Assignment 1:
Supervised Learning

Vipul Koti
vkoti7@gatech.edu

1 ABSTRACT

This assignment seeks to understand the predictive and computational characteristics of five supervised learning algorithms (Decision Trees, Neural Networks, Boosting, Support Vector Machines, and K-Nearest neighbors). These algorithms are implemented and analyzed on two Datasets, sourced from Kaggle (Both Dataset exists on Kaggle UCI Machine Learning Repository). First, we did Exploratory Data analysis, preprocessing, encoding, scaling, and splitting of the data. Then we analyze 5 classifiers individually. Later on, we compared 5 algorithms performances. We used Scikit-learn Python library [1] for the implementation of all 5 Machine Learning models on two non-trivial datasets.

2 DATASETS AND PREPROCESSING

2.1 Introduction to Datasets and Exploratory Data Analysis

- **Heart Failure Prediction Dataset [2]** This dataset was created by combining 5 heart datasets over 11 common features. Cleveland: 303, Hungarian: 294, Switzerland: 123, Long Beach VA: 200, Stalog (Heart) Data Set: 270, Total: 1190 Duplicated: 272 Final dataset: 918 observations. All the 5 datasets originally sourced from UCI Machine Learning Repository.There are 918 rows 12 columns out of which 5 are continuous and outcome is binary.Age, sex, fasting blood sugar, Exercise Angina and old peak are positively correlated to heart disease. The class distribution is as follows, 55.3% positive, 44.7% negative.(Almost Balanced). This dataset is referred to henceforth as dataset 1.
- **Wine Quality Dataset [3]** This dataset is related to red variants of the Portuguese "Vinho Verde" wine, This dataset is available on Kaggle UCI Machine Learning Repository.There are 1143 rows 13 columns and outcome is multiclass. Alcohol content, sulfate, citric acid are positively correlated to wine quality. The initial class distribution is as follows, 5 (42.5%) , 6(40.2%), 7(12.0%), 4(3.2%), 8(1.5%) and 3(0.6%). We removed Duplicates and reduced the output to three classes, 5(44.2%), 6(41.8%) and 7(14.0%). As we can see, class 5 and 6 dominates the dataset (Un-Balanced). This dataset is referred to henceforth as dataset 2.
- **How the datasets are interesting and different ?** The two datasets are different and interesting, as the output class has different properties. Dataset 1 has binary class output it is balanced. On the other hand, Dataset 2 is unbalanced and has multiclass output, it is a more complex classification problem than dataset 1. Dataset 1 is a combination of 5 different heart datasets, which makes it interesting even though it's balanced and binary. In dataset 1, false negative are most important, so we will use recall as evaluation metric, whereas in dataset 2 as it is multiclass, we will use f1_micro score as our evaluation mertic to get overall accuracy of the classifier.

2.2 Pre-Processing of Datasets

Dataset 1, Dataset 2 has no missing/null values data, but Dataset 2 we had duplicate rows which we deleted. On both datasets, we applied hist plot and pair plot to understand the distribution then, normalization(StandardScaler) is used to bring the features between 0 and 1 as most of the features are following Gaussian distribution. Later, we used box-and-whisker plot's to handle outliers, after this, in dataset 1 categorical values are encoded using Label encoder. Then the datasets are divided into two stratified splits, where 70% of them are in training/validation and 30% of them are kept in testing data. We used stratified splits so that our results are not biased to one class.

3 COMMON FLOW FOLLOWED FOR ALL 5 MACHINE LEARNING MODEL ANALYSIS

First, we ran the ML model with default hyperparameter values to check baseline performance. Then, We did hyperparameter Tuning and analyzed the effects (overfit/underfit) of hyperparameter's on the model performance using validation curve, after the basic understanding on variation and bias of model on the dataset, we did Grid Search Cross Validation (Referred to henceforth as GSCV), as telescopic search to values near to best from validation curve, to find the best hyperparameters set. Using these best hyperparameters set, we captured the best model train and test clock time, performance metrics and created a Learning Curve to understand training progress, predict model performance and improvement over time. In Next sections, we will discuss each model separately.

4 DECISION TREE (DT)

For Decision tree, we are using Gini Index as the function to measure the quality of a split. We tuned max-depth and Minimal Cost-Complexity Pruning hyperparameters for model performance and improvement. When the max depth is small, the model is usually non-complex, it over-generalized/underfit and as we increase the max depth, the model is complex, it learns the training data specifics, and it overfits. In Cost Complexity Pruning, we reduced the number of leaf nodes, and we generalize the model more as we increase pruning. post our analysis on validation curve, we used GSCV to get the best parameter set as (ccp_alpha=0.00714 max_depth=5).

