Supervised Learning Assignment

Description:
In the first part of the assignment, I wanted to choose a non-trivial dataset that could be able to make predictions on an interesting, scalable situation. My data has information of about 47,194 adult people. On Table 1, we can observe the attributes of the data and their relevant information.

Attribute	Type	Domain
Age	Continuous	[18, ∞]
Work class	Categorical	Private, Self-employed, Federal-government, Local-government, State-government, Without-pay, Never-worked
Weight	Continuous	[0, ∞]
Education	Categorical	Bachelors, Some-college, 11th, High school-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
Education-numerical	Categorical	0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
Marital-status	Categorical	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Occupation	Categorical	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
Relationship	Categorical	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Race	Categorical	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Binary	Female, Male
Capital gain	Continuous	[0, ∞]
Capital loss	Continuous	[0, ∞]
Work hours per week	Continuous	[0, ∞]
Native Country	Categorical	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands

Table 1: Adult dataset information

As we can observe, there are 14 attributes that describe each person, each instance, as much as possible. Intuitively, by looking at attributes such as education level, capital gain and native country, we can generally get a good sense of the economic situation of a person. This data maps a variety of attributes to the

annual income of 47,194 subjects. In this report, I will compare different Supervised Learning algorithms to predict whether a person has an annual income over 50K. Consequently, the output of the models I will build is binary, with two possible outcomes: >50K and <=50K.

I believe the adult dataset is interesting because of the diverse array of attributes that can vastly make an impact on whether a person makes over 50k a year. By merely using my intuition and common sense, I would think that there are some variables such as Native Country and Education that will have more influence on the output. For instance, it is highly probable that someone who graduated from college and whose native country is the United States makes more money that someone who didn't completed high school in an under-developed country. Similarly, attributes like sex and race can have some influence on individuals from several countries where discrimination is still present. As we can notice, it is going to be very interesting to find out how the algorithms will behave with these attributes, and it is also going to be fascinating to discover what attributes have the most influence on the income.

For the second dataset, I tried hard to find data that differs from the first one. In my efforts to do so, I found a dataset related with marketing campaigns of a Portuguese banking institution. These campaigns were based on phone calls. With this data, I will explore different algorithms to figure out which one can predict whether the client acquired the product (term deposit) or not. It is important to mention that this is a smaller dataset, containing 8,923 records. This second dataset is very interesting due to its nature; it maps whether a client purchased a banking product or not depending on 16 different attributes related to the client's features and to the communication between the client and the bank. By finding a model that approximates the behavior of this data, banking institutions could improve their marketing strategies to increase the sales of their products. Finding which attributes

have greater influence on the output and those that had no influence at all, will yield solid conclusions regarding this classification problem.

On Table 2, we can observe the attributes of this data and their description.

Attribute	Type	Domain
Age	Continuous	[18, ∞]
Job	Categorical	Admin., Unknown, Unemployed, Management, Housemaid, Entrepreneur, Student, Blue-collar, Self-employed, Retired, Technician, Services
Marital	Categorical	Married, Divorced, Single; *note: divorced means divorced or widowed
Education	Categorical	Unknown, Secondary, Primary, Tertiary
Credit Default	Binary	Yes or No
Balance	Continuous	Average yearly balance, in euros
Housing	Binary	Yes or No
Loan	Binary	Yes or No
*Contact	Categorical	Contact communication type: Unknown, Telephone, Cellular
*Day	Discrete	Last contact day of the month; numeric
*Month	Categorical	Last contact month of year. Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
*Duration	Continuous	Last contact duration, in seconds
Campaign	Continuous	Number of calls performed during this campaign and for this client
P_days	Continuous	Number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)
Previous	Continuous	Number of contacts performed before this campaign and for this client
P_outcome	Categorical	Outcome of the previous marketing campaign. Unknown, Other, Failure, Success

