ANALYSIS OF COMMON SUPERVISED LEARNING ALGORITHMS THROUGH APPLICATION

Palak Narula

Adobe Inc., Noida, India

ABSTRACT

Supervised learning is a branch of machine learning wherein the machine is equipped with labelled data which it uses to create sophisticated models that can predict the labels of related unlabelled data. the literature on the field offers a wide spectrum of algorithms and applications. However, there is limited research available to compare the algorithms making it difficult for beginners to choose the most efficient algorithm and tune it for their application.

This research aims to analyse the performance of common supervised learning algorithms when applied to sample datasets along with the effect of hyper-parameter tuning. for the research, each algorithm is applied to the datasets and the validation curves (for the hyper-parameters) and learning curves are analysed to understand the sensitivity and performance of the algorithms. The research can guide new researchers aiming to apply supervised learning algorithm to better understand, compare and select the appropriate algorithm for their application. Additionally, they can also tune the hyper-parameters for improved efficiency and create ensemble of algorithms for enhancing accuracy.

KEYWORDS

Decision Trees, Neural Networks, Boosting, Support Vector Machines, K-nearest neighbours

1. INTRODUCTION

ML and AI technologies are fuelling the growth of industries across the globe. There has been a rapid transformation and focus towards using these technologies to improve the existing systems as well as develop new systems that can understand the existing trends and predict new trends. A vast majority of these applications require supervised learning algorithms as a part of their complex systems. Though there exists a wide variety of algorithms that can be used for different applications, it is difficult for a beginner to decide the best-suited algorithm for the application. If the algorithm is not well-suited for the application, it can lead to issues (like reduced accuracy, difficulty in maintenance, increased downtime due to learning) when the application is deployed for actual clients.

The objective of this research is to bridge the gap between theory and real-world application of supervised learning algorithm for the new researchers. The analysis of the algorithms is done on two physical world applications – company bankruptcy prediction and breast cancer classification. The common supervised learning algorithms are applied to these datasets (split into training and testing data) and an analysed based on the validation curve for different values of hyper-parameter, learning curve for different data sizes and finally the accuracy and RMSE metrics of the entire data (before and after hyper-parameter tuning) [6,7]

DOI: 10.5121/acii.2023.10303

2. RELATED WORK

There are some researches available that study common supervised learning algorithms and aim to compare the algorithms. Iqbal Muhammad and Zhu Yan [1] surveyed various supervised learning algorithms through a theoretical approach highlighting various issues with supervised learning algorithms, common metrics for comparison and the internal working of the algorithms. R. Saravanan and Pothula Sujatha [2] reviewed, compared and classified supervised machine learning algorithms to give readers an overview of these algorithms in terms of data classification. The research classifies the algorithms as linear and probabilistic classifiers and highlights how a label is chosen for each category. Burkart, N. and Huber, M.F [3] provided essential definitions, principles and methodologies of supervised learning algorithms and conducted a survey to review the approaches.

The related researches have analysed the supervised learning algorithms theoretically and did not discuss about the practical approaches to compare the algorithms and understand and improve metrics through hyper-parameter tuning.

3. CLASSIFICATION PROBLEMS

3.1. Company Bankruptcy Prediction

As mentioned on Kaggle [4], this data was collected from Taiwan Economic journal from 1999 to 2009. This data s based on the business regulation of Taiwan Stock exchange. This data can be used for a binary classification problem where the historical combination of these features can be used to predict the probability of a company getting bankrupt in the future. The data includes around 95 features (almost all features being continuous) to define the current standing of the company in the Taiwan stock market. The data also includes two types of labels for these companies - bankrupt and not bankrupt.

3.2. Breast Cancer Classification

This dataset is also derived from Kaggle [5]. The data represents an important binary classification problem where the historical data can be used to understand the role of various features in defining the diagnosis of breast cancer tissue. It includes the historical data of around thirty features (almost all being continuous) to define whether the diagnosis of the breast tissue should be malignant (cancerous) or benign (non-cancerous).

4. DECISION TREES

Decision trees [8] are a type of supervised machine learning algorithm where the data is split continuously according to a certain parameters and finally the tree leaves represent the output/label of the data. Following hyper-parameters are chosen for tuning and analysis-

- Max depth - This parameter defines the max depth of the decision tree. This parameter forces the algorithm to prune the tree to avoid overfitting.
- Min_sample_split - This parameter defines the minimum number of samples that should be present at a node for splitting the node.

4.1. Validation Curves for Breast Cancer Data

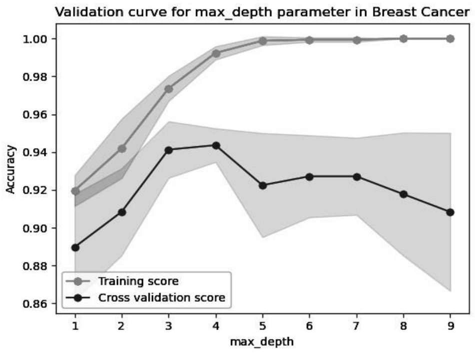

Figure 1. Validation curve for max_depth parameter

The above validation curve represents the accuracy of the decision tree classifier with different values of max depth of the tree. The graph shows that the accuracy of the training score keeps increasing which is expected since the classifier would try to create the maximum fitted decision tree from the training data with each leaf representing the label.

The next plot represents the accuracy of the training data and the cross-validation data for different values of minimum sample split. As expected, the training data has the maximum accuracy when there is no restriction on minimum number of samples for splitting the data. The accuracy decreases with the increased restriction.

