Assignment 1 (Supervised Learning)

Cindy Nyoumsi
snyoumsi@gatech.edu

***Abstract*—** In this report, I will be implementing five (5) different supervised machine learning algorithms, namely Decision trees (with some form of pruning), Neural Networks, Boosting, Support Vector Machines and k-nearest neighbors on two datasets and comparing the prediction results and performance of each of these algorithms on each data set.

1 CLASSIFICATION PROBLEM DESCRIPTION

I will be exploring two classification problems outlined below;

1.1 College vs Non-College Graduates Classification Description

For the first classification problem, I will be trying to classify adults into *college graduates* vs *non-college graduates* using census demographic data.

This might be useful/interesting in a customer segmentation analysis where as a data science consultant, I want to help a college graduate career service company optimize their marketing investment by focusing only on college graduates in a large pool of potential clients.

1.1.1 *Adult Dataset Description*

- The adult dataset is made up of 14 attributes with about 49000 instances.
- Every attribute is a demographic description of a specific adult such as their sex, marital status, age, race, country of origin etc.
- For the Education attribute which I will be predicting, I had to re-label the data so as to turn it from a multi-class into a binary attribute.

1.2 Risky vs Non-Risky Lenders Classification Description

For the second classification problem I will be trying to classify adults into *risky* vs *non-risky lenders* using demographic and payment history data.

This might be useful/interesting in a customer segmentation analysis where as a data science consultant, I want to help a loan company identify how risky their lending portfolio client base is.

1.2.1 *Credit Dataset Description*

- The Credit dataset is made up of 24 attributes with 30000 instances.
- 6 of the attributes are demographic such as age, gender, marital status, while the remaining 18 attributes are monthly payment history.
- The credit default attribute was already binary so no major pre-processing was required.

2 TRAINING AND TESTING ERROR RATES

For both datasets, I followed five (5) main steps with each model.

1. Implemented a simple non-defined version of the model to train the data on.
2. Performed simple and stratified cross validation on the train data..
3. Plot a validation curve to identify best parameter values for training the data.
4. Implemented a Grid Search to identify best estimator parameter values on the train dataset.
5. Used the best estimator parameter values from steps 3 and 4 to run the model on the test data.

The table below shows a summary of the results for each model from the Decision Tree to KNN.

2.1 Adult Data Set

	Decision Trees	**Neural Networks**	**Boosting**	**SVM**	**K-NN**
Training Error Rates	0.1534	0.2300	0.1817	0.0016	0.1976
Test Error Rates	0.1748	0.2315	0.2310	0.2311	0.2429
Training Accuracy Ra	0.8466	0.7700	0.8183	0.9984	0.8024
Test Accuracy Rates	0.8252	0.7685	0.7690	0.7689	0.7571
Test Precision Rates	0.8146	0.6826	0.8233	0.6946	0.6870
Test Recall Rates	0.8252	0.7685	0.7690	0.7689	0.7571
Test F1-score Rates	0.8171	0.6740	0.7837	0.6761	0.6943

Figure 1 — Table showing different Classifier Quality Scores for the adult data set.

2.2 Credit Data Set

	Decision Trees	Neural Networks	Boosting	SVM	K-NN
Training Error Rates	0.1125	0.2436	0.1803	0.0079	0.1811
Test Error Rates	0.1910	0.2197	0.2420	0.2242	0.2288
Training Accuracy Rates	0.8875	0.7564	0.8197	0.9921	0.8189
Test Accuracy Rates	0.8090	0.7803	0.7580	0.7758	0.7712
Test Precision Rates	0.7884	0.7412	0.7833	0.7002	0.7164
Test Recall Rates	0.8090	0.7803	0.7580	0.7758	0.7711
Test F1-score Rates	0.7883	0.7426	0.7678	0.6927	0.7197