* related with last contact of the current campaign

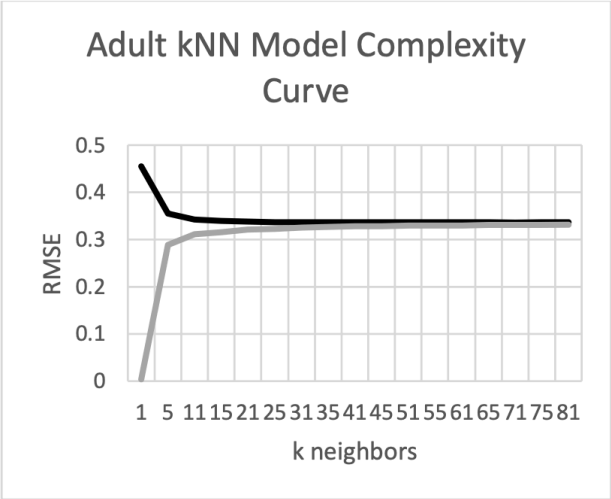
Table 2: Bank dataset information

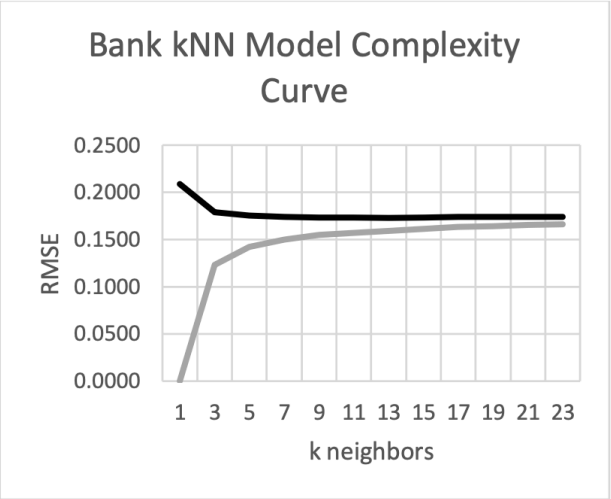
After taking a glance at the second table, it is easy to notice differences between both data. The Bank dataset has more binary attributes and its categorical attributes have smaller domains. This could lead us to believe that it would be easier to create an accurate model from the Supervised Learning algorithms. However, the banking data has less instances and two more

attributes, which can make it harder to build a model due to the curse of dimensionality. It will be very interesting to discover which dataset can be modeled with the least possible error.

k-Nearest Neighbors

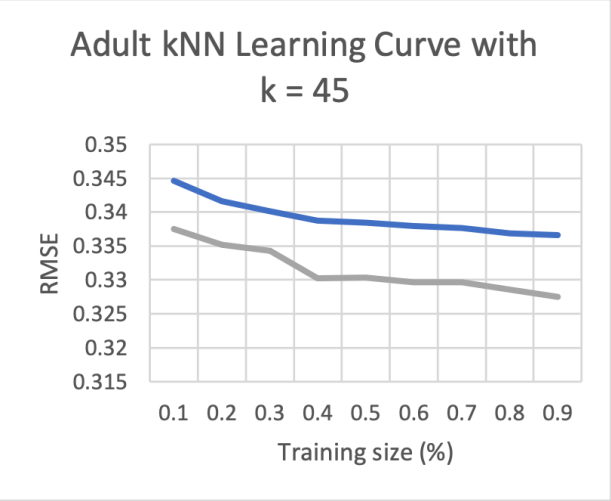
KNN is a non-parametric lazy learning algorithm. This means that it does not make any assumptions on the underlying data distribution and that it does not use the training data points to do any generalization. In other words, there is no explicit training phase so the algorithm makes a decision based on the entire training data set. kNN algorithm was relatively fast for both datasets; however, it required a great amount of memory space since it stored all training data. My implementation of this algorithm in Weka software used Euclidean distance to calculate how far away is each point in the feature space from the target. Now, let's analyze how this specific algorithm behaves as we increase the number of neighbors k. In the following graphs, the black curve represents the 10-fold cross-validation error and the gray curve represents the training error.

