Name: Sol Won

Machine Learning Supervised Learning

Description of classification problems:

Two datasets are used to analyze the different machine learning algorithms. (Breast cancer Wisconsin data and Diabetic Retinopathy Debrecen dataset)

Breast Cancer Wisconsin (Diagnostic) Dataset: There are total of 30 attributes of cell nucleus (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension (SE, mean and worst)) and they are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Classification problem is to predict whether the tissue is benign or malignant by looking at those 30 attributes. The reason of using this dataset is as follows. First, there are lot of attributes in the dataset. For instance, there are three types of radius which are radius SE, radius mean and radius worst. This made the dataset attractive in a sense that there might be tons of different results or changes that can happen. Moreover, the dataset is well distributed. There are 357 benign cases and 212 malignant cases. The dataset is not perfectly balanced (which is 50% for each class) but good enough to be applied and analyzed.

Diabetic Retinopathy Debrecen Dataset: This dataset consists of attributes from Messidor image to predict whether an image has diabetic retinopathy sign or not. There are total of 18 attributes from the image (quality, prescreening, MA detection results, exudates, Euclidean distance, diameter, binary result of classification). Classification problem is to predict whether the image has sign of diabetic retinopathy or not. This dataset is used since it is well balanced. There are 540 positive cases and 611 negative cases. Also, I personally found this dataset interesting since I always wonder about how well I can train the information based on the image.

Machine learning algorithm analysis:

There are 5 machine learning algorithms used for this analysis. (Decision tree, neural network, SVMs (rbf and linear), Adaboost and KNN (with different K values))

Each analysis contains confusion matrix, classification report, ROC curve, learning & validation curve, hyperparameter tuning, confusion matrix after tuning, classification report after tuning and roc curve after tuning.

Decision Tree:

In the decision tree classifier, Gini criterion is used. Gini measures the probability of a variable that is wrongly classified when it is randomly chosen. Gini impurity measurement can be used for this analysis since the datasets are binary classified. Max depth is set to none for pruning purposes and hyperparameter tuning will happen later to see the comparison. Random state of 1 is used to see the consistent graph.

	Predict [0]	Predict [1]
True [0]	137	6
True [1]	12	73

Figure1: Confusion matrix of dataset 1

Confusion matrix is used to see the result of prediction. There are 137 True Positive, 6 False Negative, 12 False Positive, 73 True Negative. By looking at the confusion matrix, the prediction is considered as good since there are 137 records that the benign cases are predicted benign and 73 records that the malignant cases are predicted malignant.

	Predict [0]	Predict [1]
True [0]	138	78
True [1]	107	138

Figure 2: Confusion matrix of dataset 2

For second dataset, there are 138 True Positive, 78 False Negative, 107 False Positive, 138 True Negative. By looking at the confusion matrix, the prediction is not very good since there are 78 records and 107 records that the machine predicted wrong. There are total of 276 records that the machine predicted right.

	precision	recall	F1-score	Support
0	0.92	0.96	0.94	143
1	0.92	0.86	0.89	85
Accuracy			0.92	228
Macro avg	0.92	0.91	0.91	228
Weighted avg	0.92	0.92	0.92	228

Figure 3: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.56	0.64	0.60	216
1	0.64	0.56	0.60	245
Accuracy			0.60	461
Macro avg	0.60	0.60	0.60	461
Weighted avg	0.60	0.60	0.60	461

Figure 4: Classification report of dataset 2

Figure 3 and 4 is a classification report for dataset 1 and 2. These table shows 4 different scores (accuracy, precision, recall and f1-score). Individual scores have different purposes of usage but for this analysis, Accuracy and F1-score is used to compare the results. Dataset 1 has accuracy of 0.92 with F-1 score of 0.94 for the first class and 0.89 for the second class. Accuracy and f1-score are considered as good since it is above 0.9. Dataset 2 has a result with accuracy of 0.60 and f1-score 0.60 for both classes. Scores of Dataset 2 still need some improvement.

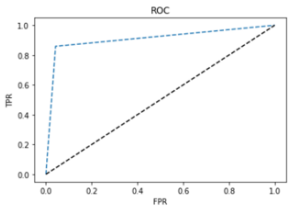

Figure 5: ROC curve for dataset 1

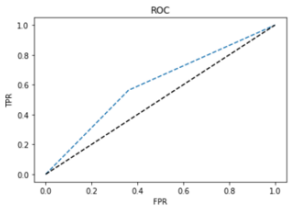

Figure 6: ROC curve for dataset 2

ROC (receiver operating characteristic curve) curve is a graphical illustration of binary classification system as its discrimination threshold is varied. Figure 5 shows excellent TPR and FPR which is above 0.9 which means the decision tree model has good sensitivity and specificity for this dataset. For figure 6, it shows that the dataset 2 has TPR score of 0.6. The values are above 0.5 so it is in a positive side, but sensitivity and specificity does not belong to excellent range.