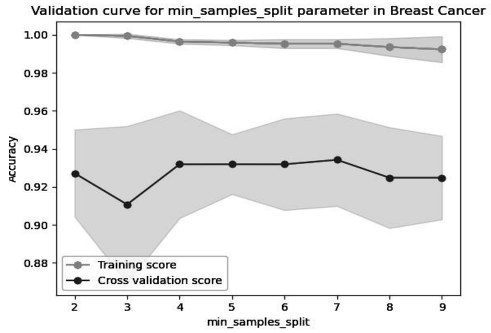

Figure 2. Validation curve for min_sample_split parameter

4.2. Comparison

Table 1. Comparison based on hyper-parameter tuning

Metric	**Default Parameters**	**Post hyper-parameter tuning**
Accuracy	91.61%	93.71%
Root mean squared error	8.39%	6.29%

The Decision tree classifier with the default parameters has also been able to provide a decent accuracy level but with the tuned hyper-parameters, (pruned decision tree and more restrictive rules for splitting the data) there was a limited scope of overfitting the training data and hence better accuracy.

4.3. Learning Curve for Breast Cancer Data

The plot represents the learning curve of Decision Tree with Breast Cancer Data. The model seems to have a decent learning with the accuracy of cross validation curve increasing with the increase in training data size. There is not much learning of the model post 200 sample size which can be pointed as the optimal size for the learning of Decision tree for breast cancer data.

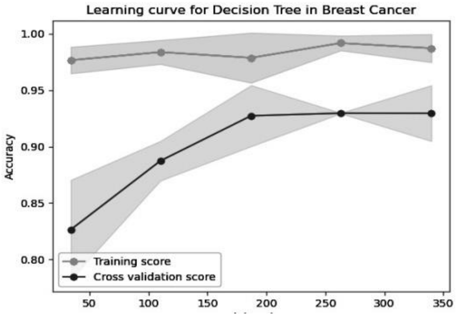

Figure 3. Learning curve for Breast cancer Data

4.4. Validation Curves for Bankruptcy Data

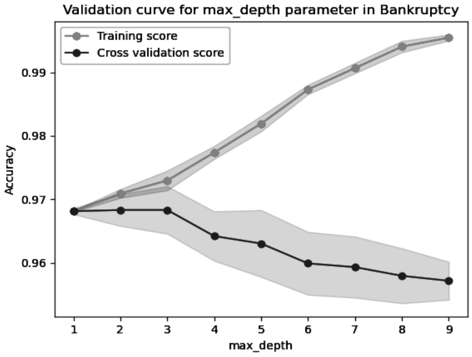

Figure 4. Validation curve for max_depth parameter

The above plot represents the validation curve for max_depth parameter of Decision tree for Bankruptcy data. Similar to the previous dataset, the curve for training score keeps improving accuracy with the increased value of max_depth as it tries to overfit the data.

The accuracy of the cross-validation remains constantly high upto max_depth=3 and then starts decreasing which indicates that the Decision tree classifier can work optimally with **max_depth = 3** (which is also the value suggested by the Grid Search [13])

The validation curve in fig.5 has features similar to the validation curve for Breast cancer data. The accuracy of the learning curve decreases with increasing restrictions in sample size for splitting. The cross-validation curve has a consistent accuracy for different splitting values with slightly higher value achieved at least restriction.

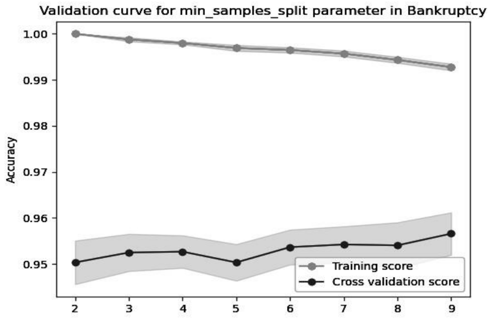

Figure 5. Validation curve for min_sample_split parameter

4.5. Comparison

Table 2. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	95.48%	96.36%
Root mean squared error	4.51%	3.63%

The performance metrics reflects an improvement in the accuracy of the results delivered by the Decision tree post tuning of the two hyper-parameters - max_depth and min_sample_split.

4.6. Learning Curve for Bankruptcy Data

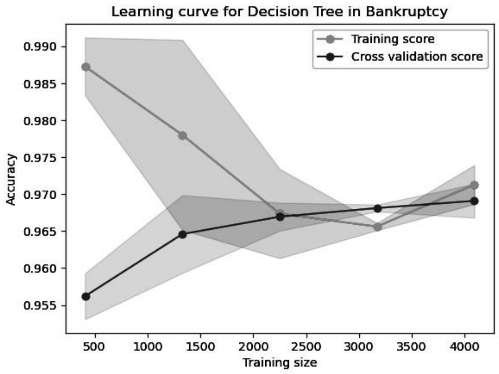

Figure 6. Learning curve for Bankruptcy Data

The learning curve for the data shows an ever-increasing line for the cross-validation data which indicates a good learning for the Decision tree classifier. The training data works best when the amount of the data is low, and the classifier can overfit the data easily. The accuracy of the training score keeps decreasing with the increasing data size. The two curves intersect at around 2250 data set size which should be the ideal data size for training the Decision tree for Bankruptcy data.

