Supervised Learning: A1 - CS7641

Perry Francois-Edwards (pdfe3@gatech.edu)

1 INTRODUCTION

This report will introduce two datasets that will compare their characteristics and performance using machine learning algorithms learning and validation curves.

1.1 Datasets

The two datasets chosen were a Cars Acceptability Evaluation (CAE) and a Red Wine Quality Measure (RWQM). Both initially attracted attention because of personal interests. However, these datasets were also chosen based on their unique data features. Both datasets share similarities such as multi-class, no missing data and imbalance. However, their specific differences below were hoped to provide insight into algorithm performance: (1) The RWQM has 6 target classes, while the CAE has 4; (2) The RWQM has 11 features and the CAE has 6 features; (3) The CAE has Categorical Features, while the RWQM has continuous numerical values; and lastly (4) their unique imbalances where the CAE has one target value that has the majority with roughly 70% of the target values and the RWQM has 2 of the target classes with a distribution of 40% each.

1.2 Hypothesis

My Hypothesis is that the CAE will outperform the RWQM in the Neural Networks, KNN and SVM algorithms because of its numerical precision and lower complexity, along with the distance computations. The RWQM will outperform the CAE with the Decision Tree and Ensemble algorithms because of its multi-class imbalance uniqueness. Performance will be measured by a weighted f1_score to account for the imbalance. Max clock times will be recorded from cross validation. Expect that where performance is better, clock times are better.

2 METHODS

After selecting the classification datasets, preprocessing with LabelEncoder and standardization, and completing a random 80/20 train/test split, the two experiments, Learning and Validation Curves, were produced for each dataset to compare algorithm performance based on f1_score due to data imbalance.

Learning curves used the base models, tuning, cross-validation and random state as a baseline to observe the amount of data needed to accurately perform the model. These curves have the ability to show bias, variance, overfitting and underfitting.

The Validation curves were used as tools to visualize the impacts of hyperparameters in the models over multiple iterations and to choose an optimal hyperparameter, based on a similar structure of the learning curves. Depending on the increased complexity, the graph should be able to predict bias and variance in the model and its effect on fit.

3 DECISION TREES

The scikit-learn tree, DecisionTreeClassifier, was used. 'Entropy (or Information Gain)' was used to split the data because of its benefits for imbalanced data. The pruning methods were max depth and leaf nodes.

3.1 Learning and Validation Curves

The figures below show the Learning and Validation curves in respect to the CAE and RWQM and shows the CAE performed better. The CAE converges to ~ .84 where the RWQM converges to ~.54. Both learning curves converged which shows adequate data was available for both datasets, but the RWQM's low performance shows bias and underfitting. The validation curves show similar characteristics. Both Max Depth hyperparameters start with an underfit until reaching an optimal value. The CAEs Max Depth has a low variance, but high bias until it reaches a plateau. The RWQMs Max Depth curve has slight underfit, but after the optimal point of ~3, it begins to overfit and have a large variance. The RWQMs Max Leaf Nodes optimize around ~6 and then begin to overfit as variance increases. The CAEs Max Leaf Nodes underfit for the majority of the trace.