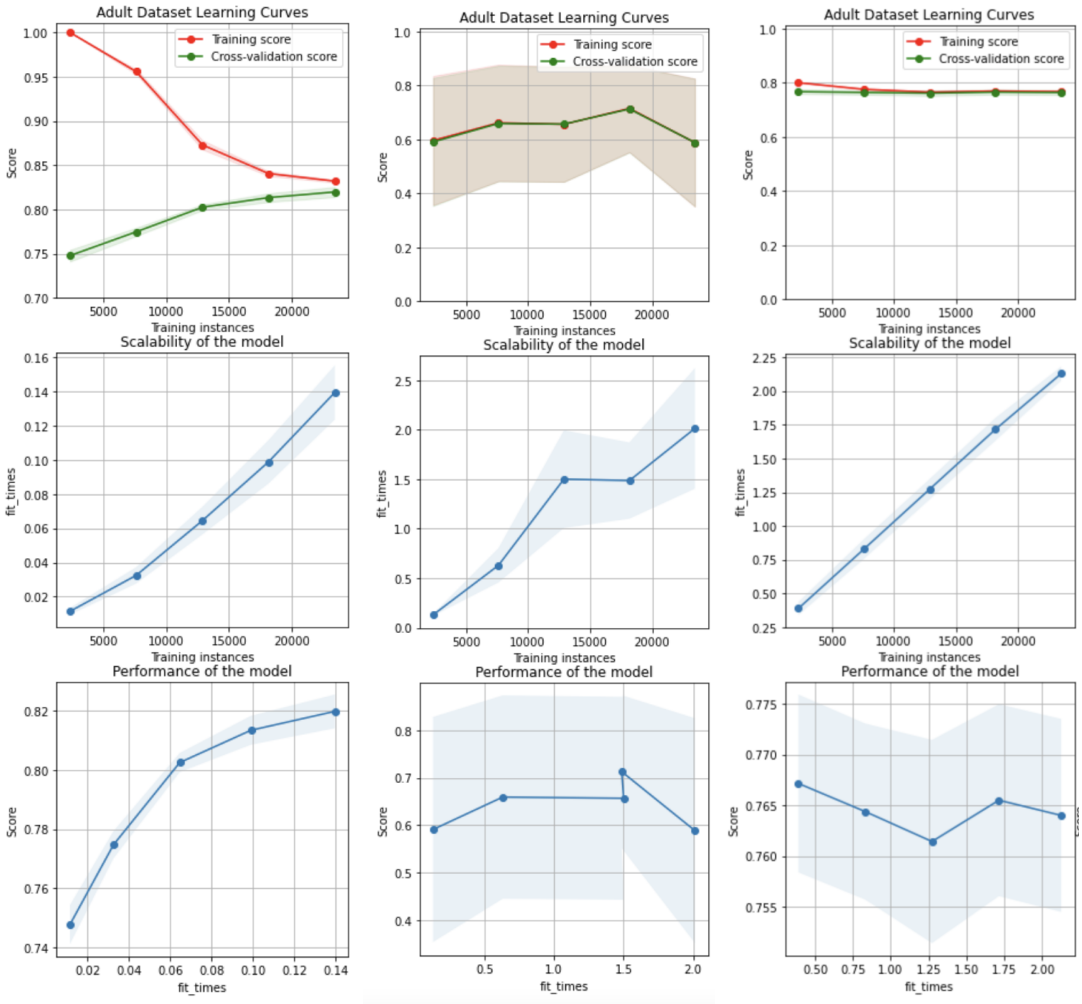
Figure 2 — Table showing different Classifier Quality Scores for the adult data set.

3 LEARNING CURVE GRAPHS

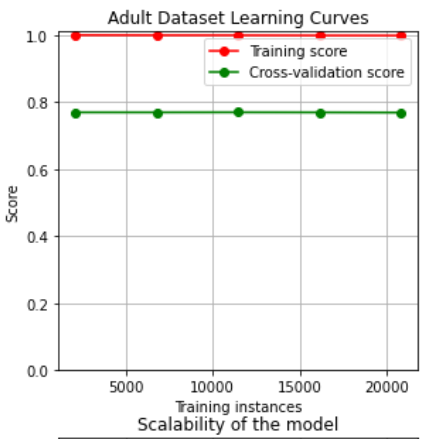
I then proceeded with using Gridsearch to find the best parameter values to use with the stratifiedKfold and plotted the learning curve results as a function of the number of training instances. I also plotted a validation curve to identify the best pruning alpha value to use for the model implementation. I will be referencing the charts below when diving deeper into the performance of the models in sections 4 and 5 of this paper.