5. BOOSTING

Boosting [9] is a machine learning algorithm which works by fitting a weak classifier (Decision Tree by default) on the training data and then using the same classifier again with a higher focus on previously incorrectly classified data. The final algorithm is a combination of these weak classifier with adjusted weights. Following hyper-parameters are chosen for tuning-

- n_estimators - This parameter defines the maximum number of estimators where the boosting can stop.
- learning_rate - This defines the learning rate to be applied to each classifier during the iteration

5.1. Validation Curves for Breast Cancer Data

The following plot represents the validation curve for n_estimators parameter for Breast cancer dataset. The plot shows that the accuracy of training curve constantly increases until it reaches the maximum accuracy and then it remains constant. This is because the boosting algorithm tries to focus on incorrectly labelled data and with the increased number of estimators, the model tries to overfit the data. The cross-validation curve shows an increased accuracy with the increasing number of estimators, but the accuracy eventually decreases as the model tries to overfit the data. There is also an increasing variance with the increasing number of estimators. So, the plot suggests that the classifier should perform optimally for **n_estimators=90.**

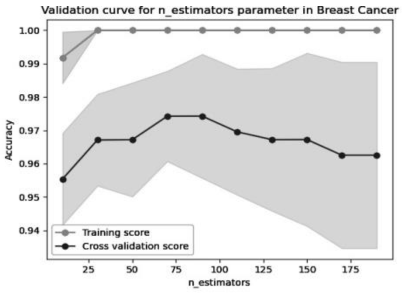

Figure 7. Validation curve for n_estimators parameter

The following plot represents the Validation curve for learning rate parameter with Breast Cancer dataset. The lines for both the training score and the validation scores seem to overlap for most of the parts in the plot and have a similar variance. The plot indicates that the classifier performs best for this data with **learning_rate = 1.0**

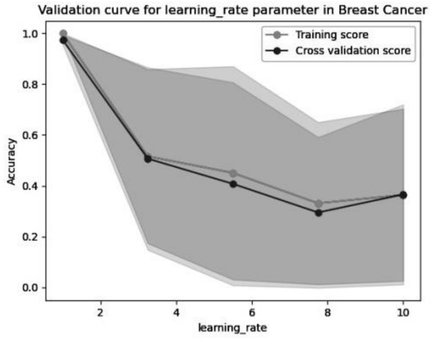

Figure 8. Validation curve for learning_rate parameter

5.2. Comparison

Table 3. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	95.80%	96.50%
Root mean squared error	4.19%	3.5%

Comparing the performance metrics of Breast Cancer data with Boosting algorithm, it can be observed that even though the algorithm uses multiple weak classifiers (Decision tree in this case), it has a better overall performance than the decision tree. Also, the performance improves with tuning of hyper-parameters.

5.3. Learning Curve for Breast Cancer Data

The learning curve shows a continuous improvement in the learning of the model for both the learning and the cross-validation curves. Thus, for the Breast Cancer Data, the Boosting algorithm will perform better with the use of entire learning data.

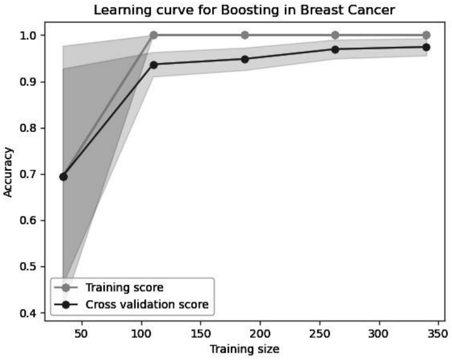

Figure 9. Learning curve for Breast cancer Data

5.4. Validation Curves for Bankruptcy Data

The next plot represents the validation curve for n_estimators parameter with Bankruptcy dataset. Similar to the plot for Breast cancer, the curve for the training data keeps improving until the maximum accuracy is achieved indicating that the classifier is trying to overfit data with increased number of weak classifiers.

The cross_validation curve indicates that the optimal value for this parameter for the given dataset should be around 125.

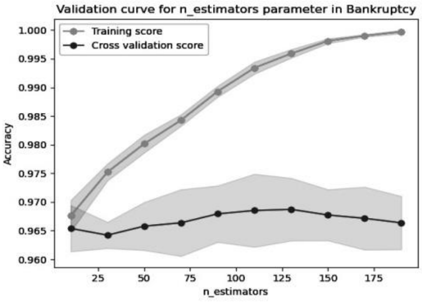

Figure 10. Validation curve for n_estimators parameter

The validation curve for learning_rate for Bankruptcy dataset is similar in its properties to Breast Cancer dataset. The classifier performs most optimal with **learning_rate = 1.0** and as we increase the weight of weak classifiers, the performance of the Boosting algorithm decreases.