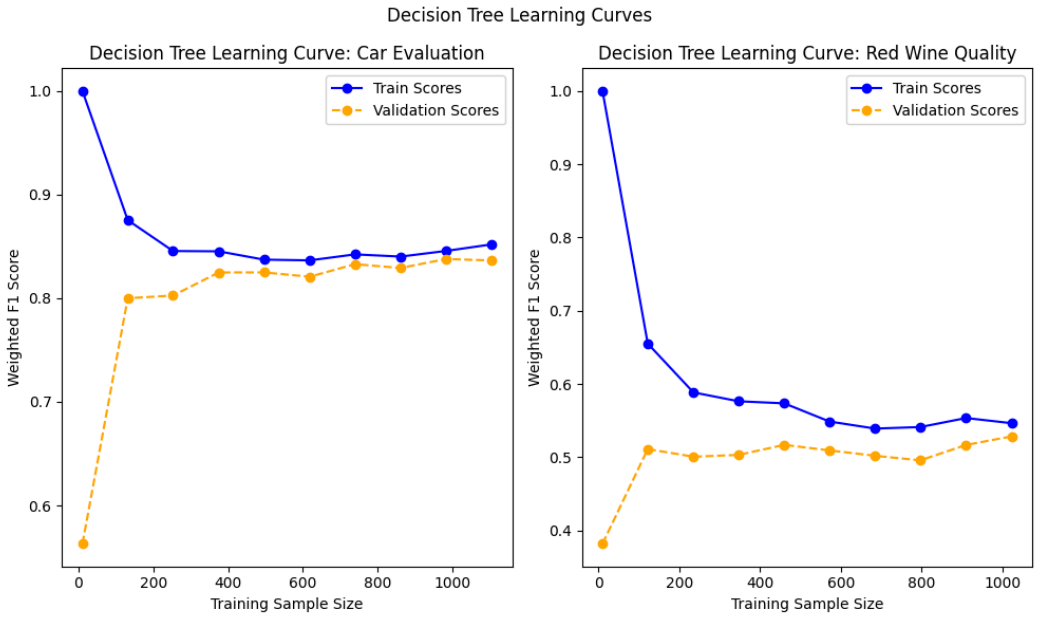

Figure 1—Decision Tree Learning Curves comparison.

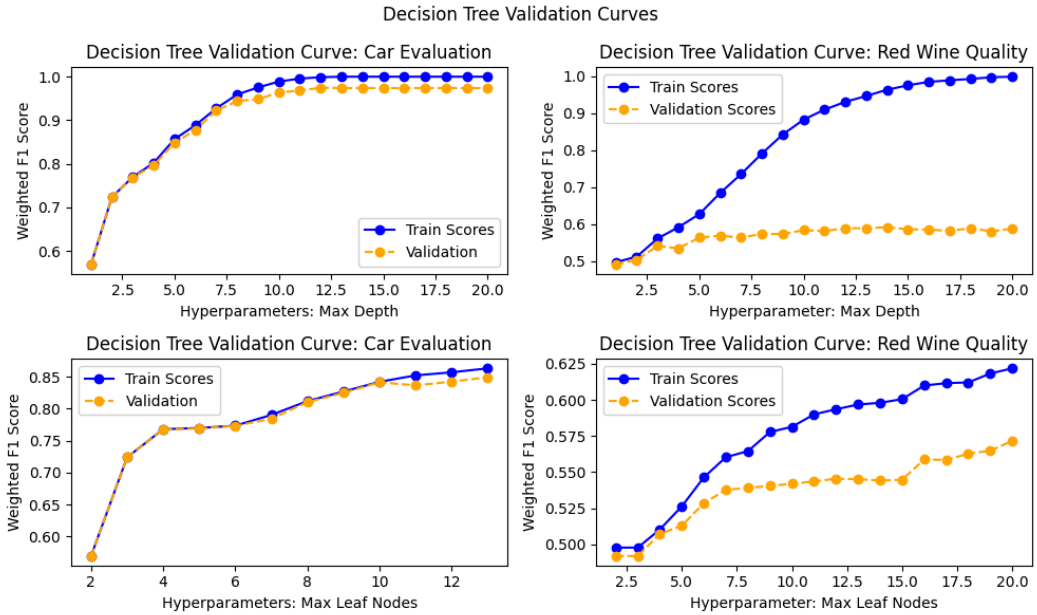

Figure 2—Decision Tree Validation Curves comparison.

4 NEURAL NETWORKS

The scikit-learn neural networks, MLPClassifier, was used. The activation was 'relu' and the solver was 'adam'.

4.1 Learning and Validation Curves

The CAE Learning Curve starts to converge over time to a high performance score, where the unforeseen data, validation scores, begin to fit the model better. Low variance occurs at the end of the sampling size, but overall the data converges as expected. The RWQM converges around ~ .6, which shows high bias with the potential to underfit the data. However, there is low variance due to minimal distance between the training and validation sets. The Validation Curves show interesting results. The RWQM shows more potential for the hyperparameters to negatively impact the model. If the RWQM increases Hidden Layer Size, the data will expectedly overfit, whereas if the alpha is increased it is expected to have high bias with the potential of underfitting. In regards to the CAE curves, the Hidden Layer Sizes start with high bias at very low values, but both begin to converge overtime to low bias and low variance, which constitutes a higher value should be used. The CAEs Alpha hyperparameter shows little variance, but a decreasing score with added complexity. The Loss Curve shows the level of certainty on the models predictions and how they compare to each other when fitted with training data. With losses greater than 1 and RWQM

losses being greater than the Cars, it confirms the poor performance. To improve performance further, the next steps would be to investigate the use of different activation conditions and layer sizes.