Classification report of the Model:

	precision	recall	f1-score	support
0	0.88	0.80	0.84	123
1	0.85	0.91	0.88	153
accuracy			0.86	276
macro avg	0.86	0.86	0.86	276
weighted avg	0.86	0.86	0.86	276

Classification report of the Model:

	precision	recall	f1-score	support
5	0.82	0.62	0.71	130
6	0.55	0.72	0.62	123
7	0.34	0.27	0.30	41
accuracy			0.62	294
macro avg	0.57	0.54	0.54	294
weighted avg	0.64	0.62	0.62	294

Figure 1—DT, Classification report of Model for Dataset 1 and Dataset 2

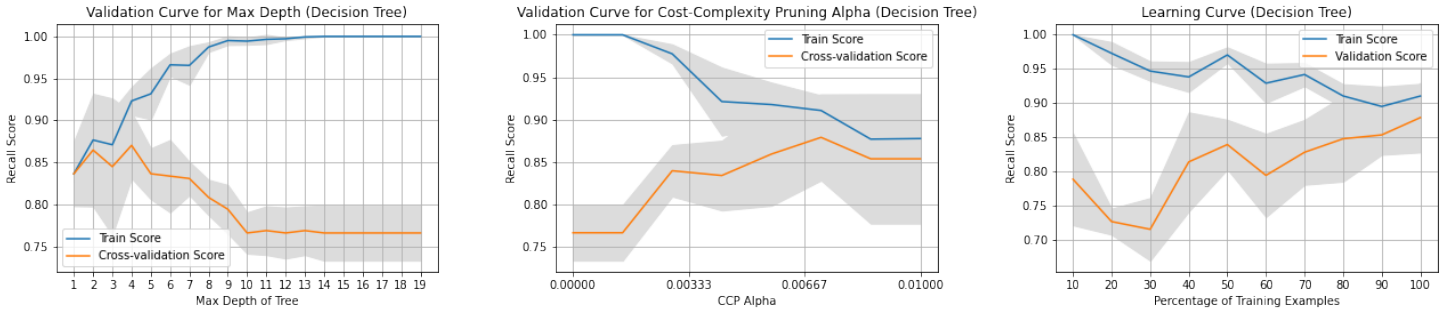

Figure 2—DT, Validation Curves Learning Curve for Dataset 1

4.1 Decision Tree for Dataset 1- Heart Failure Prediction Dataset

From the validation curves (Figure 2), we can see dataset 1, recall score is 80+ even on depth 1 (non-complex model), which means dataset is easily separable using this model, and we see from the graph, the best value of Max Depth is 4, above 4 the validation error falls and training error increases which is high variance, signaling Overfitting (As model complexity increases). For cost-Complexity Pruning, we can see high variance at 0, small value as no pruning means complex model and as the pruning increases the validation recall score increases and train score decreases. But above 0.0071 we see both training and validation score decreases, signaling high bias, suggesting the model is not robust enough. From the Learning Curve with the best parameter model, we can see the training and validation score increases with the amount of training samples, which suggests more data will improve the performance of this model.

4.2 Decision Tree for Dataset 2- Wine Quality Dataset

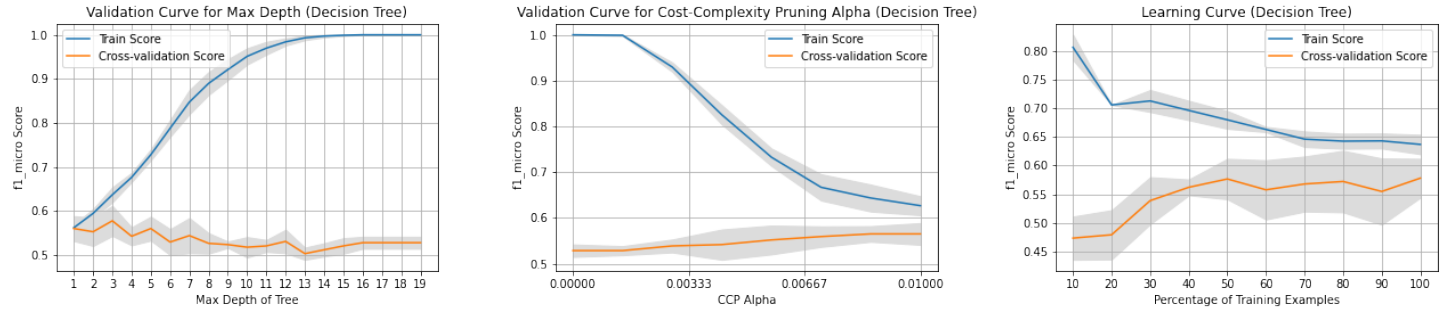

Figure 3—DT, Validation Curves Learning Curve for Dataset 2

Dataset 2 is a complex dataset with multi-class output, From both the validation curves (Figure 3). We can clearly see the data is complex for a model like Decision Tree. There is no much improvement on validation score even after tuning the hyperparameters. The best hyperparameter combination is ccp_alpha=0.00714 max_depth=5 with f1score 0.552. From the learning curve for the Best DT learner for Dataset 2, we can see high variance and bias over all. Even though if we provide more data, the accuracy will be below 0.6(approx)

as we see for training data in the curve which convince us to say that the dataset is too complex for this model. We can add new features and remove in-significant features for better model performance.

5 NEURAL NETWORK (NN)

For Neural Network, we choose hidden layer size and Learning Rate Init as our two hyperparameters. As we increase the hidden layers, we make the model more complex and prone to over-fitting. Also, learning rate init is one of the most important hyperparameter for NN as small learning rate (step size) can result in a failure to train, and large step size can result is suboptimal results.