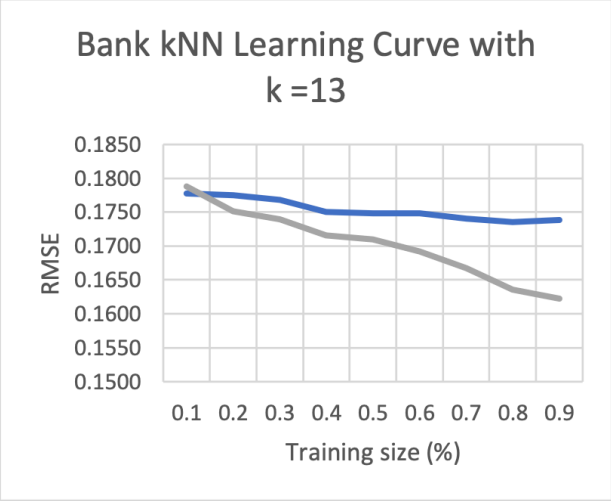




As we can observe, both the adult and bank dataset yielded similar graphs for the model complexity curve. In cross-validation, as I increased the number of neighbors that the algorithm takes into account to make a decision, the error decreased dramatically first until it reached a plateau when k equals 45 in adult and 13 in

bank. By Ockham's Razor, I chose these number of neighbors, 45 and 13, as the best complexity because they provided the same results even though I tried to increase k. We can conclude that adult data has more noise than the bank data because there is a steeper decrease in CV error. On the other hand, we can observe that both training curves increase dramatically at the beginning, reaching a plateau as well as k increase. When k is 1, the algorithm basically selects the points it is looking for, producing basically no error at all.

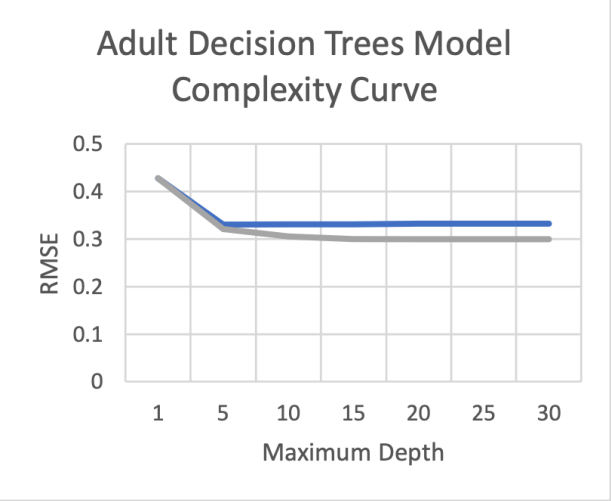


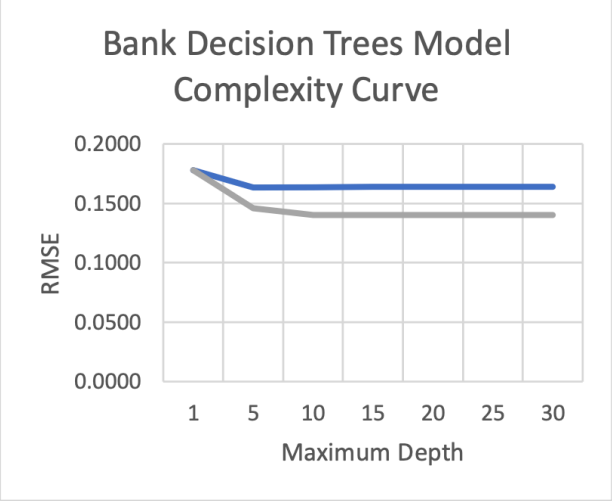


After looking at the learning curves, we can notice that they are quite different. On one hand, there is high variance on the adult one because the cross-validation curve (blue) and the training curve (gray) have a significant distance between them. There isn't a specific training percentage that performs better than the others. If we had more data, we could reduce this variability issue and both lines could probably converge as training size increase. On the other hand, the bank learning curve depicts high bias as both curves seem to diverge as training size increases; if more data would be available, these two lines will continue to diverge.