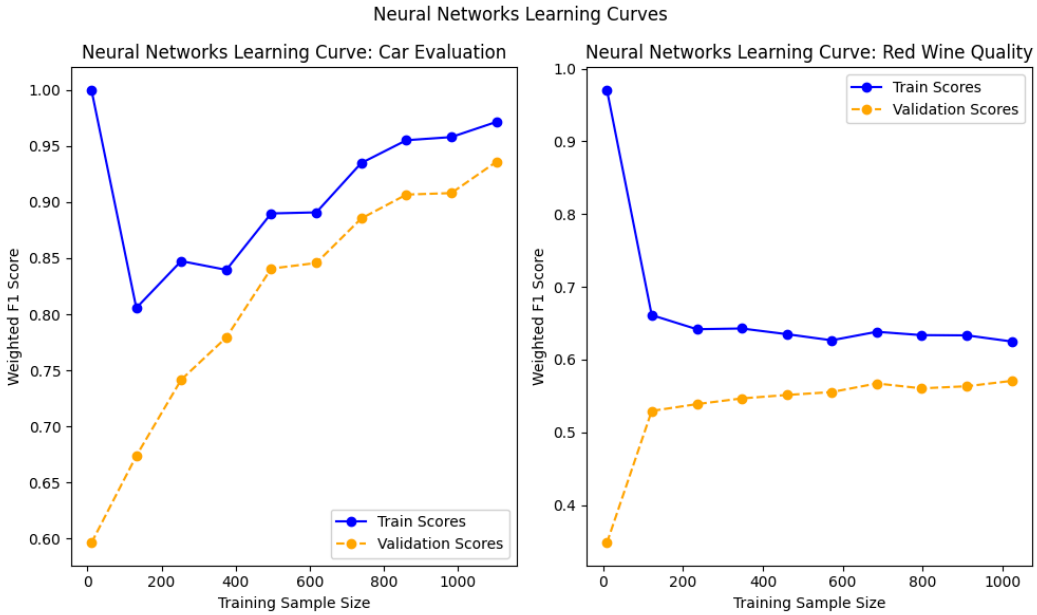

Figure 3—Neural Network Learning Curves comparison.