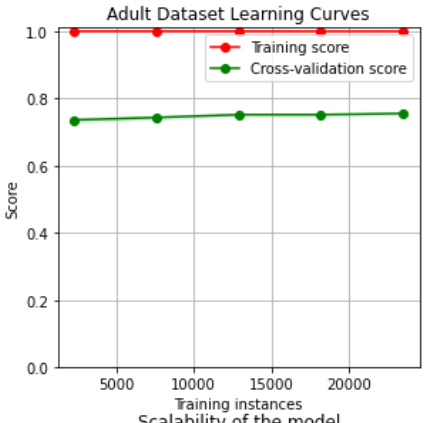
3.1 Adult Dataset Learning Curves

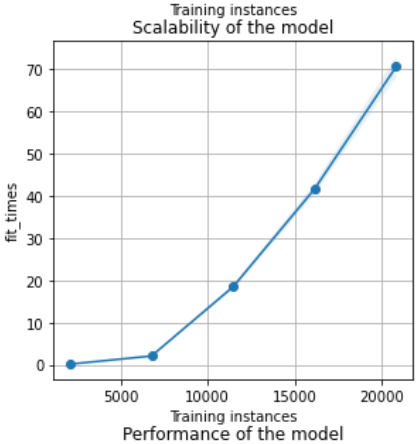
Decision Tree Neural Network BoostingEnsemble

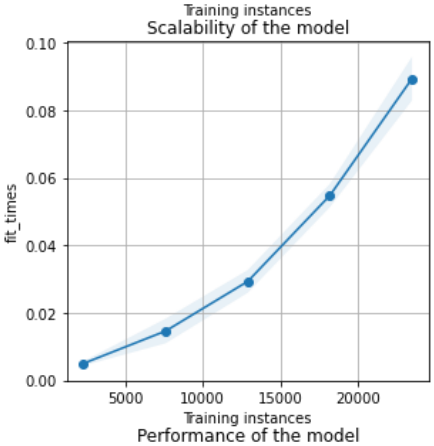


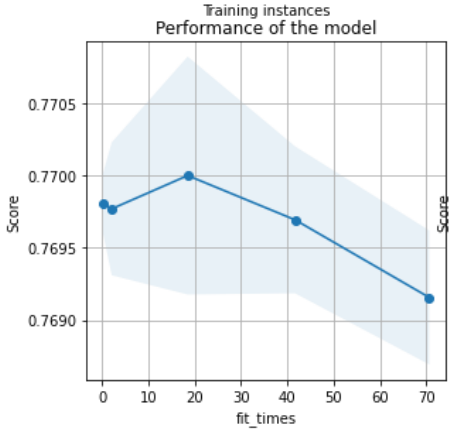
Support Vector Machine K-Nearest Neighbors

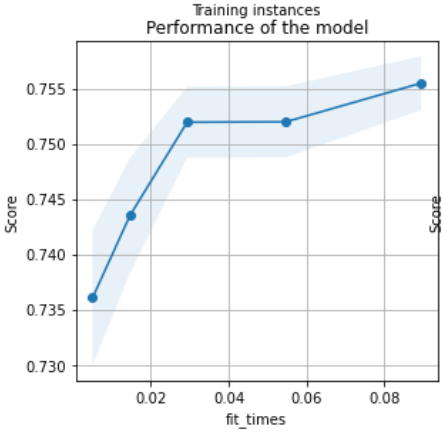

Adult Dataset Learning Curves
Training score
Cross-validation score
Score
Training instances


Adult Dataset Learning Curves
Training score
Cross-validation score
Score
Training instances