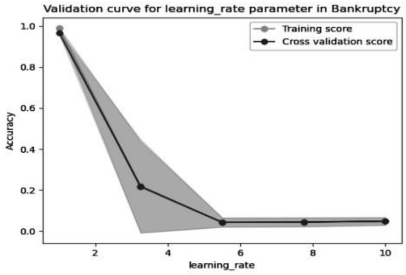

Figure 11. Validation curve for learning_rate parameter

5.5. Comparison

Table 4. Comparison based on hyper-parameter tuning

Metric	**Default Parameters**	**Post hyper-parameter tuning**
Accuracy	96.71%	96.83%
Root mean squared error	3.28%	3.17%

5.6. Learning Curve for Bankruptcy Data

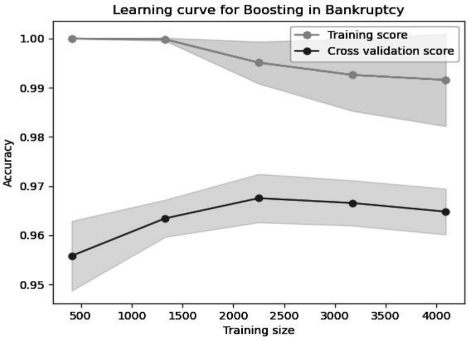

Figure 12. Learning curve for Bankruptcy Data

The learning curve suggests that the boosting algorithm will have the most optimal learning for data size = 2250 where the cross-validation curve peaks. For the lower data size, the algorithm tries to overfit the data as is evident from the training score curve and for the higher data sizes, the algorithm seems to be generalising, so the accuracy is decreasing.

6. SUPPORT VECTOR MACHINES

Support vector machines [10] are a type of machine learning algorithm that works by selecting a hyperplane in an N-dimensional space to classify the data points. Following hyper-parameters are chosen for tuning and analysis here-

- kernel- This parameter specifies the kernel type to be used in the algorithm. We will be focusing on the following four types - linear, poly, rbf and sigmoid
- max_iter - This parameter defines the maximum number of iterations that the solver can use

6.1. Validation Curves for Breast Cancer Data

The next plot represents the accuracy of the SVM classifier with different available kernels for Breast Cancer data. The curve represents four distinct values for different types of kernels so nothing can be deduced from the shape of the curve.

For both the training and validation scores, the linear Kernel seems to be performing most optimally followed by rbf and poly kernels and finally sigmoid doesn’t seem to fit well for the breast cancer data. So, it seems for the Breast Cancer data, **linear kernel for SVM** would be the best choice.

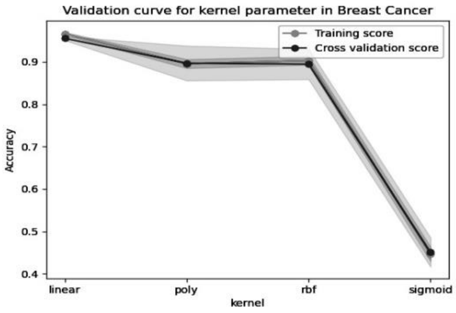

Figure 13. Validation curve for kernel parameter

Next, we analyse the algorithm for max_iter parameter. For the analysis, I chose the linear kernel and plotted the validation curve for different values of max_iter parameter. The curve shows an increasing accuracy with an increased limit in maximum number of allowed iterations. The curve eventually flattens because the classifier starts to converge before utilising the maximum available limit. Since the classifier is converging pretty fast, selecting **max_iter = 60** safely allows the classifier to work at its optimal capacity.

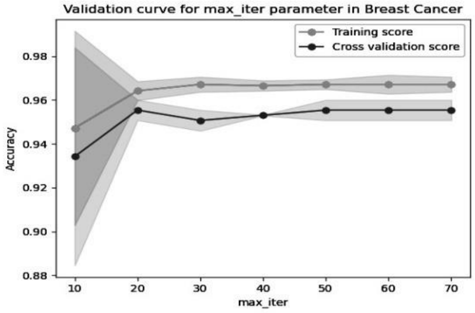

Figure 14. Validation curve for max_iter parameter

6.2. Comparison

Table 5. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	95.10%	97.20%
Root mean squared error	8.39%	2.79%

The performance metrics indicates a significant improvement in the accuracy of the SVC with parameter hyper-tuning. The accuracy is also better than the previous two machine learning classifiers- decision tree and boosting.

6.3. Learning Curve for Breast Cancer Data

The learning curve indicates a continuous improvement in the accuracy of the classifier with the increasing training data size. The training score is highest when the available data is low since overfitting is possible but with increased data, the bias is reduced, and cross-validation score keeps improving. Overall, it seems that the maximum available data should be used to train an SVC for Breast Cancer data.

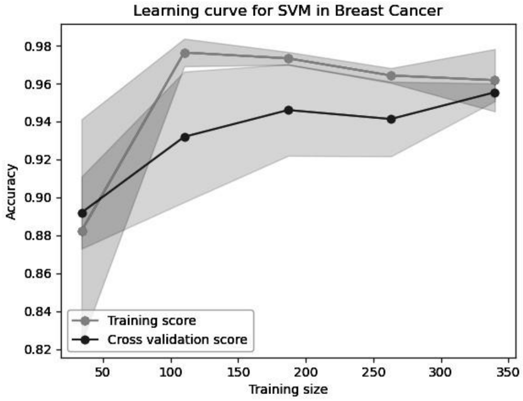

Figure 15. Learning curve for Breast cancer Data

6.4. Validation Curves for Bankruptcy Data

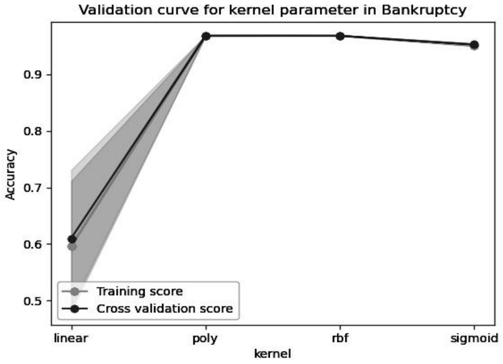

Figure 16. Validation curve for kernel parameter

The above curve indicates the discrete accuracy values for different possible kernels for SVC. For the Bankruptcy data, linear kernel seems to be performing the worst. The other three kernels have a comparable accuracy with a slightly higher values for poly and rbf.