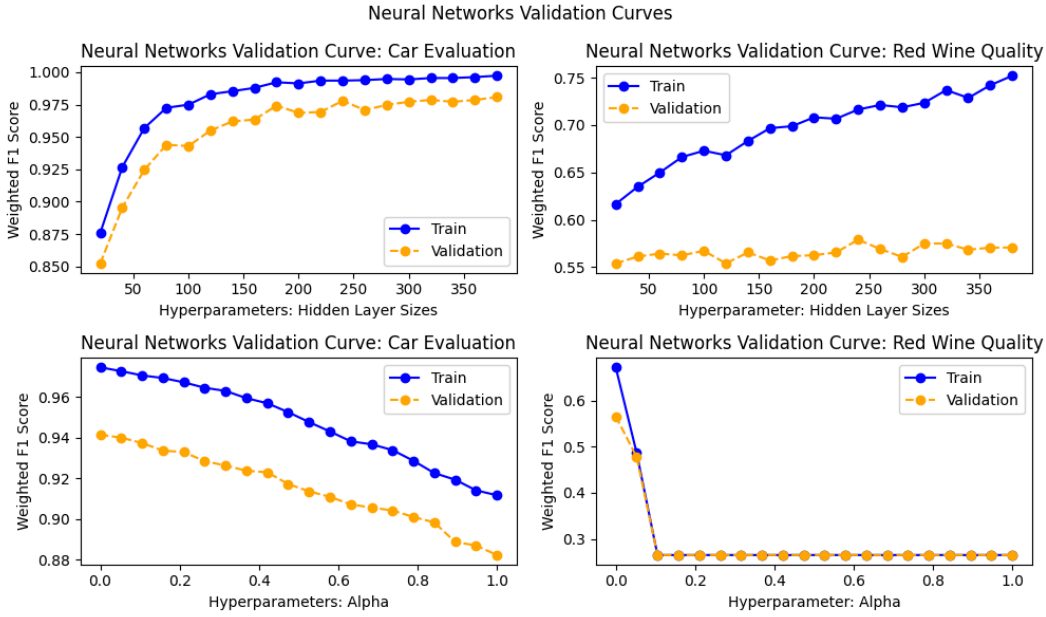

Figure 4—Neural Network Validation Curves comparison.

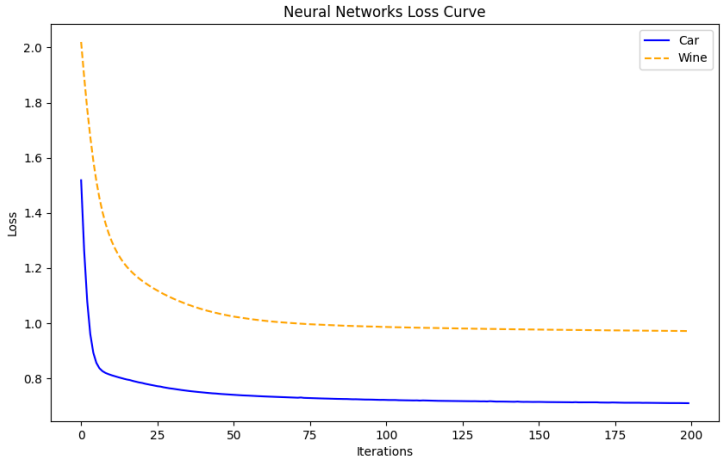

Figure 5—Neural Network Loss Curves comparison.

5 BOOSTED DECISION TREES

The scikit-learn tree, DecisionTreeClassifier, was used as the base estimator to build the scikit-learn ensemble, AdaBoostClassifier.

5.1 Learning and Validation Curves

Similar to the Decision Trees, both datasets attempted to converge on the Learning Curves, but only the CAE seems to perform well. The RWQM is trying to converge to a low value and seems to be underfitting and showing signs of high bias. Thus further noting more data or balancing may be needed for RWQM. It's interesting because it is performing worse than the Decision Tree. The Validation Curves for the estimators show different aspects. The CAE shows performance decreases after ~20 estimators, which can expect overfitting to occur after this point. The n_estimators for the RWQM continually increase but it points to overfitting because the training data is increasing at a higher rate than the validation data. It was difficult to deduce the Learning Rate hyperparameters because they both increase and plateau. For these values, it was important to choose a value that was unbiased and increasing. Overall, the RWQM hyperparameters are still underfitting the model because of the poor performance, but the promise of learning is apparent. To improve performance further, the next steps would be to investigate the use of different ensemble methods and how to be more aggressive in pruning methods. That should help the RWQM perform in a better fashion.