For validation, 5-fold cross validation was used for both datasets. In this case, validation ran for the whole process 5 times with 5 section divided data and the accuracy was calculated by getting the average. First dataset has cross validation accuracy of 0.918 +/- 0.012 and second dataset have cross validation accuracy of 0.574 +/- 0.026.

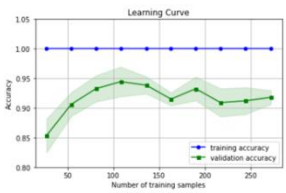

Figure 7: Learning curve for dataset 1

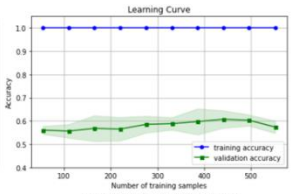

Figure 8: Learning curve for dataset 2

Figure 7 and 8 is a graph of accuracy with X-axis of 'number of training samples'. Figure 7 shows a good accuracy scores but there is overfitting issue (high variance). It is because there is some gap between training accuracy and validation accuracy. It might be because there was not much types of samples trained, For figure 8, it is suffering from underfitting. Validation accuracy goes up slowly but it drops at the end which means validation accuracy cannot follow the trainig accuracy.

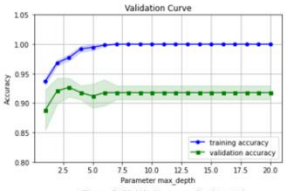

Figure 9: Validation curve for dataset 1

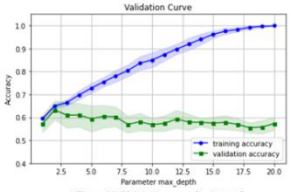

Figure 10: Validation curve for dataset 2

Figure 9 shows the underfitting problem for the first few depth but start to have overfitting problem. Training accuracy keeps on going up and reach 100% accuracy but testing accuracy does not. For figure 10, it also shows the overfitting problem since validation accuracy cannot follow the training accuracy. By just looking at the graph, ideal max depth for figure 9 is 3 and max depth 2 for figure 10.

Max depth parameter is used to tune the hyperparameter. For max depth, range of 1 to 20 is used. As a result, first data had optimal depth of 3. For second dataset, depth of 2 was optimal. It matches like the validation curve result.

Training time was also calculated. Training time for dataset 1 is 2.29 seconds and training time for dataset 2 was 2.36 which is fast enough.

	Predict [0]	Predict [1]
True [0]	140	3
True [1]	13	72

Figure 11: Confusion matrix of dataset 1

After hyperparameter tuning, data was fit to the tree with the tuned depth and leaf size to compare to the original values. Compare to the first confusion matrix, there were more truly predicted values for both classes.

	Predict [0]	Predict [1]
True [0]	183	33
True [1]	133	112

Figure 12: Confusion matrix of dataset 2

For dataset 2, the result was interesting. There are fewer truly predicted values for second classes. The reason might be due to the pruning. Confusion matrix before hyperparameter tuning had max depth of none.

	precision	recall	F1-score	Support
0	0.92	0.98	0.95	143
1	0.96	0.85	0.90	85
Accuracy			0.93	228
Macro avg	0.94	0.91	0.92	228
Weighted avg	0.93	0.93	0.93	228

Figure 13: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.58	0.85	0.69	216
1	0.77	0.46	0.57	245
Accuracy			0.64	461
Macro avg	0.68	0.65	0.63	461
Weighted avg	0.68	0.64	0.63	461

Figure 14: Classification report of dataset 2

Figure 13 and 14 shows the classification report after the hyperparameter tuning. This is to compare the result before and after the hyperparameter tuning. Dataset 1 had a result where it increased overall values including accuracy (0.92 to 0.93) and f1-scores. For the second dataset, most of the values increased as well (accuracy: 0.60 to 0.64). ROC curve is skipped because the graph looks very similar to the initial graph since there were not that much of a change (just a little improvement).

Cross validation score also increased from 0. 0.918 +/- 0.012 to 0.927 +/- 0.016 for the first dataset and 0.574 +/- 0.026 to 0.632 +/- 0.041 for the second dataset.

A Neural Network:

In the mlp classifier, Adam solver is used. Different solvers were used such as lbfgs but adam solver seem to have better accuracy. Adam solver is a stochastic gradient based optimizer. Max iteration value of 1000 is used to solve the iteration warnings. Random state of 1 is used to see the consistent graph.