Decision Trees

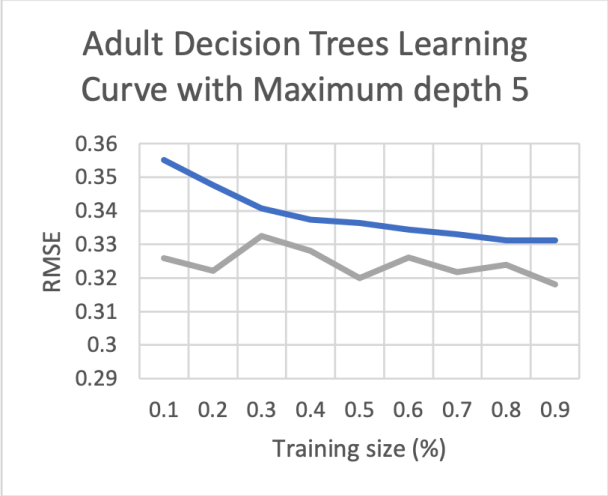
Decision trees learn from the training data, and it builds a predictive model which is mapped to a tree structure. The goal is to achieve perfect classification with minimal number of decisions. In this assignment, I used REPT decision tree in Weka, which is a fast learner that uses information gain to expand its nodes and backfitting reduced-error pruning. This algorithm was the fastest classifier by far, and at the same time it generated more accurate models than other complex algorithms we will discuss later. I was interested in varying the maximum depth of the tree to increase the complexity.

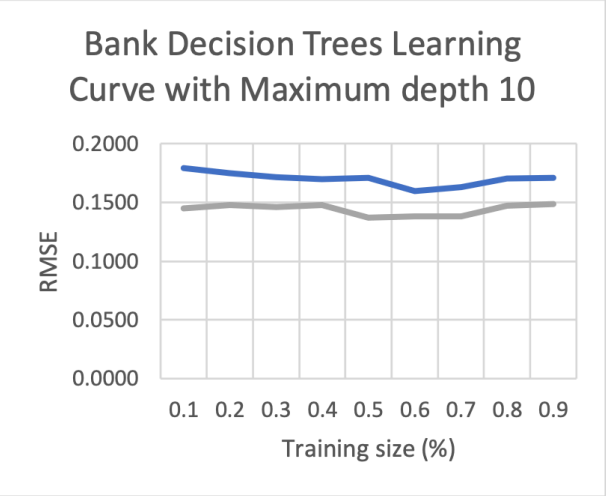




Both datasets produced interesting results for the model complexity graphs. First, it can be noticed that small depths, between 1 and 5, generate inaccurate trees. Since we don't allow them to continue using information gain and expand, these small trees make decisions on very few attributes, leading to a high number of wrong classified instances. Surprisingly, both dataset don't have signs of overfitting. Both, the adult and bank datasets reach a plateau at a max depth of 5 and 10 respectively, and, as we can see, the cross-validation curve never seems to increase after this. We can conclude that this specific decision trees are not suffering from overfitting, and the reason this occurs is that

the Weka REPT tree uses an aggressive pruning technique. It cuts any nodes that won't lead to a significant decrease in accuracy. Therefore, I think that no matter by how much we increase the maximum depth, we will end up with extremely similar trees. A possible explanation for this is that both datasets have few attributes with high information gain; basically, they have few attributes that have a significant influence in the output. All the other attributes might have small correlation with the output, again making it hard to create an accurate model.