Classification report of the Model:

	precision	recall	f1-score	support
0	0.88	0.80	0.84	123
1	0.85	0.92	0.88	153
accuracy			0.87	276
macro avg	0.87	0.86	0.86	276
weighted avg	0.87	0.87	0.87	276

Classification report of the Model:

	precision	recall	f1-score	support
5	0.79	0.73	0.76	130
6	0.60	0.63	0.61	123
7	0.45	0.46	0.46	41
accuracy			0.65	294
macro avg	0.61	0.61	0.61	294
weighted avg	0.66	0.65	0.66	294

Figure 4—NN, Classification report of Model for Dataset 1 and Dataset 2

5.1 Neural Network for Dataset 1- Heart Failure Prediction Dataset

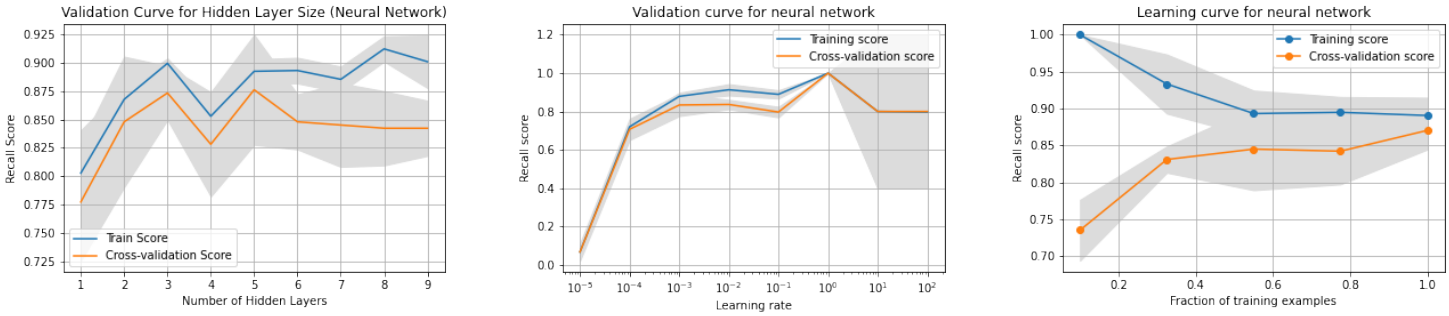

Figure 5—NN, Validation Curves Learning Curve for Dataset 1

from Figure 5(i) we can see as we increase the number of hidden layers, the training score gets better and for validation score we see two peaks at 3 and 5, after 5 the training score improves but cross-validation score drops suggesting high variance and overfitting. Learning rate init from Figure 5(ii) we see the recall score improves for training and validation if the step size is near to 0.1. Above 0.1 both drops suggesting high bias, suboptimal model at high learning rate. From the Learning Curve, we can see the training score after 50% of training data is consistent and validation score is getting better and closer to training score. Because the training score is consistent around 0.90 if we want the model to perform even better, we need to add new features.

5.2 Neural Network for Dataset 1- Loss Curve

From the loss curve's on Dataset 1, we can see very high learning rate as the dataset is simpler in comparison to dataset 2. We get best performance at learning rate init as .1, As an experiment we also checked graph at 0.001 which showed gradual loss and MSE graph similar to Dataset 2. But thr performance is better at learning rate init 0.1 with 3 hidden layer, compared to learning rate 0.001 with hidden layer 5. And from the score curve, we can see comparable score performance on training and validation, suggesting little variance.