	Predict [0]	Predict [1]
True [0]	135	8
True [1]	8	77

Figure 15: Confusion matrix of dataset 1

Confusion matrix is used to see the result of prediction. There are 135 True Positive, 8 False Negative, 8 False Positive, 77 True Negative. By looking at the confusion matrix, the prediction is in a good state.

	Predict [0]	Predict [1]
True [0]	184	32
True [1]	82	163

Figure 16: Confusion matrix of dataset 2

For second dataset, there are 184 True Positive, 32 False Negative, 82 False Positive, 163 True Negative. By looking at the confusion matrix, the prediction is pretty good compare to the decision tree confusion matrix.

	precision	recall	F1-score	Support
0	0.94	0.94	0.94	143
1	0.91	0.91	0.91	85
Accuracy			0.93	228
Macro avg	0.92	0.92	0.92	228
Weighted avg	0.93	0.93	0.93	228

Figure 17: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.69	0.85	0.76	216
1	0.84	0.67	0.74	245
Accuracy			0.75	461
Macro avg	0.76	0.76	0.75	461
Weighted avg	0.77	0.75	0.75	461

Figure 18: Classification report of dataset 2

Figure 17 and 18 is a classification report for dataset 1 and 2. Dataset 1 has accuracy of 0.93 with F-1 score of 0.94 for the first class and 0.91 for the second class. As same with decision tree, neural network confusion matrix also have high scores. Dataset 2 has a result with accuracy of 0.75 and f1-score 0.76 for class 0 and 0.74 for second class.

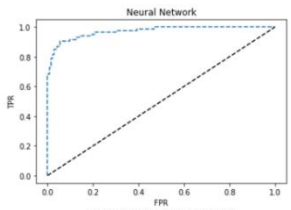

Figure 19: ROC curve for dataset 1

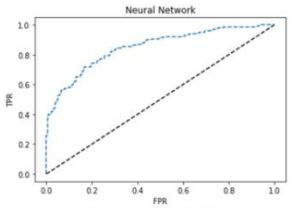

Figure 20: ROC curve for dataset 2

Figure 19 and 20 is a ROC curve. Figure 5 shows excellent TPR and FPR which is above 0.9 which we can say the neural network model has good sensitivity and specificity for this dataset. For figure 6, it shows that the dataset 2 has fair TPR and FPR with neural network model.

For validation, 5-fold cross validation was used for both datasets. First dataset has cross validation accuracy of 0.921 +/- 0.044 and second dataset have cross validation accuracy of 0.730 +/- 0.031.

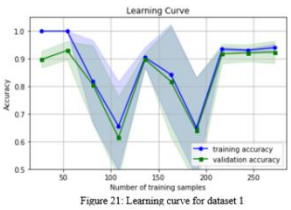

Figure 21: Learning curve for dataset 1

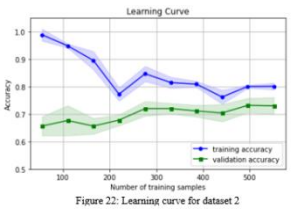

Figure 22: Learning curve for dataset 2

Figure 21 and 22 is a graph of accuracy based on increasing volumn of sample sizes. Figure 21 shows accuracy scores fluctuating but overall, training and testing data accuracy matches well. It means the data is well trained and tested. For figure 8, it is suffering from underfitting. Training

set and validation set accuracy is getting closer but their accuracy score in general is not excellent (under 0.8). So the model has high bias.

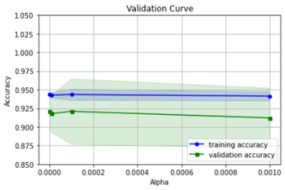

Figure 23: Validation curve for dataset 1

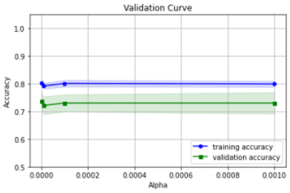

Figure 24: Validation curve for dataset 2

Figure 23 shows the overfitting problem. Training accuracy keeps on going down in a steady rate. Test accuracy keeps decreasing like training rate but even more as alpha increases. For figure 24, it shows a steady straight accuracy line for both training and validation sets. So, overfitting and underfitting is not an issue here. Now, hyperparameter will be used to figure out which alpha has the best score.

Two parameters are used for hyperparameter tuning. One is the hidden layer size and another one is the alpha. For alpha, 1e-06, 1e-05, 0.0001, 0.001 is set as a range and for hidden layer size, (5,5), (5,10), (10,5), (10,10) is used. After the tuning, the optimal value of dataset 1 is alpha with 0.001, and hidden layer with (10, 5). For dataset 2, alpha with 1e-05 and hidden layer of (10,10) was optimal.