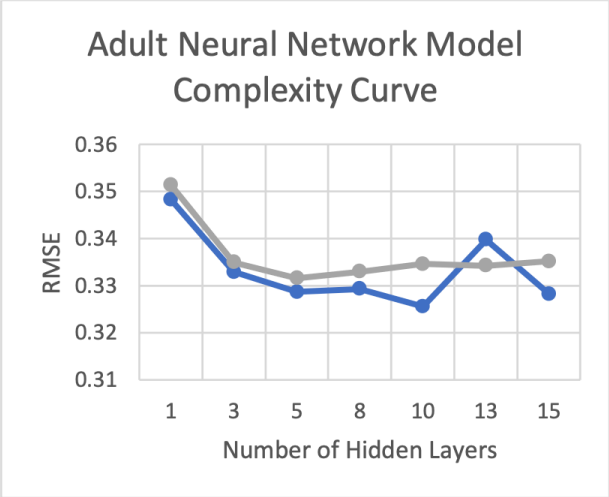
These learning curves have high bias; even if we had more data to run our decision trees, the cross-validation curve and the training curve will not converge. Moreover, we can observe that the adult training curve (gray) seems to oscillate; there isn't a clear picture of whether more training data increases the error. This can be caused by all the noise in the adult data. Even though this set has 6 times more instances than the bank one, it is much noisier, as it has attributes that give us almost no information gain at all.

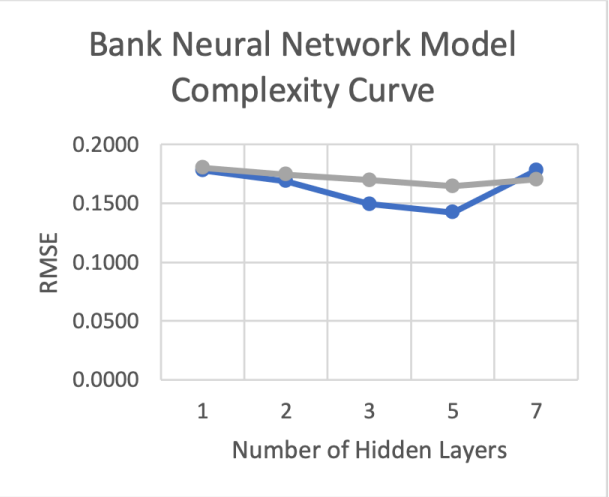
Neural Networks

A Neural Network is an algorithm that simulates the behavior of the brain and neurons. It is composed by a basic unit called the perceptron, which is connected with many others; links can be

enforcing or inhibitory in their effect on the activation state of connected neural units. It uses an activation function which has a threshold on each connection: such that the signal must surpass the limit to propagate to other perceptrons.
The Weka implementation of this algorithm is a Multilayer network that uses backpropagation to classify instances.

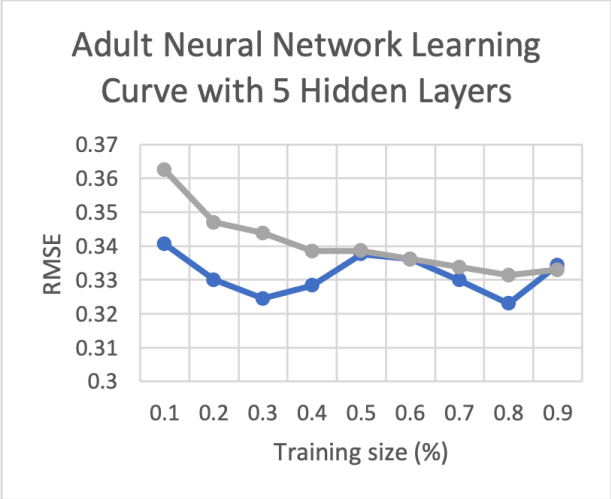
My Neural Networks was one of the slowest algorithms, however, it produced some of the most interesting results. I decided to vary the number of hidden layers in the network to increase the complexity.