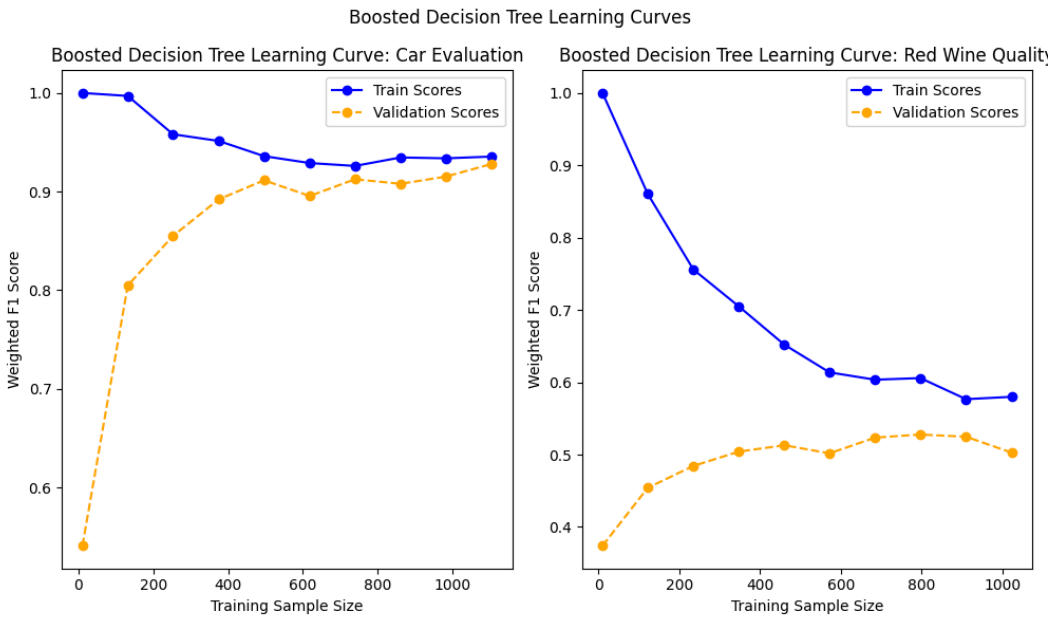

Figure 6—Boosted Decision Tree Learning Curves comparison.

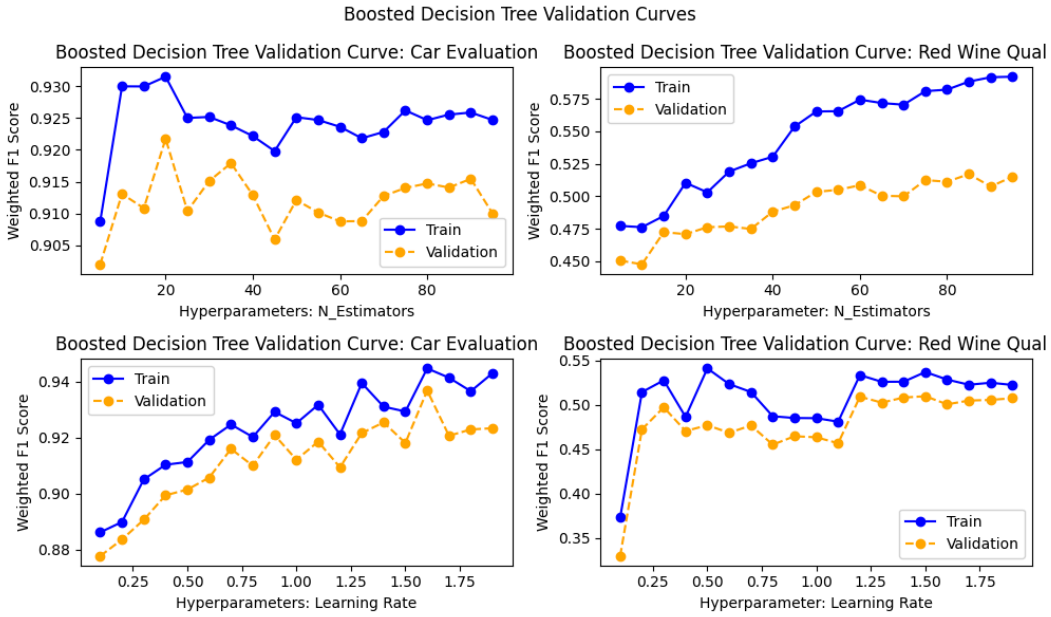

Figure 7—Boosted Decision Tree Validation Curves comparison.

6 SUPPORT VECTOR MACHINES

The scikit-learn, svm.SVC, was used to build a Support Vector Classification model. The RBF kernel was used for the learning curves.

6.1 Learning and Validation Curves

The CAE Learning curve increases and looks to perform well with low variance and low bias. For the RWQMs Learning curve, unfortunately it's low performance score, high bias, and high variance where the scores try to converge point to a model that is underfitting. Thus the need to improve complexity, data and feature addition.