Training time was also calculated. Training time for dataset 1 is 4.462 seconds and training time for dataset 2 is 9.559.

	Predict [0]	Predict [1]
True [0]	137	6
True [1]	10	75

Figure 25: Confusion matrix of dataset 1

There are 137 True Positive, 6 False Negative, 10 False Positive, 75 True Negative. Better result for class 1 but not for class 2 compare to the result before the hyperparameter tuning.

	Predict [0]	Predict [1]
True [0]	168	48
True [1]	68	177

Figure 26: Confusion matrix of dataset 2

For second dataset, there are 168 True Positive, 48 False Negative, 68 False Positive, 177 True Negative. After hyperparameter tuning, the result was not good compare to the previous results. The result might be because there are two parameters used for hyperparameter variables (hidden layer and alpha) and it had to find the optimal result using both parameters.

	precision	recall	F1-score	Support
0	0.93	0.96	0.94	143
1	0.93	0.88	0.90	85
Accuracy			0.93	228
Macro avg	0.93	0.92	0.92	228
Weighted avg	0.93	0.93	0.93	228

Figure 27: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.71	0.78	0.74	216
1	0.79	0.72	0.75	245
Accuracy			0.75	461
Macro avg	0.75	0.75	0.75	461
Weighted avg	0.75	0.75	0.75	461

Figure 28: Classification report of dataset 2

Figure 27 and 28 shows the classification report after the hyperparameter tuning. There was not that much of a difference in terms of accuracy. It was the same. (0.93 for dataset 1 and 0.75 for the second dataset.) Compare to decision tree model, accuracy of dataset 2 increased a lot (around 12 percent).

Cross validation score also increased from 0.921 +/- 0.044 to 0.930 +/- 0.021 for the first dataset and 0.730 +/- 0.031 to 0.725 +/- 0.028 for the second dataset (less standard deviation).

Support Vector Machine:

In svm classifier, linear kernel is used. Different kernels were used such as linear and rbf (In the code section, both linear and rbf is implemented). but linear kernel has better scores, so for this report, linear kernel will be used. Random state of 1 is used to see the consistent graph.

	Predict [0]	Predict [1]
True [0]	137	6
True [1]	7	78

Figure 29: Confusion matrix of dataset 1

Confusion matrix is used to see the result of prediction. There are 137 True Positive, 6 False Negative, 7 False Positive, 78 True Negative. The prediction of the model is good.

	Predict [0]	Predict [1]
True [0]	191	25
True [1]	91	154

Figure 30: Confusion matrix of dataset 2

For second dataset, there are 191 True Positive, 25 False Negative, 91 False Positive, 154 True Negative. The prediction is pretty good for the first class but not for the second class.

	precision	recall	F1-score	Support
0	0.95	0.96	0.95	143
1	0.93	0.92	0.92	85
Accuracy			0.94	228
Macro avg	0.94	0.94	0.94	228
Weighted avg	0.94	0.94	0.94	228

Figure 31: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.68	0.88	0.77	216
1	0.86	0.63	0.73	245
Accuracy			0.75	461
Macro avg	0.77	0.76	0.75	461
Weighted avg	0.77	0.75	0.75	461

Figure 32: Classification report of dataset 2

Figure 31 and 32 is a classification report for dataset 1 and 2. Dataset 1 has accuracy of 0.94 with F-1 score of 0.95 for the first class and 0.92 for the second class. Like the neural network, classification report of svm also have high scores. Dataset 2 has a result with accuracy of 0.75 and f1-score 0.77 for first class and 0.72 for second class. Dataset 2 scores for svm is the highest so far (compare to decision tree and neural network).

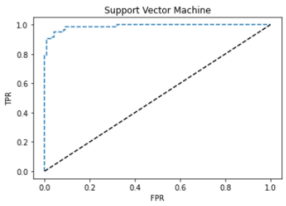

Figure 33: ROC curve for dataset 1

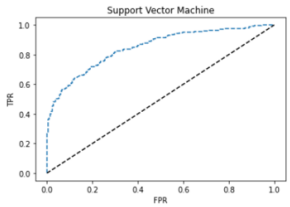

Figure 34: ROC curve for dataset 2

Figure 33 and 34 is a ROC curve before hyperparameter tuning. Figure 33 shows excellent TPR and FPR which is above 0.9 which we can say the svm model has good sensitivity and specificity for this dataset. For figure 6, it shows that the dataset 2 has fair TPR and FPR with svm.

For validation, 5-fold cross validation was used for both datasets. First dataset has cross validation accuracy of 0.933 +/- 0.019 and second dataset have cross validation accuracy of 0.730 +/- 0.012.