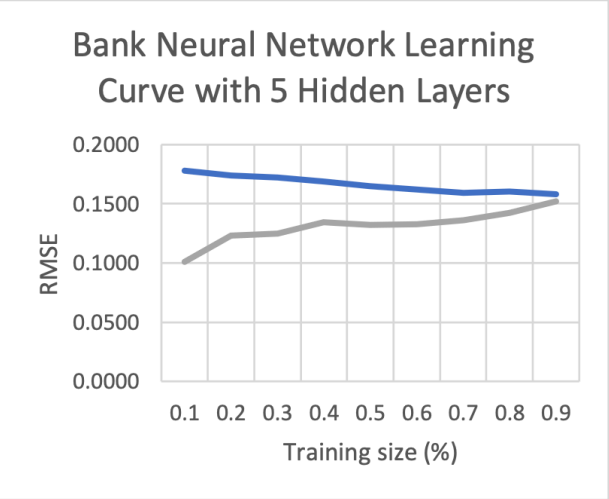




As more hidden layers were used, there was a significant increase in accuracy in the first part of both graphs. When we increased the numbers of layers beyond 5 for both adult and bank, the effects of over-fitting can be noted by the increase in the cross-validation error after reaching the optimal complexity. This is caused because the Neural Networks believe too much in the data. With more hidden layers, this algorithm built more complex model trying to follow misleading patterns in data. It is important to mention a couple of issues with both graphs. On one hand, we can see that for the adult data, the curves are smooth except when there are 13 hidden layers. This unexpected increase in the error is another sign of noise in our data and of the high variability among the instances. On the other hand, the bank cross-

validation curve doesn't show a significant decrease in the error when complexity increases, and the training curve has a surprising peak in error when there are 7 hidden layers.



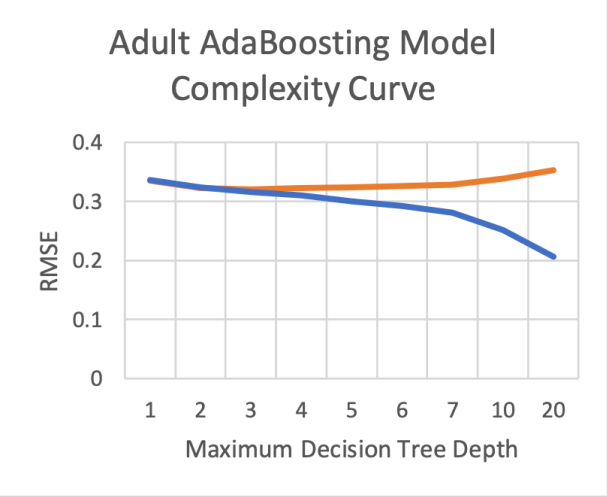


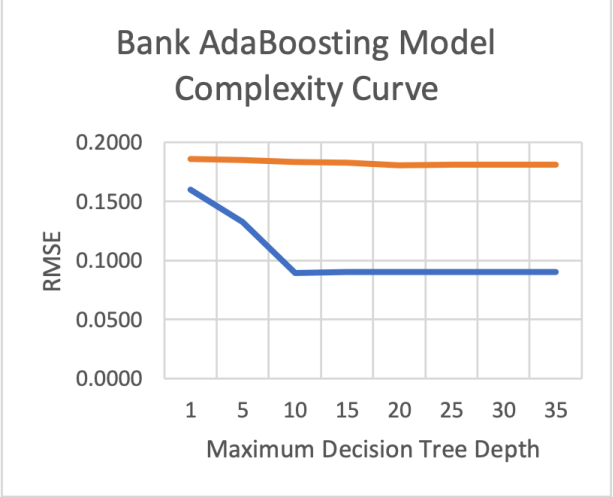
After increasing the training size and plotting the Learning curves for both data, we reinforce some ideas mentioned in the previous paragraph. First, by looking at the learning curve for bank, we get a decreasing cross-validation error and an increasing training error. This graph shows high variance in the data, since there is a big gap between the two curves. However, as training size increases, these lines seem to converge which means that more data instance could fix the high variability issue. Now, let's shift our focus to the adult learning curve. Even though the two lines seem to converge in the end, we can observe a that the training error decreases as more data is fed to the Neural Network, except in the middle where there is an increase.