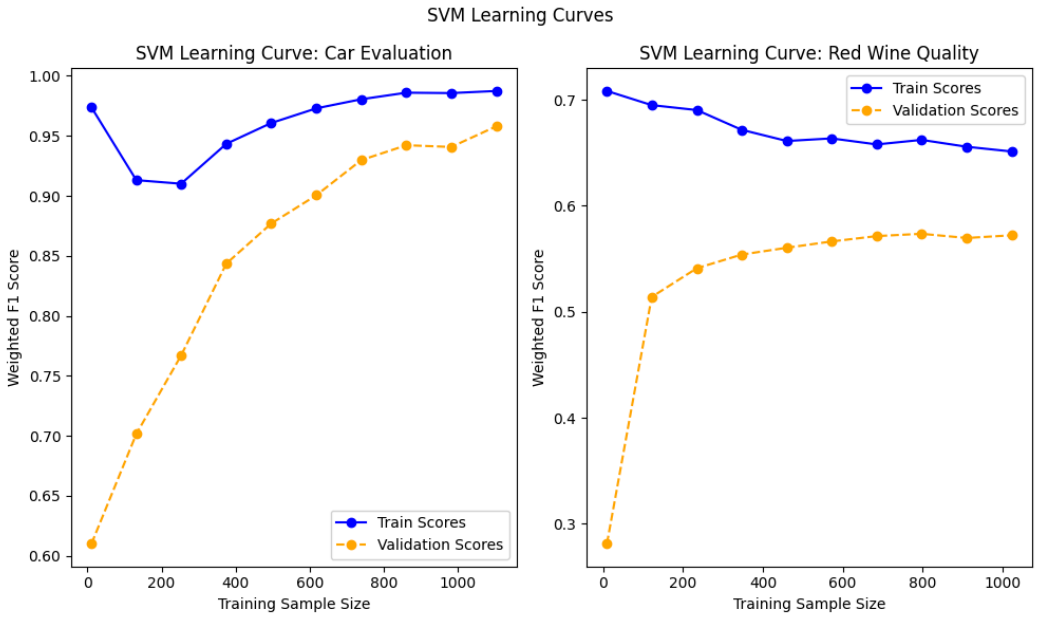

Figure 8—SVM Learning Curves comparison.

In regards to the Validation curves, there are three different kernels (RBF, Sigmoid, Poly) used to represent the differences and performance of the algorithm. The best performing algorithm was RBF, shown by its high performance in both the CAE and RWQM and its clear optimal characteristics. In the Validation curve, it shows clear optimal points for RBF. For the CAE Validation curve, the optimal point is ~.4 to prevent underfitting the model with a lower value and prevent overfitting the model with a higher value. Same goes for the RWQM dataset, but its optimal point for performance is ~ .1. Here, the performance trace is clear what values will produce optimal results. To improve performance further, the next steps would be to investigate the use of more hyperparameters on a given function. It was difficult to consistently tune and record validation curves, especially once they became time consuming. Finding an optimal C value for SVM would have been ideal and I believe it would have helped get RWQM to an acceptable f1_score.