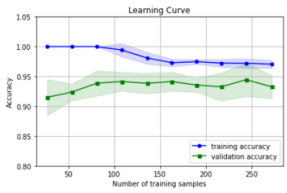

Figure 35: Learning curve for dataset 1

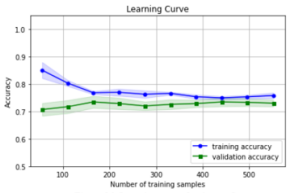

Figure 36: Learning curve for dataset 2

Figure 35 and 36 is a graph of accuracy based on increasing volumn of sample sizes. Figure 35 shows training accuracy scores going down but validation accuracy going up. It is a overfitting and therefore, has high variance. There should be more samples to train. It is happening because current testing set is not enough to well educate the validation set. For figure 36, it also suffers from underfitting. Training and validation accuracy is getting closer but their accuracy score in general is not excellent. So the model has high bias (underfitting).

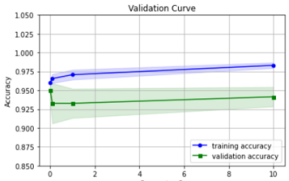

Figure 37: Validation curve for dataset 1

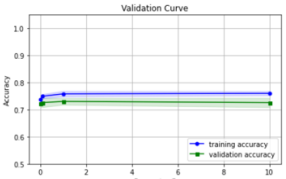

Figure 38: Validation curve for dataset 2

Figure 37 shows the overfitting problem. Training accuracy keeps on going up in a steady rate. Test accuracy keeps increasing like training rate. For figure 38, it shows a steady straight accuracy line for both training and validation sets. So, the model is not applicable for both overfitting and underfitting problems. Now, hyperparameter will be used to figure out which parameter C has the best score.

Parameter C is used for hyperparameter tuning. For parameter C, 0.01, 0.1, 1.0, 10 is set as a range. After the tuning, the optimal result of dataset 1 is C value with 0.01. For dataset 2, C value with 1.0 is optimal.

Training time was also calculated. Training time for dataset 1 is 11.359 seconds and training time for dataset 2 is 68.891 seconds.

	Predict [0]	Predict [1]
True [0]	136	7
True [1]	8	77

Figure 39: Confusion matrix of dataset 1

There are 136 True Positive, 7 False Negative, 8 False Positive, 77 True Negative. There are few fault results but still predicted most of the samples.

	Predict [0]	Predict [1]
True [0]	191	25
True [1]	91	154

Figure 40: Confusion matrix of dataset 2

For second dataset, there are 191 True Positive, 25 False Negative, 91 False Positive, 154 True Negative. After hyperparameter tuning, the result was the same with the result before hyperparameter tuning.

	precision	recall	F1-score	Support
0	0.94	0.95	0.95	143
1	0.92	0.91	0.91	85
Accuracy			0.93	228
Macro avg	0.93	0.93	0.93	228
Weighted avg	0.93	0.93	0.93	228

Figure 41: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.68	0.88	0.77	216
1	0.86	0.63	0.73	245
Accuracy			0.75	461
Macro avg	0.77	0.76	0.75	461
Weighted avg	0.77	0.75	0.75	461

Figure 42: Classification report of dataset 2

Figure 41 and 42 shows the classification report after the hyperparameter tuning. There was not that much of a difference in terms of accuracy. It was almost the same. (0.94 for dataset 1 and 0.75 for the second dataset.)

Cross validation score also increased from 0.933 +/- 0.019 to 0.950 +/- 0.018 for the first dataset and 0.730 +/- 0.012 to 0.730 +/- 0.012 for the second dataset.

The result of kernel rbf had final hyperparameter tuned cross validation score of 0.921 +/- 0.022 for the first dataset and 0.722 +/- 0.041 for the second dataset. Compare to the linear kernel, cross validation score did not increase that much. But the fun part was the learning curve and the validation curve. It had good curves where training and validation sets accuracy mostly matched. In other words, the graphs did not have neither overfitting nor underfitting problems. It is interesting because linear kernel had better accuracy score but the rbf kernel had the better graphs for both learning and validation curves.

Adaboost:

Decision tree was initially used to set up adaptive boosting. Criterion 'gini' was used with max depth of 1 for the decision tree. Adaboost gathers weak classifiers and produce a strong classifier. Random state of 1 is used to see the consistent graph.

	Predict [0]	Predict [1]
True [0]	138	5
True [1]	9	76

Figure 56: Confusion matrix of dataset 1

Confusion matrix is used to see the result of prediction. There are 138 True Positive, 5 False Negative, 9 False Positive, 76 True Negative. The prediction of the model is in good stage.