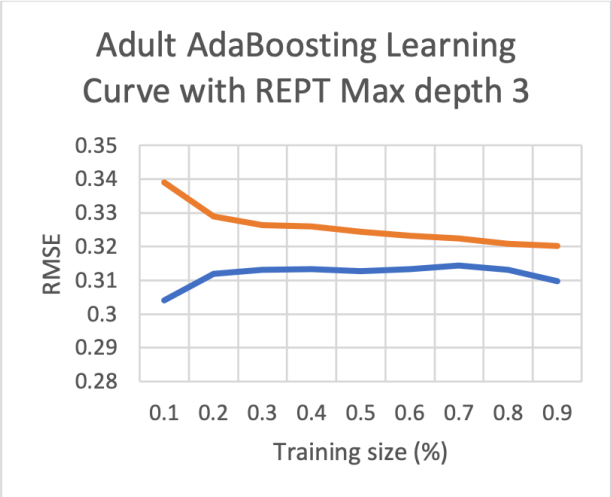
Boosting

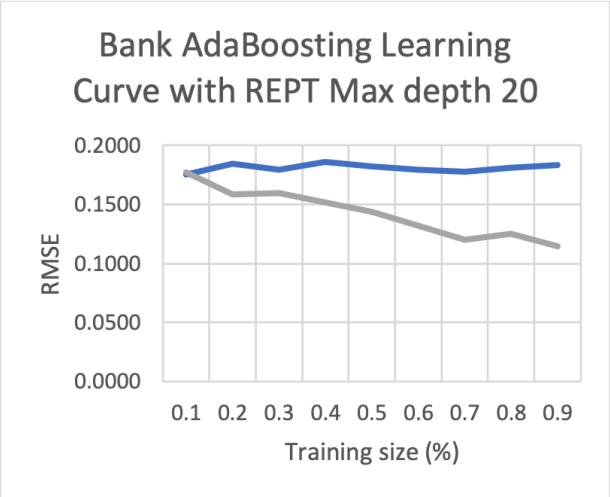
Boosting is an ensemble algorithm used in Machine Learning to reduce bias and variance. The idea behind this technique is to take a weak learner and improve, or "boost", its performance by running iterations, and produce a strong learner which is built by the weighted sum of the iterations. In this experiment, I used Weka's AdaBoost classifier, and I decided maintain the ideal number of iterations constant, in this case 10. Adaboost used

REPT decision trees as the weak learner. Hence, I wanted to increase the complexity of this Boosting technique by increasing the maximum depth allowed for the weak learner.







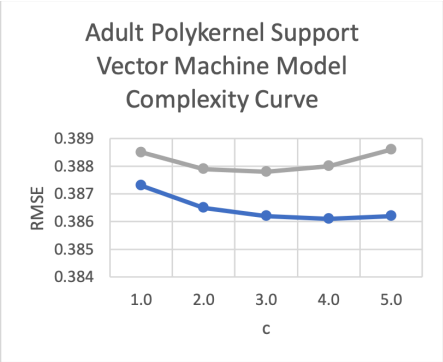


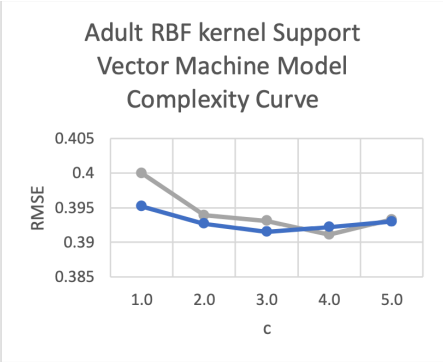
It was expected that a noisy data such as the adult one has a low decision tree maximum depth when boosting the algorithm. This means that the numbers of attributes that have significant information gain are small. The rest is just again a sign all the noise present in this data. There is evident overfitting depicted by the CV graph, as the tree gets more complex, we deviate from our ideal model. The learning curve is smooth; as we increase training size, CV error decreases while training error increases. These curves seem to converge if we had more instances. This algorithm has the lowest error across all the classifiers and it was

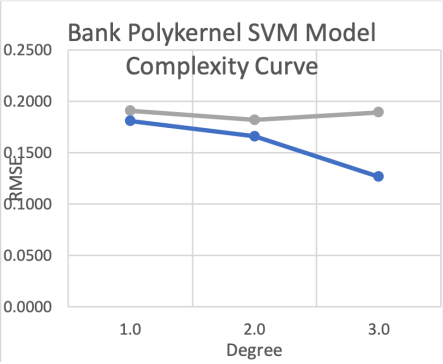
fast. With the bank dataset, boosting doesn't overfits data at any point. The model complexity curves reach a plateau and have the same issue as the decision trees we analyzed before. The learning curve for bank is the total opposite of the adult since in this case, the two lines diverge as training size increase.