Scalability of the model
fit_times
Training instances

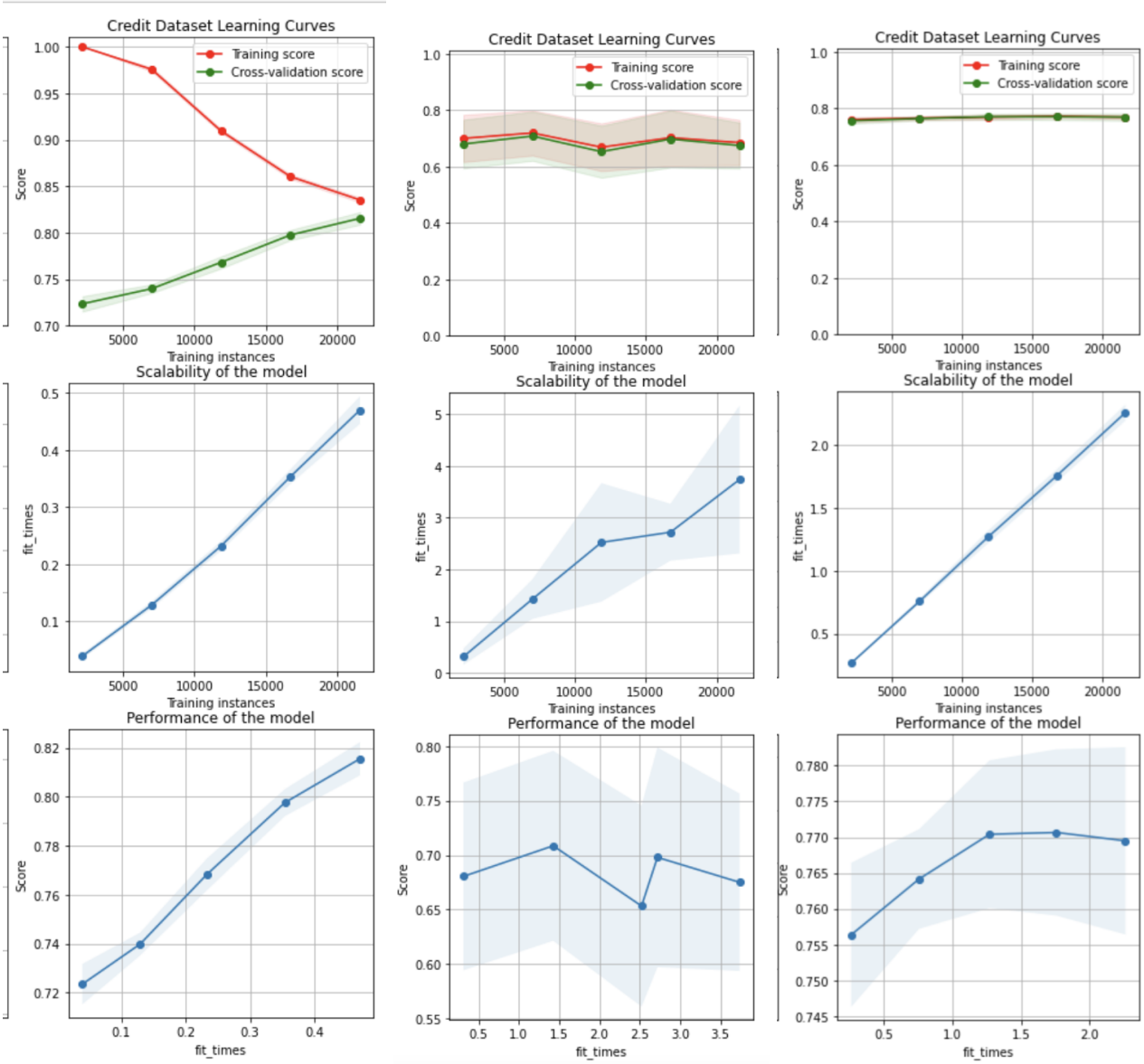

Scalability of the model
fit_times
Training instances


Performance of the model
Score
fit_times

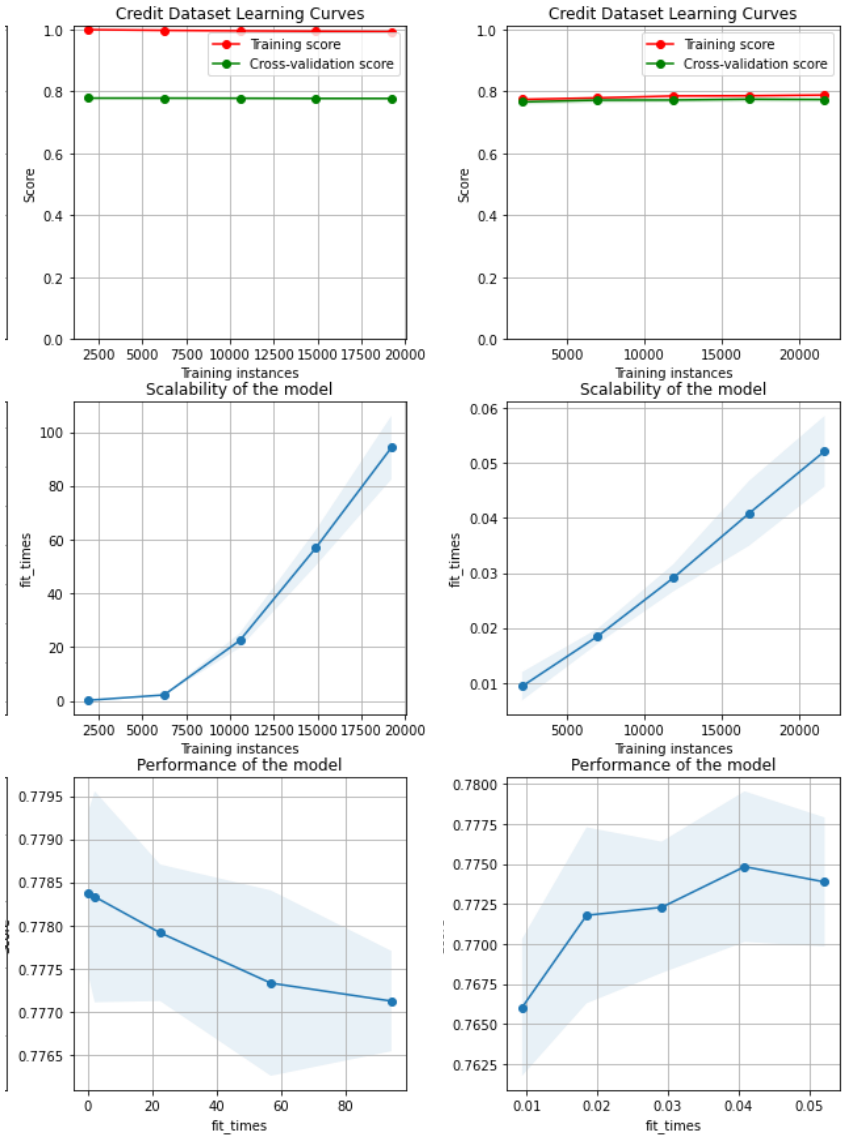

Performance of the model
Score
fit_times

3.2 Credit Dataset Learning Curves

Decision Tree Neural Network BoostingEnsemble.



Support Vector Machine K-Nearest Neighbors



4 PERFORMANCE ANALYSIS

In assessing the performance of each of these models, I will be looking at 4 main metrics in context of the classification problem and the gravity of the predictions. The

four metrics I will be looking at are Accuracy, Precision, Recall and F1-Score with respect to the type of data, the classification problem and the size of the data.