	Predict [0]	Predict [1]
True [0]	148	68
True [1]	92	153

Figure 57: Confusion matrix of dataset 2

For second dataset, there are 148 True Positive, 68 False Negative, 92 False Positive, 153 True Negative. Predict result was not successful compare to what is expected.

	precision	recall	F1-score	Support
0	0.94	0.97	0.95	143
1	0.94	0.89	0.92	85
Accuracy			0.94	228
Macro avg	0.94	0.93	0.93	228
Weighted avg	0.94	0.94	0.94	228

Figure 58: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.62	0.69	0.65	216
1	0.69	0.62	0.66	245
Accuracy			0.65	461
Macro avg	0.65	0.65	0.65	461
Weighted avg	0.66	0.65	0.65	461

Figure 59: Classification report of dataset 2

Figure 58 and 59 is a classification report for dataset 1 and 2. Dataset 1 has accuracy of 0.94 with F-1 score of 0.95 for the first class and 0.92 for the second class. Dataset 2 has a result with accuracy of 0.65 and f1-score 0.65 for first class and 0.66 for second class. By just looking at the scores, it shows that both classes have low f1 scores and accuracy scores.

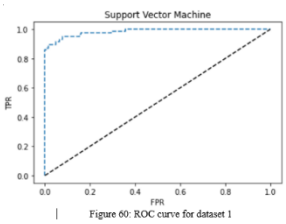

Figure 60: ROC curve for dataset 1

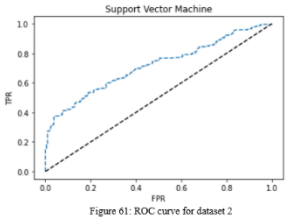

Figure 61: ROC curve for dataset 2

Figure 33 and 34 is a ROC curve before hyperparameter tuning. Figure 60 shows excellent TPR and FPR which is above 0.9 which we can say the adaboost model has good sensitivity and specificity for this dataset 1. For figure 6, it shows that the dataset 2 has fair TPR and FPR with adaboost model.

For validation, 5-fold cross validation was used for both datasets. First dataset has cross validation accuracy of 0.959 +/- 0.024 and second dataset have cross validation accuracy of 0.626 +/- 0.031.

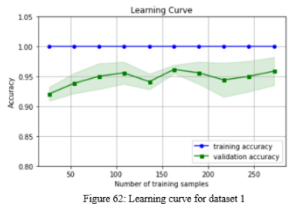

Figure 62: Learning curve for dataset 1

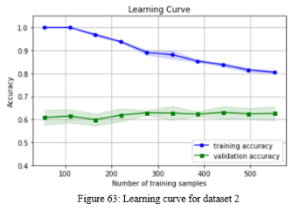

Figure 63: Learning curve for dataset 2

Figure 62 and 63 is a graph of accuracy based on increasing volumn of sample sizes. Figure 62 shows training accuracy scores at 1.00 for the whole time. For validation accuracy, it wennt up but not really close to the training scores. Therefore, it is suffering from overfitting (high variance). For figure 63, training score starts from 1.00 but decreases until 0.8. Validation accuracy goes up but does not grow significantly. The model is suffering from high bias (underfitting).

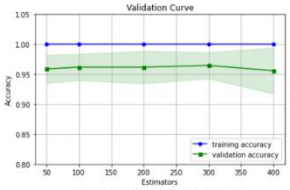

Figure 64: Validation curve for dataset 1

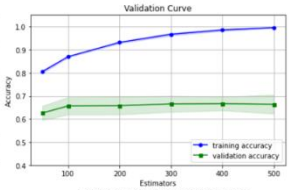

Figure 65: Validation curve for dataset 2

Figure 64 shows the overfitting problem. Training accuracy is steady at 1.0. Test accuracy also is steady but went down as sample number increases. For figure 65, training accuracy goes up and reach 1.0. validation accuracy also goes up but stops after 100 samples. The model is suffering from overfitting since the validation accuracy cannot match with the training accuracy. Now, hyperparameter will be used to figure out which estimators has the best score.

N estimators is used for hyperparameter tuning. For estimators, 50, 100, 200, 300, 400, 500 is set as a range. After the tuning, the optimal result of dataset 1 is n estimators value with 300. For dataset 2, n estimators value with 400 was optimal.

Training time was also calculated. Training time for dataset 1 was 2.725 seconds and training time for dataset 2 was 2.950 seconds.

	Predict [0]	Predict [1]
True [0]	142	1
True [1]	9	76

Figure 66: Confusion matrix of dataset 1

There are 142 True Positive, 1 False Negative, 9 False Positive, 76 True Negative. Adaboost has a best result for class 1 so far (just 1 prediction off).