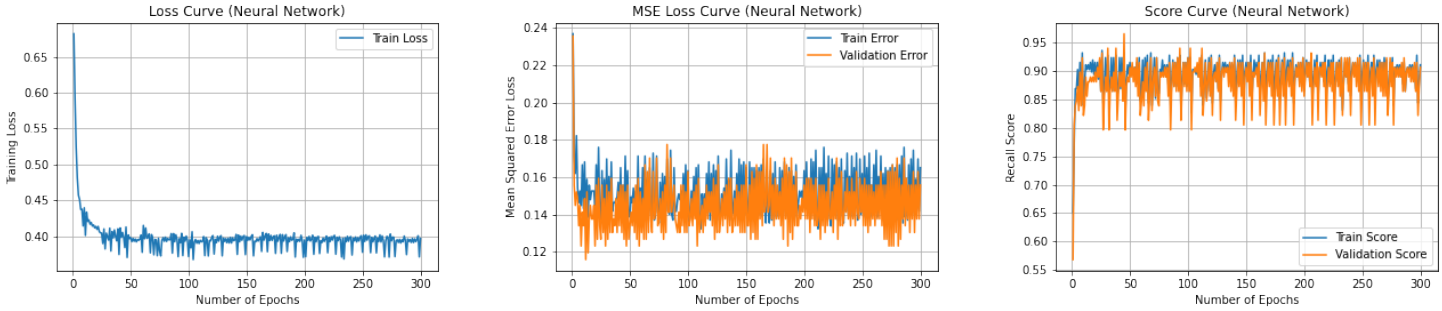

Figure 6—NN, Loss Curves Score Curve for Dataset 1

5.3 Neural Network for Dataset 2- Wine Quality Dataset

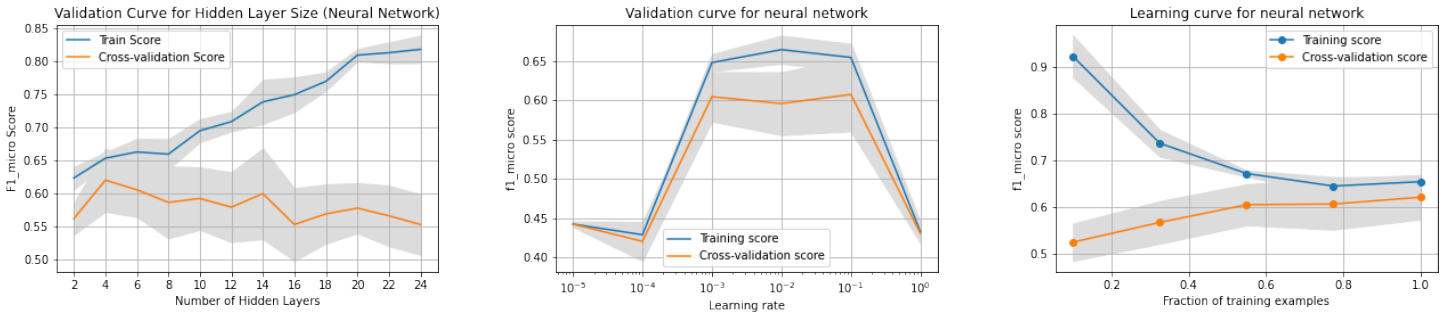

Figure 7—NN, Validation Curves Learning Curve for Dataset 2

From the validation curve for hidden layer, we can see the training score of 80+ as we increase the hidden layers, the model is learning the specifics of the training data. Best value is 4 as greater than 4 the validation score drops. Learning rate init seems to be best at 0.001 which is a smaller learning size as compare to dataset 1, as we know dataset 2 is complex, so it is expected. Best parameter set is 'hidden_layer_sizes': 4, 'learning_rate_init': 0.001. From the learning curve we can see there is a sharp drop in the training score as we increase the data, which tells us about the noise and randomness in our data which is makes it complex. Even though the validation score is consistently increasing with data, but it is constrained by upper limit of training score as 0.65.

5.4 Neural Network for Dataset 2- Loss Curve

From the loss curves on dataset 2, we can say the learning rate is low 0.001 in comparison to the dataset 1 i.e 0.1 as the dataset 2 is complex. 0.001 gives better score than 0.1. And from the score curve, we can see strong overfitting and high variance on the training data for this model.

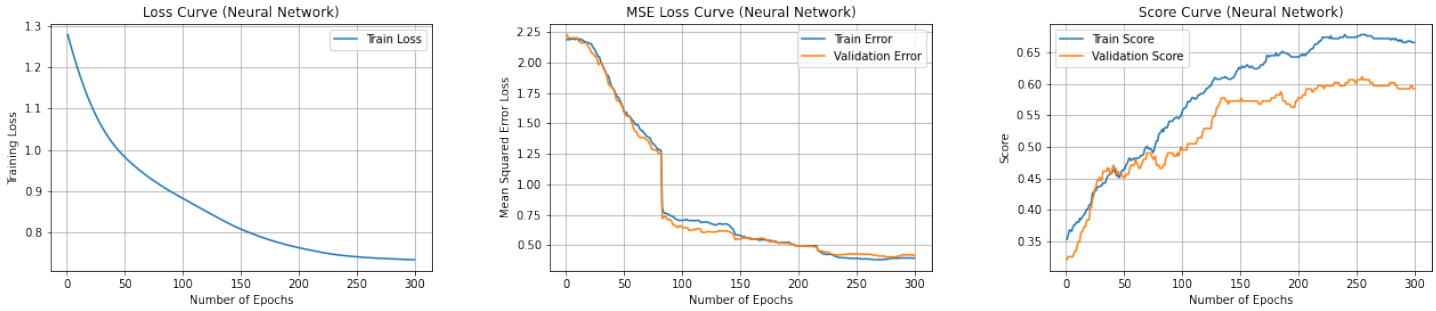

Figure 8—NN, Loss Curves Score Curve for Dataset 2

6 BOOSTING

For Ada Boosting classifier, we choose n_estimators and learning rate as our two hyperparameters. As Boosting is an ensemble learning method, with n_estimators we choose the number of weak learners. If we increase the number of weak learners, we make a complex model and eventually overfit. Increase in learning rate, increase the contribution of each classifier, a low learning rate result in longer training period and very high learning rate results in suboptimal model.

Classification report of the Model:

	precision	recall	f1-score	support
0	0.87	0.76	0.81	123
1	0.82	0.91	0.86	153
accuracy			0.84	276
macro avg	0.85	0.83	0.84	276
weighted avg	0.84	0.84	0.84	276

Classification report of the Model:

	precision	recall	f1-score	support
5	0.80	0.66	0.72	130
6	0.55	0.69	0.61	123
7	0.47	0.37	0.41	41
accuracy			0.63	294
macro avg	0.61	0.57	0.58	294
weighted avg	0.65	0.63	0.63	294

Figure 9—ADA Boosting, Classification report of Model for Dataset 1 and Dataset 2

6.1 ADA Boosting for Dataset 1- Heart Failure Prediction Dataset

From the Validation curves we can see, the increase in number of estimators doesn't impact drastically on the cross validation score, but it improves the training score drastically. We do see a dip in CV score above 600. We can say above 600 we can see overfitting as the training score increases and validation score dips. For Learning rate, we see the training and CV score both are consistent for very low learning rate, and it peaks around 0.01 and then significantly dips for larger learning rates. Best parameters are 'learning_rate': 0.01, 'n_estimators': 201. from the learning curves initially, we see a dip in both training and CV score when

