metrics. The gradient boosting classifier (GBT) is the best on six of the seven metrics for the WBCD data. It is the best on four of the seven metrics for the HFP data. Thus, for both data, the GBT better captures the underlying cause and effect relations. The KNN's performance on both data is comparable to the GBT. The DT for both data has the best training and prediction time. The DT has a high predictive power; however, it is the least performer compared to other algorithms on these data.

Table 4: Comparison of algorithms on seven model performance metrics with test data. Top performance model is highlighted in green for each metric

	Breast Cancer Diagnosis							Heart Failure Prediction						
Algorithm	Accuracy	AUC	Precision	Recall	F1	Train time (s)	Prediction time (s)	Accuracy	AUC	Precision	Recall	F1	Train time (s)	Prediction time (s)
Decision Trees (DT)	0.91	0.88	0.98	0.77	0.86	0.02	0.02	0.80	0.80	0.83	0.82	0.82	0.03	0.01
Neural Networks (NN)	0.91	0.88	0.96	0.78	0.86	0.35	0.02	0.86	0.86	0.89	0.86	0.87	0.61	0.03
Boosting (GBT)	0.96	0.95	1.00	0.89	0.94	0.10	0.02	0.89	0.89	0.88	0.93	0.91	0.07	0.02
Support Vector Machines (SVM)	0.95	0.93	0.98	0.88	0.93	0.07	0.02	0.88	0.88	0.88	0.92	0.90	0.92	0.03
k-Nearest Neighbors (KNN)	0.95	0.94	1.00	0.88	0.93	0.03	0.03	0.89	0.89	0.90	0.90	0.90	0.04	0.05

4.7 What else to improve model performance?

The algorithms have multiple hyper-parameters. This analysis only focused on one or two, in isolation. Concurrent hyper-parameter tuning should improve each algorithm's performance. Also, a pre-modeling step of feature selection has the potential to improve model performance.

5.0 Conclusion

This analysis explored the impact of model complexity and model scalability on performance of five algorithms. All algorithms demonstrated overfitting with increased model complexity. Thus, hyper-parameter tuning is required to identify the optimal level of complexity. The analysis showed that more training data is one option to minimize overfitting. The SVC showed exponential training time with increase in data size, while other algorithms' training time trend was mostly linear. All algorithms had a constant prediction time. The GBT was the best algorithm while the DT was the least performer across seven model performance metrics.

References

1. Kaggle 2022. <https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>, accessed 02-11-2022
2. Cancer 2022. <https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html>
3. Fedesoriano 2022. <https://www.kaggle.com/fedesoriano/heart-failure-prediction>