	Predict [0]	Predict [1]
True [0]	139	77
True [1]	78	167

Figure 67: Confusion matrix of dataset 2

For second dataset, there are 139 True Positive, 77 False Negative, 78 False Positive, 167 True Negative.

	precision	recall	F1-score	Support
0	0.94	0.99	0.97	143
1	0.99	0.89	0.94	85
Accuracy			0.96	228
Macro avg	0.96	0.94	0.95	228
Weighted avg	0.96	0.96	0.96	228

Figure 68: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.64	0.64	0.64	216
1	0.68	0.68	0.68	245
Accuracy			0.66	461
Macro avg	0.66	0.66	0.66	461
Weighted avg	0.66	0.66	0.66	461

Figure 69: Classification report of dataset 2

Figure 68 and 69 shows the classification report after the hyperparameter tuning. There was not that much of a difference in terms of accuracy. There was an increase. Accuracy score went up from 0.94 to 0.96 for dataset 1 and 0.65 to 0.66 for dataset 2.

Cross validation score also increased from 0.959 +/- 0.024 to 0.965 +/- 0.022 for the first dataset and 0.626 +/- 0.031 to 0.667 +/- 0.030 for the second dataset.

KNN:

In KNN classifier, n neighbor value of 5 is used. 5 is used since it is the default. Also, minkowski distance is also used for the metric. Random state of 1 is used to see the consistent graph.

	Predict [0]	Predict [1]
True [0]	137	6
True [1]	10	75

Figure 70: Confusion matrix of dataset 1

Confusion matrix is used to see the result of prediction. There are 137 True Positive, 6 False Negative, 10 False Positive, 75 True Negative.

	Predict [0]	Predict [1]
True [0]	144	72
True [1]	93	152

Figure 71: Confusion matrix of dataset 2

For second dataset, there are 144 True Positive, 72 False Negative, 93 False Positive, 152 True Negative. The prediction was not good since more than 30% was wrong.

	precision	recall	F1-score	Support
0	0.93	0.96	0.94	143
1	0.93	0.88	0.90	85
Accuracy			0.93	228
Macro avg	0.93	0.92	0.92	228
Weighted avg	0.93	0.93	0.93	228

Figure 72: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.61	0.67	0.64	216
1	0.68	0.62	0.65	245
Accuracy			0.64	461
Macro avg	0.64	0.64	0.64	461
Weighted avg	0.65	0.64	0.64	461

Figure 73: Classification report of dataset 2

Figure 72 and 73 is a classification report for dataset 1 and 2. Dataset 1 has accuracy of 0.93 with F-1 score of 0.94 for the first class and 0.90 for the second class. Dataset 2 has a result with accuracy of 0.64 and f1-score 0.64 for first class and 0.65 for second class.

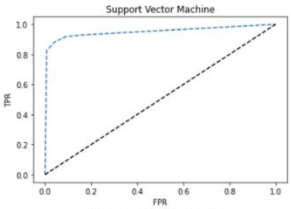

Figure 74: ROC curve for dataset 1

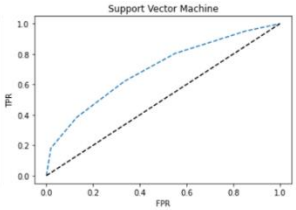

Figure 75: ROC curve for dataset 2

Figure 74 and 75 is a ROC curve before hyperparameter tuning. Figure 74 shows excellent TPR which is above 0.9 which we can say the svm model has good sensitivity and specificity for this dataset. For figure 75, it sees fair but lot of them are lacking.

For validation, 5-fold cross validation was used for both datasets. First dataset has cross validation accuracy of 0.912 +/- 0.035 and second dataset have cross validation accuracy of 0.636 +/- 0.032.

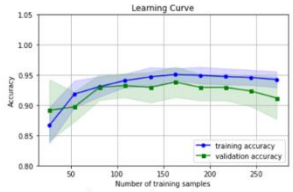

Figure 76: Learning curve for dataset 1

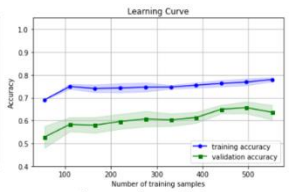

Figure 77: Learning curve for dataset 2

Figure 76 and 77 is a graph of accuracy based on increasing volumn of sample sizes. For figure 76, training accuracy goes up and slowly goes down after 150 sample is trained. Validation accuracy also shows a similar pattern but the gap becomes larger as sample sizes increases. Therefore, it is suffering from overfitting issue. Figure 77, the gap between trainning accuracy and validation accuracy gets thinner as sample sizes increases but accuracy is around 0.6~0.7. So it suffers from the underfitting issue.