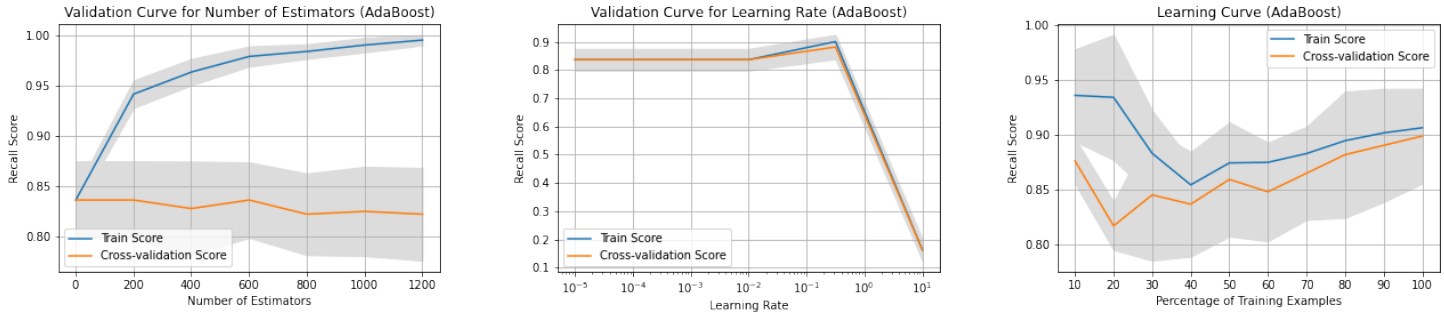

Figure 10—ADA Boosting, Validation Curves Learning Curve for Dataset 1

we increase the training example, which might be because of an unnecessary complex model for that small dataset. But as we see after 20% of training example the cross-validation score is consistently increasing. More data will be helpful for this model to perform even better.

6.2 ADA Boosting for Dataset 2- Wine Quality Dataset

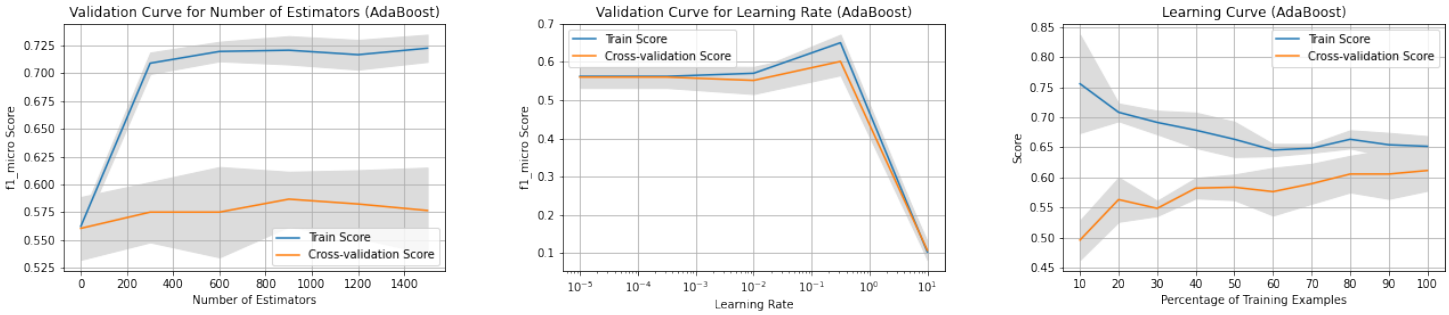

Figure 11—ADA Boosting, Validation Curves Learning Curve for Dataset 2

For dataset2, from learning curves, we can see the number of estimators above around 900 causes the model to overfit as the CV score dips above 900. For learning rate, from the validation curve we can see a sharp fall starting between 0.01 and 1, suggesting non-optimal model above that step size range. Best parameters are 'learning_rate': 0.031623, 'n_estimators': 900 From the learning curve, we can see sharp drops with increase in training examples initially suggesting high noise, but later on after 60% of training data we can see training score is consistent at 0.65 and validation score is increasing. The model will perform better than current on cross-validation score if we increase the data. But as we have upper bound because of training score, we can add more features to increase overall performance.

7 K-NEAREST NEIGHBOR (KNN)

For KNN, we used nearest neighbor count and power parameter for the Minkowski metric. n_neighbors=1 means the sample is using itself as reference, which is overfitting case. So increasing the n_neighors count to generalize the model and reduce variance. Power metric p=1 is equivalent to using manahattan_distance, and

p=2 as euliddean_distance. For arbitrary p, minikowski distance (l_p) is used. Usually the choice is between 1 and 2.

Classification report of the Model:

	precision	recall	f1-score	support
0	0.88	0.82	0.85	123
1	0.86	0.91	0.89	153
accuracy			0.87	276
macro avg	0.87	0.86	0.87	276
weighted avg	0.87	0.87	0.87	276

Classification report of the Model:

	precision	recall	f1-score	support
5	0.74	0.71	0.72	130
6	0.55	0.60	0.57	123
7	0.43	0.37	0.39	41
accuracy			0.62	294
macro avg	0.57	0.56	0.56	294
weighted avg	0.62	0.62	0.62	294