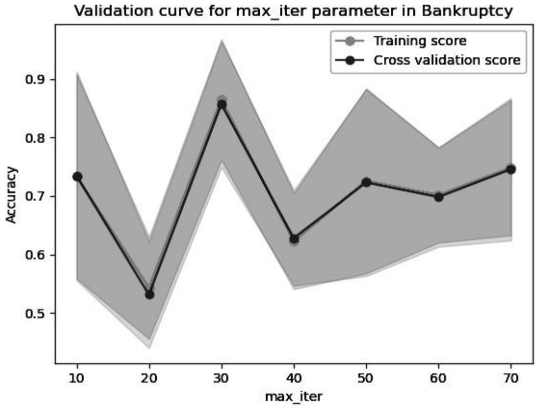

Figure 17. Validation curve for max_iter parameter

Since the SVC with rbf kernel was converging so quickly that the graph was a straight line with maximum accuracy from max_iter=1. So, to analyse the parameter better, I chose the worst performing kernel, that is linear kernel and plotted the above graph for various values of max_iter. The above validation curve shows high variance in accuracy metric for Bothe the training and cross validation scores. There is also a great fluctuation in the accuracy of the results. Since the linear kernel doesn’t fit well for the Bankruptcy dataset, it is important to put a hard stop on number of iterations to avoid the classifier to run forever. In this scenario, the fluctuations in the accuracy values seems to stabilise after max_iter = 50, so that should be value for tuning the max_iter hyper-parameter

6.5. Comparison

Table 6. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	96.65%	96.65%
Root mean squared error	3.34%	3.34%

Since the default SVC kernel is rbf and the classifier converges fast enough that the max_iter won't affect it's processing so the performance metrics of default parameters is same as the classifier with tuned hyper-parameters.

6.6. Learning Curve for Bankruptcy Data

The learning curve of SVC with Bankruptcy data is quite different from the learning curves of previous classifier. In this the cross-validation score remains constantly higher than the learning score. This is probably because the classifier is able to fit data and converge quickly even with the partial data, but it learns better with increased data size. So, for this dataset, SVC should perform best with the maximum possible data size (more than 4000 data entries in this case).

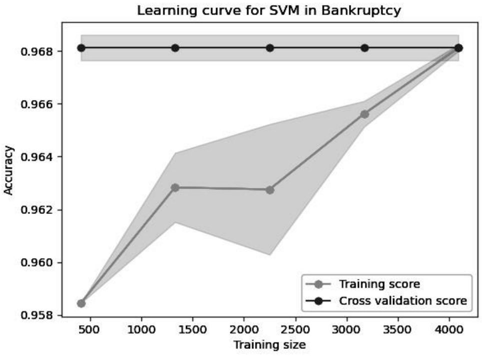

Figure 18. Learning curve for Bankruptcy Data

7. Neural Networks

Neural network [11] is a supervised learning algorithm inspired by the human neural system where the input is processed by a number of neutrons arranged in layers and a relationship between the inputs and labels is generated. Following hyper-parameters are chosen for tuning and analysis here-

- hidden_layer_sizes - It takes tuple as an input where the ith element represents the number of neutrons in the ith layer
- max_iter - This parameter represents the upper limit on the number of iterations that the solver can use

7.1. Validation Curves for Breast Cancer Data

The below validation curve is plotted by varying the number of neurons for a single hidden layer in the neural network. The curve shows continuous fluctuations and high variance in the accuracy of the classifier for lower hidden layer sizes. Eventually both the learning and crossvalidation scores stabilises around hidden_layer_size=60 with the maximum score at **hidden_layer_size=80** which should be the value of the hyper-parameter tuning.

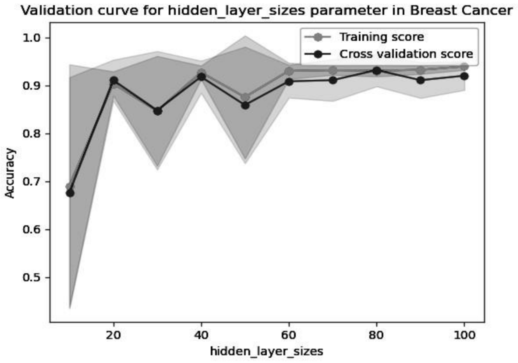

Figure 19. Validation curve for hidden_layer_sizes parameter

The validation curve for max_iter parameter shows a continuous improvement in the accuracy of the classifier with the increased value of max_iter parameter since the classifier will get a greater number of iterations and thus will be able to perform better. Both the training and crossvalidation curves begin to flatten around 140 so that should be the optimal value of **max_iter parameter = 140**

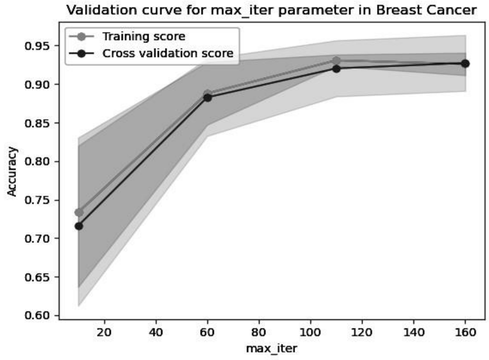

Figure 20. Validation curve for max_iter parameter

7.2. Comparison

Table 7. Comparison based on hyper-parameter tuning

Metric	**Default Parameters**	**Post hyper-parameter tuning**
Accuracy	93%	93%
Root mean squared error	6.99%	6.99%

Since the default value of both the parameters (hidden_layer_sizes and max_iter) are much higher than our chosen values, the neural network classifier with its default hyper-parameters perform same as the classifier with the optimal setting of the hyper-parameters. But, by setting the hyper-parameters, a lot of unnecessary computation can be saved, and the classifier will take lesser time to complete.