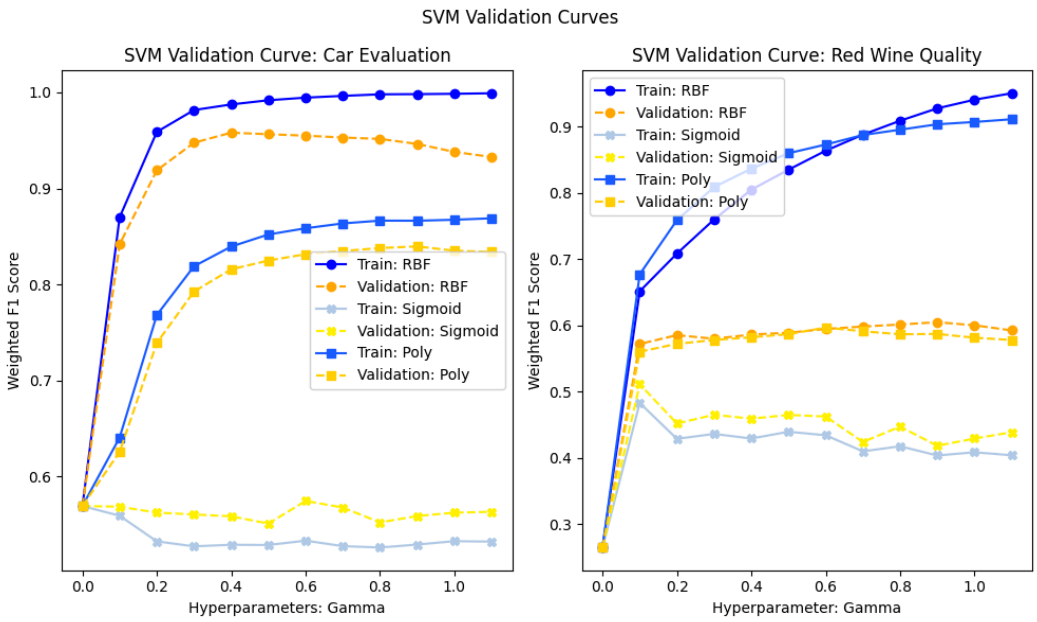

Figure 9—SVM Validation Curves comparison.

7 K-NEAREST NEIGHBORS (KNN)

The scikit-learn neighbors, KNeighborsClassifier, was used to build the model. The weights were uniform and the algorithm was default auto.

7.1 Learning and Validation Curves

Here, both Learning curves begin with high bias and underfitting. However, they both behave differently. The CAEs train and validation continually increase which reduces bias, but they do not directly converge to a performance score. That can lead to more data being necessary, but high performance will be expected with the current data available. The RWQM does not converge either, but it does have a slight decrease towards the end of the sampling size. The RWQM Learning curve has a high bias and high variance, indicating the need of more data to improve the fit for this algorithm and underfitting. In terms of Validation curves, it was difficult getting multiple hyperparameters to perform well, even Kernel count had difficulty. The CAE validation curve overfits with a low value and shows an optimal value of ~8. The RWQM Validation curve also shows signs of overfit to the left while the validation curve begins to decrease. The RWQMs optimal point seems to be ~75 kernels for consistent performance. To improve performance further, the next steps would be to look into weight distribution and different metrics to better compute distance.

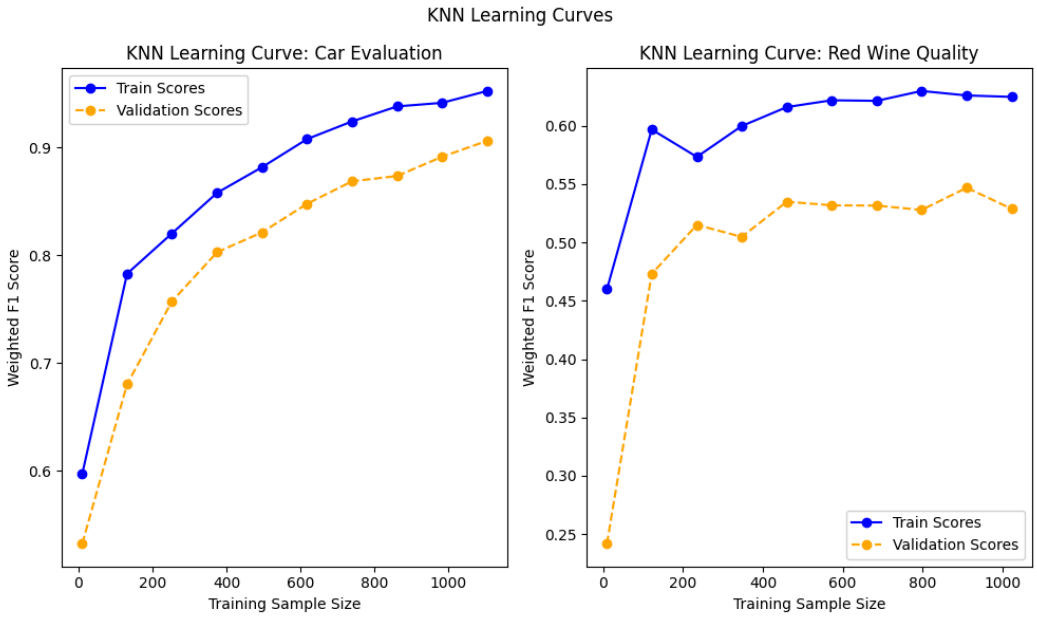

Figure 10—KNN Learning Curves comparison.