Figure 12—KNN, Classification report of Model for Dataset 1 and Dataset 2

7.1 KNN for Dataset 1- Heart Failure Prediction Dataset

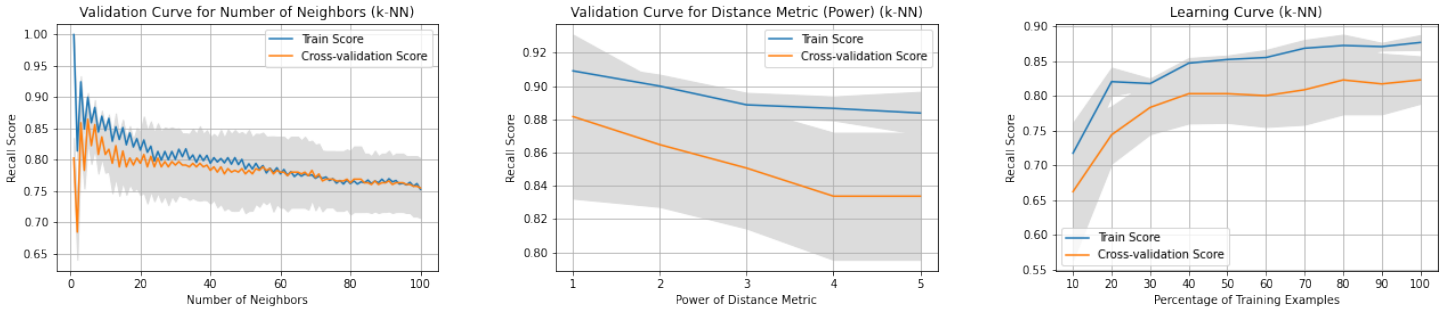

Figure 13—KNN, Validation Curves Learning Curve for Dataset 1

For dataset 1, we can see from the validation curve the CV score drops after neighbor count increases beyond 5. Below 3 it's overfitting and there is high variance, and above 5 it's highly biased and both Train and CV score drops. For Minkowski metric, power = 1 we get the best score for train and CV. Best set of parameters are 'n_neighbors': 6, 'p': 1. from the learning curve, we see a gradual increase in score as we increase the data. More data will result in better performance of the model.

7.2 KNN for Dataset 2- Wine Quality Dataset

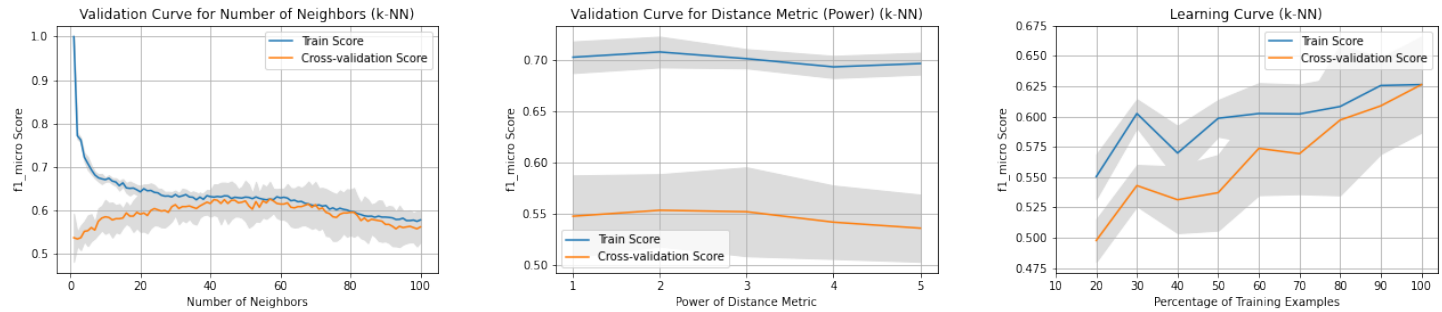

Figure 14—KNN, Validation Curves Learning Curve for Dataset 2

For Dataset 2, the score improves until the n_neigbors value is between 40 and 60. Beyond 60 it drops. In comparison to dataset 1, we know Dataset 2 is complex, and it requires more complex model for generalization. For Minkowski metric, power = 2 we get the best score for train and CV. Best set of parameters are 'n_neighbors': 57, 'p': 2. From the learning curve, we can see as the 'n_neighbors': 57 the curve starts from 20% of the data and the score gets better with the increase is training data. But we can see the training score and CV score are very close. There is no scope of improvement for the model with more data. We can add more features and remove less correlated features to increase the performance of the model.

8 SUPPORT VECTOR MACHINE (SVM)

For SVM, we used kernel, Degree, and penalty parameter C as our hyperparameters. For Dataset 1, Polynomial Kernel works better, that means the straight/bent ruler lines works better for this classification, so we use degree hyperparameter with this kernel. Whereas for Dataset 2, RBF kernel works better as it's more complex and the data needs to be transformed to higher dimensions before we classify, and we use parameter C as hyperparameter with this Kernel. C controls the trade-off between smooth decision boundary and classifying the training points correctly, Whereas the Kernel parameter decides the type of hyperplane used to separate the data.