7.3. Learning Curve for Breast Cancer Data

The learning curve shows a continuous improvement in the accuracy of the classifier with the increase in data size but the variance decreases making the results more reliable. The curve suggests using data size = around 260 for most optimal results with the Breast Cancer data while using Neural Network classifier.

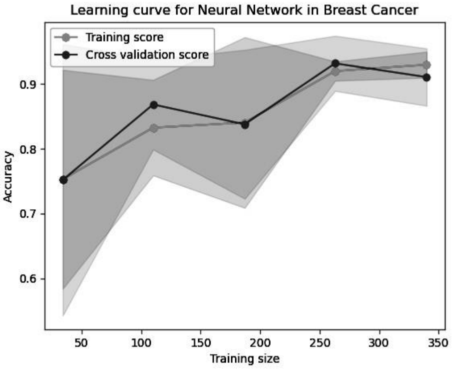

Figure 21. Learning curve for Breast cancer Data

7.4. Validation Curves for Bankruptcy Data

The validation curve for the hidden layer size parameter shows a decrease in the accuracy with increase in the number of neurons in the hidden layer. The accuracy eventually increases to a maximum value with some fluctuations and then starts to decrease. This behaviour of the curve is common for both the training and cross-validation scores. The variance in the accuracy is also high for low hidden layer size which is expected since the classifier will get fewer number of neurons to map the input to labels correctly. Looking at the graph, the most optimal value for hidden_layer_size parameter is 80.

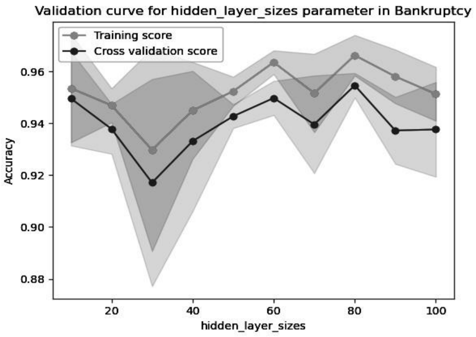

Figure 22. Validation curve for hidden_layer_sizes parameter

The validation curve for max_iter parameter shows an increase in the accuracy with the increase in the value of the parameter till max_iter=60 and then the accuracy starts decreasing and the variance starts increasing for both the learning and cross-validation curves. The decrease in accuracy indicates that the classifier tries to overfit data when it is allowed more restrictions so choosing **max_iter=60** should provide an optimal accuracy to the neural network classifier.

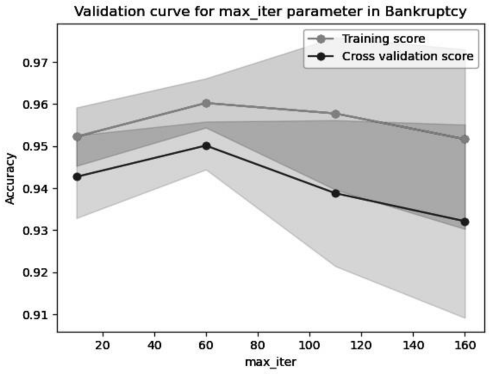

Figure 23. Validation curve for max_iter parameter

7.5. Comparison

Table 8. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	93.19%	91.73%
Root mean squared error	6.8%	8.26%

The performance metrics indicate a lower performance of the classifier post tuning of hyperparameters for bankruptcy data. This is probably because the hyper-parameter values chosen in the process to make it optimal make the classifier much more restrictive allowing lesser number of neurons and forcing the algorithm to converge quickly. To improve this, more experiments can be performed with a different range for the hyper-parameter values.

7.6. Learning Curve for Bankruptcy Data

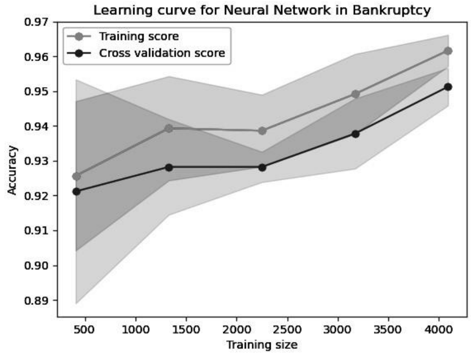

Figure 24. Learning curve for Bankruptcy Data

The learning curve of the neural network with Bankruptcy data shows a continuous improvement in the accuracy with an increased training size. There is also a reduction in the variance which indicates that for complicated dataset like Bankruptcy data, a higher data size would result in better learning for Neural Networks since it will enable the classifier to map the input to labels better.

8. K-NEAREST NEIGHBOURS

K nearest neighbours [12] is a supervised learning algorithm that works by assigning label based on class membership. The algorithm works on the belief that a data unit should have the same label as the majority adapt units around it have.