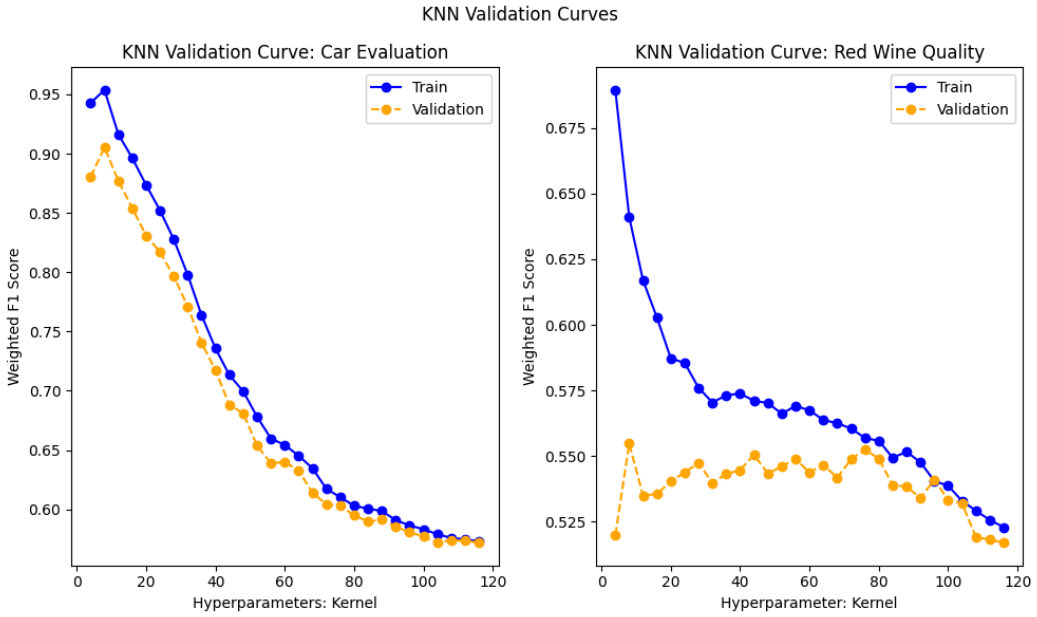

Figure 11—KNN Validation Curves comparison

8 RESULTS & CONCLUSION

In respect to each dataset, I was surprised that the CAE performed best in all algorithms whereas the RWQM models, showing high bias and low variance,

were too simple to capture the underlying patterns in the data and more data may actually be needed. In this case, I define the best performing dataset as which dataset's performance, f1_score, was higher based on the tuned hyperparameters referenced. Cross validation, standardizing and label encoding all helped improve the performance of the models in their own respective ways, but did not provide enough complexity for the RWQM to perform well. SVM performed the best in terms of differing algorithms because of its consistent performance and informative characteristics with hyperparameters and expectation of performance based on the datasets structure. However, the Neural Networks produced the highest performance metrics for both datasets. I did expect Decision Trees to perform well because of the CAE simplistic nature and it did. To improve future performance for the RWQM and CAE in each of the algorithms, I would try to find more appropriate features and increase the hyperparameters usage to observe optimization areas, along with grid search to help tune hyperparameters. It would also be beneficial to test more preprocessing methods, like balancing and normalization, to grow performance. The strengths of the study come from the use and tuning of multiple hyperparameters (even those not seen) to understand their importance and effect on the data. The limitations mostly came from time complexity and understanding of each parameter and its effect on the specific data used.

8.1 Fit Times

CAE was expected to have faster clock times based on performance, but on average it is taking longer than the RWQM for the NN and KNN algorithms.

Table 1—The max fit times taken from cross validation training.

Algorithm	Fit Time unit	CAE	RWQM
Decision Tree	*s*	~.0015	~.002-.003
Neural Networks	*s*	~.37	~.29
Boosted Decision Trees	*s*	~.03	~.15
Support Vector Machines	*s*	~.023-.025	~.034
K-Nearest Neighbors	*s*	~.002	~.001

9 REFERENCES

1. Bohanec,Marko. (1997). Car Evaluation. UCI Machine Learning Repository. https://doi.org/10.24432/C5JP48.
2. Cortez,Paulo, Cerdeira,A., Almeida,F., Matos,T., and Reis,J.. (2009). Wine Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.