I will be referencing the charts and tables above when comparing the performance of each classification model on both datasets.

4.1 Adult Dataset Performance Analysis.

4.1.1 Best Performing model on Adult Dataset and why.

When it came to the adult dataset, the best performing model was the Decision Tree Algorithm when looking across the model's accuracy, precision, recall and F1-score on the testing data as it had the highest score for each metric among all the models and fit the testing data the best.

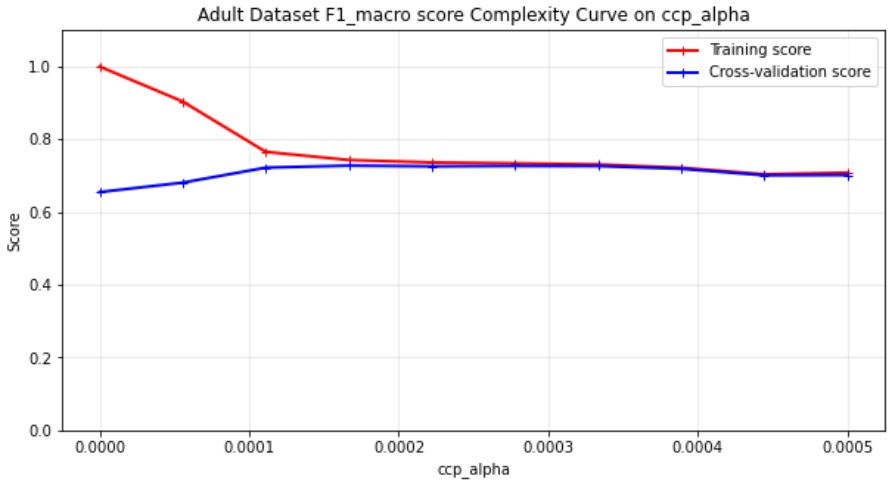
Also looking at the learning curves of each model and comparing the bias and variance, the Decision Tree also has the best performance especially as the size of the data increases, as we can see the model generalizing better as the data size increases thus overfitting less.