Classification report of the Model:

	precision	recall	f1-score	support
0	0.91	0.77	0.84	123
1	0.84	0.94	0.89	153
accuracy			0.87	276
macro avg	0.88	0.86	0.86	276
weighted avg	0.87	0.87	0.86	276

Classification report of the Model:

	precision	recall	f1-score	support
5	0.75	0.77	0.76	130
6	0.63	0.62	0.62	123
7	0.47	0.46	0.47	41
accuracy			0.66	294
macro avg	0.62	0.62	0.62	294
weighted avg	0.66	0.66	0.66	294

Figure 15—SVM, Classification report of Model for Dataset 1 and Dataset 2

8.1 SVM for Dataset 1- Heart Failure Prediction Dataset

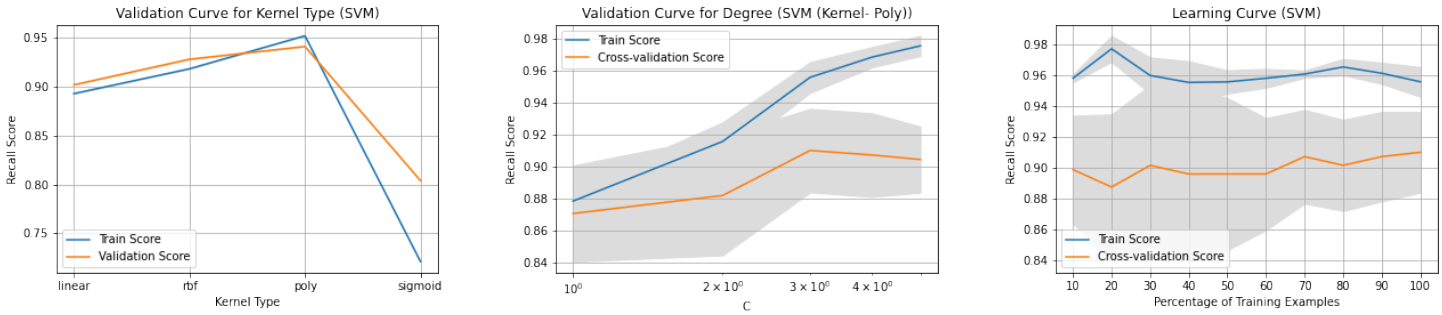

Figure 16—SVM, Validation Curves Learning Curve for Dataset 1

For Dataset 1, this dataset is not that complex and SVM with polynomial Kernel works better. So we tuned the Degree hyperparameter for the polynomial kernel, and we can see degree 3 works best, above 3 the training score increases, but the validation score falls, suggesting overfitting. Best set of parameters are 'degree': 3,

'kernel': 'poly'. From the Learning Curve, we can see the model performs good on both training and validation data, more data will be better for this model.

8.2 SVM for Dataset 2- Wine Quality Dataset

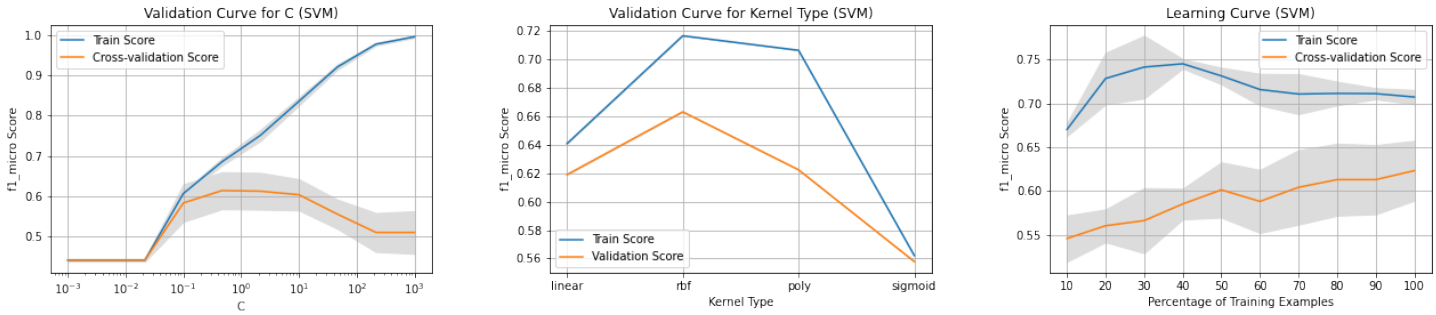

Figure 17—SVM, Validation Curves Learning Curve for Dataset 2

For Dataset 2, as it's a complex dataset, we got interesting results. For penalty parameter C, the CV score increases as we increase the value and then peaks between 0.01 and 1. Beyond 1, the train score keeps on increasing but CV score drops, suggesting high variance and overfitting. Among the kernels, rbf kernel performs a little better (66.3%) than the other (linear(61.9%), poly(62.3%), sigmoid(63.2%). Interestingly, for linear and sigmoid kernel the CV score was greater than the Training score, confirming about the randomness in the Dataset2 which is hard to separate using linear and sigmoid function hyperplane. From the learning curve we can see even though the training score is consistent below 70%. The validation score can improve if we add more data. But in order for better overall performance, we have to add more features to the dataset.

9 MODEL PERFORMANCE COMPARISON

9.1 Model Performance Comparison, Dataset 1- Heart Failure Prediction Dataset

	Classifier	precesion	recall	f1-score
0	DT	0.860000	0.860000	0.860000
1	NN	0.870000	0.870000	0.870000
2	AdaBoost	0.840000	0.840000	0.840000
3	k-NN	0.870000	0.870000	0.870000
4	SVM	0.870000	0.870000	0.860000

From the weighted scores table, we can see NN, K-NN and SVM performs equally well on Dataset 1 in comparison to DT and ADA Boost. The interesting fact about the model comparison for Dataset 1 is that even DT and K-NN perform better than Ada Boost. This is because there are less number of features and all the features seems to be significant for the prediction.