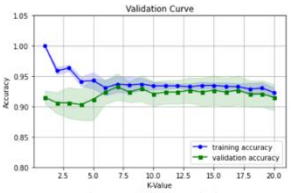

Figure 78: Validation curve for dataset 1

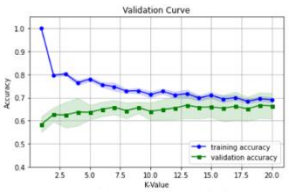

Figure 79: Validation curve for dataset 2

Figure 78, training score decreases as time passes. Validation accuracy goes up and follow the pattern of training score as sample size increases. So, it is a successful learning and thus, not suffering from either overfitting or underfitting problem. For figure 79, training score decreases

and reaches 0.7 accuracy scores. Validation accuracy increases and start to follow the pattern of training accuracy, but accuracy is still too low which the model is suffering from the underfitting issue. (high bias) Now, hyperparameter will be used to figure out which n neighbor has the best score.

N neighbor is used for hyperparameter tuning. For parameter n neighbor, 1 to 20 is set as a range. After the tuning, the result of dataset 1 is n neighbor value with 7. For dataset 2, n neighbor value with 13 was optimal.

Training time was also calculated. Training time for dataset 1 was 1.138 seconds and training time for dataset 2 was 1.284 seconds.

	Predict [0]	Predict [1]
True [0]	137	6
True [1]	10	75

Figure 80: Confusion matrix of dataset 1

There are 137 True Positive, 6 False Negative, 10 False Positive, 75 True Negative. There are few incorrect results but still classified most of the inputs.

	Predict [0]	Predict [1]
True [0]	144	72
True [1]	93	152

Figure 81: Confusion matrix of dataset 2

For second dataset, there are 144 True Positive, 72 False Negative, 93 False Positive, 152 True Negative.

	precision	recall	F1-score	Support
0	0.93	0.96	0.94	143
1	0.93	0.88	0.90	85
Accuracy			0.93	228
Macro avg	0.93	0.92	0.92	228
Weighted avg	0.93	0.93	0.93	228

Figure 82: Classification report of dataset 1

	precision	recall	F1-score	Support
0	0.61	0.67	0.64	216
1	0.68	0.62	0.65	245
Accuracy			0.64	461
Macro avg	0.64	0.64	0.64	461
Weighted avg	0.65	0.64	0.64	461

Figure 83: Classification report of dataset 2

Figure 82 and 83 shows the classification report after the hyperparameter tuning. There was not that much of a difference in terms of accuracy. It was almost the same. (0.93 for dataset 1 and 0.64 for the second dataset.)

Cross validation score also increased from 0.912 +/- 0.035 to 0.933 +/- 0.022 for the first dataset and 0.636 +/- 0.032 to 0.667 +/- 0.057 for the second dataset.

Comparison of 5 machine learning algorithms:

Algorithms	Training Time (s) Dataset 1	Training Time (s) Dataset 2
Decision Tree	0.122	0.994
Neural Network	4.642	9.559
SVM	11.359	68.891
Adaboost	2.725	2.950
KNN	1.138	1.284

First to compare is the training time of 5 different algorithms. It looks like decision tree was the fastest for training followed by KNN and Adaboost. SVM took much more compare to the other algorithms. It might be because SVM could not avoid caching the values therefore it kept on

recomputing the values, Now, let us look at the cross-validation scores after hyperparameter tuning for each algorithm.

Algorithms	CV Accuracy Dataset 1	CV Accuracy Dataset 2
Decision Tree	0.927	0.632
Neural Network	0.930	0.725
SVM	0.950	0.730
Adaboost	0.965	0.667
KNN	0.933	0.667

For dataset 1, Adaboost had the highest cross-validation score followed by SVM and KNN. For dataset 2, SVM had the highest score followed by Neural network.

One of the interesting facts I found out is that, algorithms that spent more time training had the better cross-validation scores at the end. SVM had the highest score overall (dataset 1 and 2) and followed by the Neural network, adaboost, KNN and decision tree.

References:

Dataset Citations:

Antal, Balint, and Andras Hajdu. "Diabetic Retinopathy Debrecen Data Set Data Set." *UCI Machine Learning Repository*, 2014, archive.ics.uci.edu/ml/datasets/Diabetic+Retinopathy+Debrecen+Data+Set.

Wolberg, William H. "Breast Cancer Wisconsin (Diagnostic) Data Set." *UCI Machine Learning Repository*, 1995, archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29.

Library Citations (for the code):

"1. Supervised Learning¶." *Scikit*, 2020, scikit-learn.org/stable/supervised_learning.html.