The Decision Tree algorithm also has the best model scalability and time as the training data increases.

I believe the Decision Tree performing so well on the data adult dataset had a lot to do with the nature of the data with most of the independent features of the adult data set being categorical which is ideal for an algorithm like a decision tree. The fact that the data size was of medium size was also an advantage.

4.1.2 Improving Performance of models.

I was able to improve the performance of the models by using a validation curve along with a gridsearch to identify the best pruning parameters such as alpha parameter for the decision tree model, hidden layers for the Neural networks, number of estimator bags for Boosting, gamma for SVM and number of K-neighbors for KNN. Focusing on just the decision tree since it is the best performing, the best optimized values for ccp_alpha was between 0.0002 and 0.00033 as shown in the chart below. I followed similar steps for the other models as well where I leveraged a validation curve, grid search and cross validation where appropriate to improve the performance of each algorithm and adjust for any variance and bias.



4.1.3 Worst Performing model on Adult Dataset and why.

While the worst performing model was the Neural Networks looking at the same metrics.

I believe the Neural Network algorithm did not perform so well on the adult dataset because of the size of the data and also because of the categorical nature of the independent features, making it a bad candidate for this type of model. Although it did not have the worst score across all the performance metrics I decided to look at, it had the worst score in most, especially when looking at the F1-score alongside the precision and recall scores.

Interestingly the Support Vector Machine and K-Nearest Neighbors had very similar performance across these metrics and both seem to have a big overfitting problem on the training data, thus doing not so well on the test data. Various efforts to improve performance by optimizing various parameters such as gamma, or number of nearest neighbors did not make a significant improvement. I believe this is also related to the inability for the data to be normalized since they were mostly categorical.

4.1.4 Did Cross Validation Help?

I also implemented the models using cross validation, both simple and stratified but it did not make a significant difference on the results for any of the models on the adult data set, in fact on some models such as the support vector machine the results were worse with cross validation. (See jupyter notebook for Cross-Validation results)

4.2 Credit Dataset Performance Analysis.

4.2.1 Best Performing model on Credit Dataset and why.

Looking at the Credit Dataset that shared similar traits to the adult dataset in terms of having independent categorical features, but also more continuous features like the payment history features which, the best performing model was still the Decision Tree algorithm when looking across the model's accuracy, precision, recall and F1-score on the testing data as it had the highest score for each metric among all the models and fit the testing data the best. Unlike with the adult data set though other than the Support Vector Machine, all the other models performed closely as well as the decision tree model.