From the histograms, we can see clock time comparison for train and inference, and Recall score for each Classifier Model. As it's a Heart Disease Prediction Classification. False Negative are most important. So we plotted Recall score for the third histogram.

We can see SVM performs the best with Recall score 0.94, and it also takes the least clock time to train. So it's the best algorithm for Dataset 1. whereas Adaboost takes highest clock time to train and inference, Also,

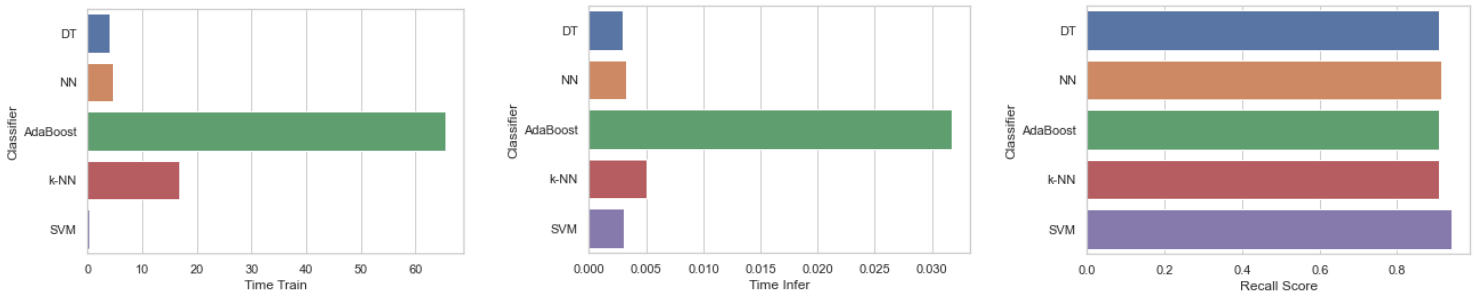

Figure 18—Model Performance Comparison Histograms for Dataset 1, Recall Scores

the recall score for Ada boost is less compare to SVM, so it is the least favored algorithm for dataset1.

9.2 Model Performance Comparison, Dataset 2- Wine Quality Dataset

	Classifier	precesion	recall	f1-score
0	DT	0.640000	0.620000	0.620000
1	NN	0.660000	0.650000	0.660000
2	AdaBoost	0.650000	0.630000	0.630000
3	k-NN	0.620000	0.620000	0.620000
4	SVM	0.660000	0.660000	0.660000

Dataset 2 is a complex dataset, and we can see from the weighted scores table. More flexible algorithms like Neural Network and SVM fits better for Dataset 2. On the Other hand K-NN did poor in comparison to other algorithms as all the features are not equally important in Dataset 2. It is not possible for KNN to distinguish more significant features.

From the histograms, we can see clock time comparison for train and inference, and Recall score for each Classifier Model. As it's a multiclass dataset. We plotted f1-score for the third histogram.

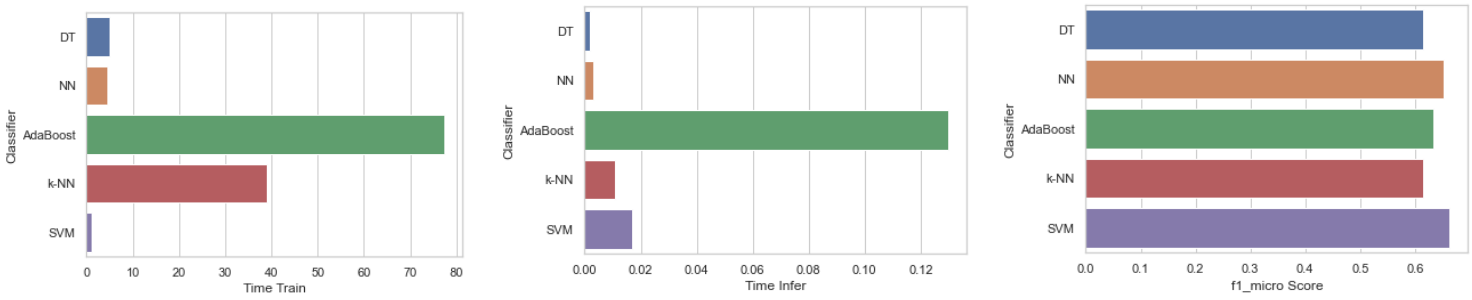

Figure 19—Model Performance Comparison Histograms for Dataset 2, $\mathrm{fi_m icroscores}$

As we know, Neural Network and SVM performs almost equally good on the dataset 2. On comparing the clock time for both the algorithms, Neural Network seems to be the best performing Algorithm on Dataset 2 as SVM take considerably more time to infer in comparison to Neural Network. This is because the hidden layer size for Neural network is small. On the other hand, as expected, ADA boost takes highest time to train and infer as it's a combination of 900 weak estimators for dataset 2.It performs slightly better than DT and KNN. This data set is complex for DT and KNN as few features are more significant to predict the quality of the wine ad the output is imbalanced.

REFERENCES

[1] Scikit-learn. (n.d.). scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. Retrieved September 24, 2022, from https://scikit-learn.org/stable/

[2] Heart failure prediction dataset. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

[3] Wine quality dataset. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/yasserh/wine-quality-dataset