Following hyper-parameters are chosen for tuning and analysis here-

- n_neighbours - this parameter determines the number of neighbours that should be considered by the classifier
- metric - this parameter defines the distance function that should be used by the classifier

8.1. Validation Curves for Breast Cancer Data

The below validation curve shows highest accuracy of the training curve for n = 0 and eventual decrease in accuracy which reflects overfitting the data. The classifier tries to map each data point to the corresponding label when it doesn't have to check the value of its' neighbours and the accuracy deceases as the number of neighbours to be consulted increases. The crossvalidation curve provides a better picture on choosing the most optimal values of n. The curve has approximately similar from 5 to 13 with decreased values in the beginning and end of the curve. The most optimal value from the plot is at **n_neighbours = 7** which has the highest accuracy achieved by the cross-validation curve.

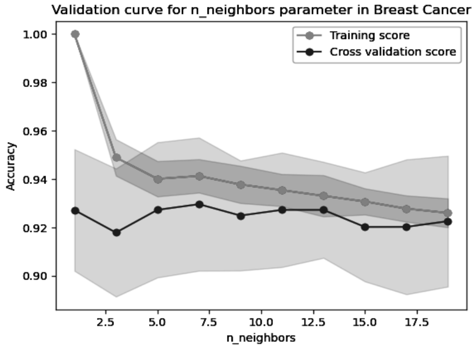

Figure 25. Validation curve for n_neighbours parameter

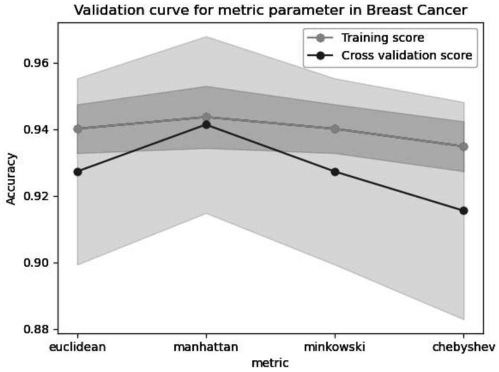

Figure 26. Validation curve for metric parameter

The validation curve for metric parameter shows distinct values of the classifier for different metrics chosen to represent the distance of the data point from its peers. Both the learning and cross-validation curves have peak accuracy with Manhattan distance so **metric=Manhattan** seems to be the most optimal setting.

8.2. Comparison

Table 9. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	94.4%	95.1%
Root mean squared error	5.59%	4.89%

As expected, the performance metrics indicate an improvement in the accuracy of the K nearest neighbour classifier post tuning of hyper-parameters.

8.3. Learning Curve for Breast Cancer Data

The learning curve of the KNN classifier with Breast cancer data shows continuous improvement in the accuracy of the predictions with higher slop in the beginning and a lower slope for higher values of the data size. Since there is no continuous decrease in the performance of the classifier, using the entire available data would be suitable for getting optimal results with KNN.

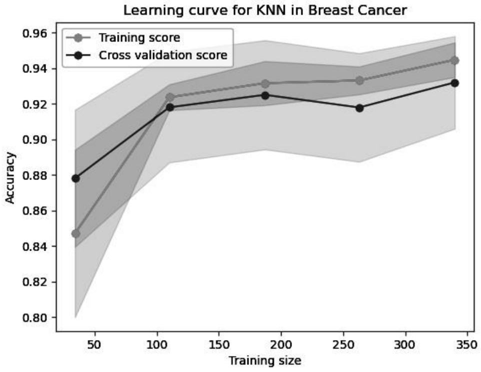

Figure 27. Learning curve for Breast cancer Data

8.4. Validation Curves for Bankruptcy Data

The curve below shows how the number of neighbours selected for KNN affects the accuracy of the algorithm. As expected, the learning curve has the highest accuracy for n=0 as it tries to map each data to the label thus overfitting the data. As the n increases, the accuracy starts decreasing. The cross-validation curve provides a better insight about the performance of the classifier. The accuracy of the classifier increases with increase in number of neighbours upto a value and then stays constant. This point indicates the most optimal value of the parameter, that is, **n_neighbours=7**

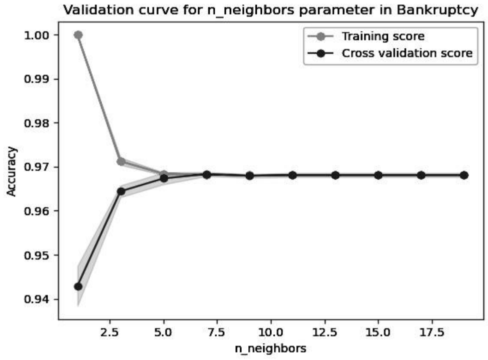

Figure 28. Validation curve for max_depth parameter

The validation curve below represents the distinct accuracy levels with different metric parameter that can be used to measure distance of a data point from its peer. Since, the maximum accuracy is achieved for Manhattan metric, it should be considered for tuning the hyper-parameter. But the GridSearchCV algorithm suggest **metric=Euclidean**. The euclidean distance also provides a comparable accuracy, it might be performing better when combined with other hyper-parameters

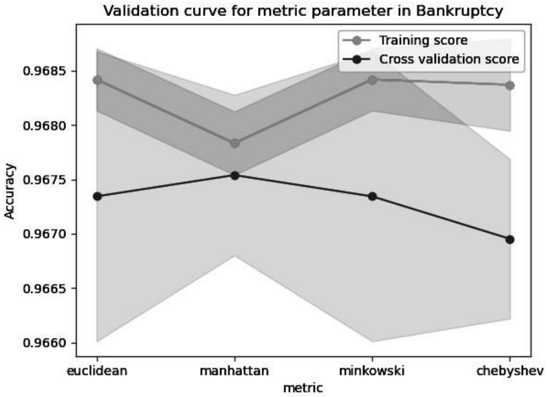

Figure 5. Validation curve for n_jobs parameter

8.5. Comparison

Table 10. Comparison based on hyper-parameter tuning

Metric	Default Parameters	Post hyper-parameter tuning
Accuracy	96.65%	96.65%
Root mean squared error	3.34%	3.34%