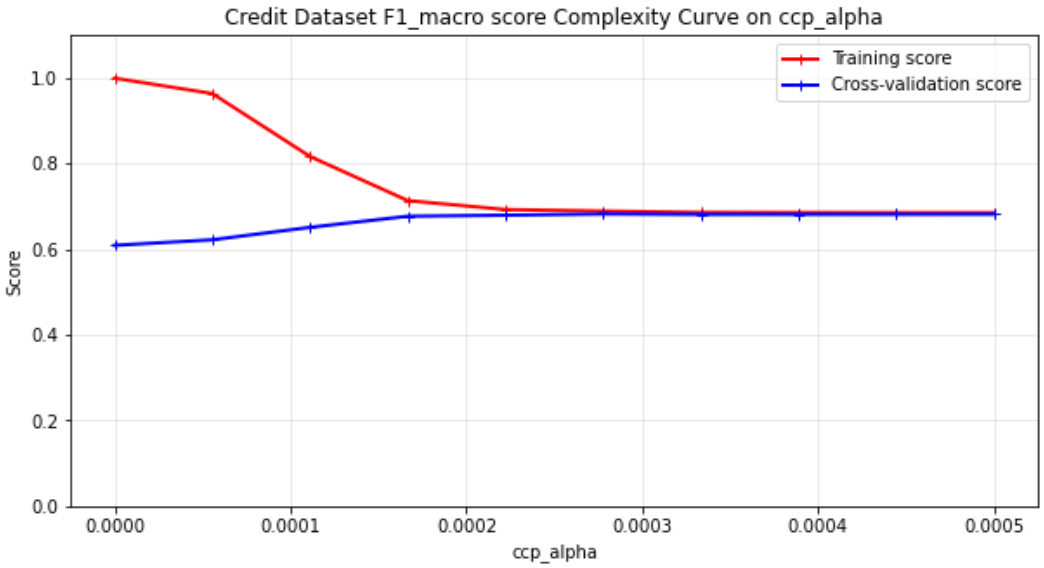
Also looking at the learning curves of each model and comparing the bias and variance, the Decision Tree also has the best performance especially as the size of the data increases, as we continue to see the model overfit less as we saw with the adult dataset. Interestingly though K-NN performed better on the Credit Dataset than on the Adult Dataset.

It is clear that overall the credit dataset is a more fitting dataset that works well with most of the classifiers implemented compared to the adult data set as it has less categorical data and more continuous data that can be normalized and are better suited for algorithms such as Neural Networks and KNN.

4.2.2 Improving Performance of models.

Similarly with the adult data, I was able to improve the performance of the models by using a validation curve along with a gridsearch to identify the best pruning parameters such as alpha parameter for the decision tree model, hidden layers for the Neural networks, number of estimator bags for Boosting, gamma for SVM and number of K-neighbors for KNN. Focusing on just the decision tree since it is the best performing, there was no clear value of the parameter ccp_alpha that improved the performance of the model, unlike with the adult dataset as shown in the chart below. I followed similar steps for the other models as well where I leveraged a validation curve, grid search and cross validation where appropriate to improve the performance of each algorithm and try to trade-off between the variance and bias of the models.

I also implemented the models using cross validation but it did not make a significant difference on the results. (See jupyter notebook for Cross-Validation results)



4.2.3 Worst Performing model on Credit Dataset and why.

The worst performing model on the credit dataset was the Support Vector Machine, unlike with the adult data whose worst performing model was the Neural Networks looking at the same metrics.

I believe the Support Vector Machine algorithm did not perform so well on the credit dataset because of the non-homogenous nature of the credit dataset both carrying a mix of categorical and continuous variables.

4.2.4 Did Cross Validation Help?

I also implemented the models using cross validation, both simple and stratified and although on the credit data set it slightly improved the accuracy scores, the improvement was insignificant, by only 2 decimal places. (See jupyter notebook for Cross-Validation results)

5 CLOCK TIME PERFORMANCE ANALYSIS

For both datasets, as can be seen in the charts from section C, the Support Vector Machine took the longest clock time to train by a long shot and even more as the size of the data increased. The next slowest model was the Boosting ensemble closely followed by the Neural Network. While the Decision Tree algorithm was the fastest in this case on both datasets.

7 REFERENCES

1. UCI Adult Data Set: https://archive.ics.uci.edu/ml/datasets/Adult
2. UCI Credit Card Data Set from Taiwan with history payments and binary: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
3. Source Code for sample decision tree with datacamp using sklearn: https://www.datacamp.com/community/tutorials/decision-tree-classification-python
4. Source Code for sample neural network classifier using sklearn: https://scikit-learn.org/stable/modules/neural_networks_supervised.html
5. Source Code for sample Adaboost classifier using sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html#sklearn.ensemble.AdaBoostRegressor
6. Source Code for sample SVM classifier using sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
7. How to use stratifiedkfold cross validation with sklearn with geeks:: https://www.geeksforgeeks.org/stratified-k-fold-cross-validation/
8. How to plot cost complexity curves with sklearn by sklearn: https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py
9. How to plot learning curves with sklearn by sklearn: https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html
10. How to plot validation curves with sklearn by sklearn: https://scikit-learn.org/stable/auto_examples/model_selection/plot_validation_curve.html#sphx-glr-auto-examples-model-selection-plot-validation-curve-py
11. Source