There is no improvement in the accuracy of the classifier by tuning of hyper-parameters probably because we tune parameters looking at the cross-validation scores but when it comes to testing data, the default parameters may perform as good as the tuned hyper-parameters

8.6. Learning Curve for Bankruptcy Data

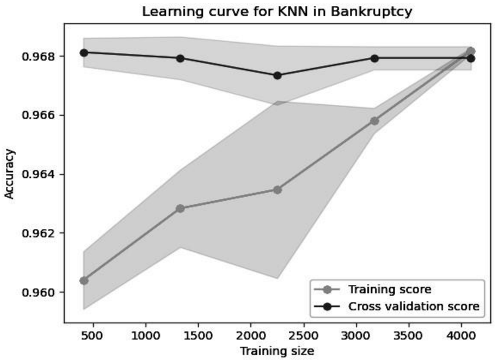

Figure 6. Learning curve for Bankruptcy Data

The training score in the curve shows a continuous improvement in the learning of the algorithm with increasing data size though the cross-validation curve shows a a stable accuracy irrespective of the data size. Honouring both the curves, choosing maximum data size for training the classifier seems to be providing the best possible accuracy.

9. CONCLUSIONS

9.1. Summarised Analysis

- The smaller datasets are faster to train but can lead to overfitting of data. An optimised size of dataset can be achieved by applying dimensionality reduction techniques.
- Hyper-parameter tuning helps in improving the accuracy of the algorithms, but it is a complicated process that requires understanding the individual and combined effect of hyper-parameter on the algorithm.
- The performance metrics should be defined before choosing an algorithm as a single algorithm can't satisfy all the metrics.
- An ensemble of algorithms might have better metrics than individual algorithms.

Table 11. Overall comparison of the five algorithms

Metric	**Best Algorithm**	**Worst Algorithm**
Accuracy (Bankruptcy data)	Boosting	Neural Networks
Accuracy (Breast Cancer data)	SVM	Neural Networks
Clock Time (Training)	Neural Networks	SVM
Clock Time (Testing)	KNN	KNN

9.2. Limitations

The research is limited to covering only a few common supervised learning algorithms with the tuning of only two hyper-parameters per algorithm. There are a wide spectrum of application, algorithms, hyper-parameters, and metrics that are beyond the scope of this research but are important for more advanced analysis and understanding of the algorithms.

9.3. Future Scope

As future work, the analysis in this research can be used to make an informed decision while selecting an algorithm (or an ensemble of algorithms) for a particular application.Additionally, the research can also be taken forward to analyse new algorithms and parameters that are not covered in this paper. Applying the research on more datasets would also help the reader understand the nature of application and the best-suited algorithm.

REFERENCES

[1] Iqbal, Muhammad & Yan, Zhu. (2015). SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. International Journal of Soft Computing. 5. 946-952. 10.21917/ijsc.2015.0133

[2] R. Saravanan and P. Sujatha, "A State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 945-949, doi: 10.1109/ICCONS.2018.8663155.

[3] Burkart, N. and Huber, M.F. A survey on the explainability of supervised machine learning. Journal of Artificial Intelligence Research.

[4] Fedesoriano, “Company bankruptcy prediction,” Kaggle, 13-Feb-2021. [Online]. Available: https://www.kaggle.com/fedesoriano/company-bankruptcy-prediction.

[5] kaggle.com. (n.d.). Breast Cancer Dataset Classification. [online] Available at: https://www.kaggle.com/dhainjeamita/breast-cancer-dataset-classification/data

[6] Mitchell, T.M. (1997). Machine learning. New York: Mcgraw-Hill

[7] omscs.gatech.edu. (n.d.). CS 7641: Machine Learning Course Videos | OMSCS | Georgia Institute of Technology | Atlanta, GA. [online] Available at: https://omscs.gatech.edu/cs-7641-machine-learningcourse-videos

[8] Scikit-learn.org. (2009). sklearn.tree.DecisionTreeClassifier — scikit-learn 0.21.3 documentation. [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeCl assifier.

[9] scikit-learn.org. (n.d.). sklearn.ensemble.AdaBoostClassifier — scikit-learn 0.22.1 documentation. [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html.

[10] Scikit-learn.org. (2019). sklearn.svm.SVC — scikit-learn 0.22 documentation. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC.

[11] scikit-learn.org. (n.d.). sklearn.neural_network.MLPClassifier — scikit-learn 0.24.1 documentation. [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network. MLPClassifier.

[12] scikit-learn developers (2019). sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.22.1 documentation. [online] Scikit-learn.org. Available at: https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.

[13] Scikit-learn.org. (2019). sklearn.model_selection.GridSearchCV — scikit-learn 0.22 documentation. [online] Available at: https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

AUTHOR

Palak Narula gained her BTech in Computer Science from H.B.T.I, India and MS in Computer Science (major in Machine Learning) from Georgia Institute of Technology, Atlanta, GA. Currently she is working as a Computer Scientist in Adobe Noida, India.