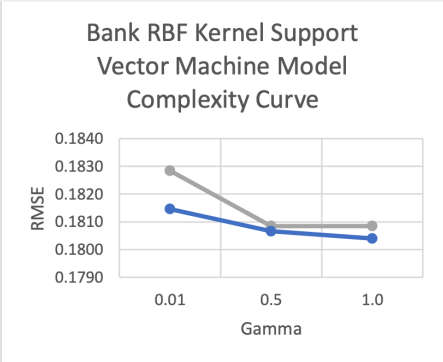
Support Vector Machines

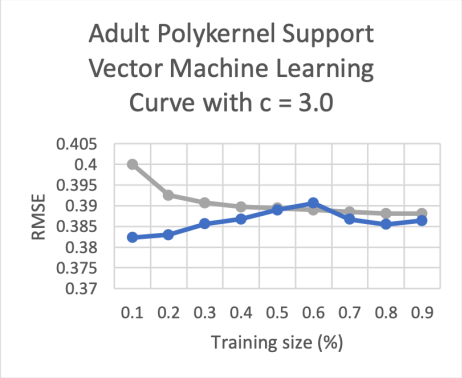
For the support vector machine experiments, I wanted to test different things to compare how this complex algorithm performs with these two-different data. I used two different kernels: a polynomial kernel with varying degree and a radial basis function kernel (RBF) with varying gamma. With the adult data, I varied the c-value to alter the complexity and with the bank data, I increased the degree of the polynomial kernel and the gamma of the RBF. Of the machine learning algorithms performed in this report, SVM produced the worst results for the adult data, resulting in terrible training and testing results for the RBF kernel. The Polynomial kernel produced the worst results for the bank dataset.

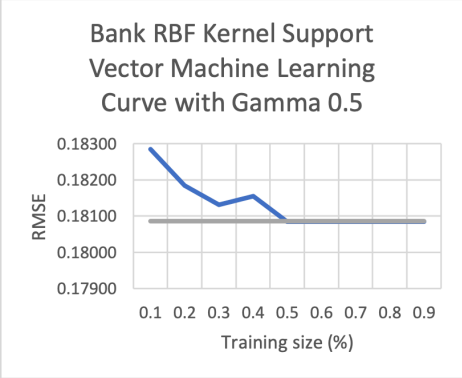










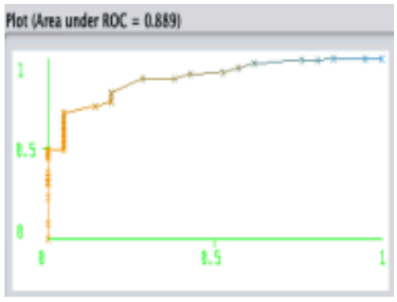


There are several interesting conclusions we can make on the six previous graphs. The first thing to note about these results is how the polynomial kernel performed significantly better for the adult dataset whereas the RBF kernel performed significantly better for the bank dataset (although the results produced were the worst among the other machine learning algorithms). In the adult set, we varied the c value, which is the slackness of the support vectors used. We can observe that with the polynomial kernel, the graph portrays a CV curve with overfitting when increasing c beyond 3.0 and the training error steadily decreased. For the sake of consistency, I continued to run this best configuration of polykernel SVM on the different training sizes of the adult dataset. Again, the effects of high variance in the adult data is depicted by this graph. There doesn't seem to be a steady decrease in the training error. On the other hand, the bank data produced results that were unexpected because the training error remained the same across the different splits of the training data. The training size appears to have no effect on the RBF Kernel with a Gamma value of 0.5. This fact shows that this bank classification problem is extremely hard to model with SVMs due to the attributes of this data and to the small number of instances. This algorithm performs better with large datasets.

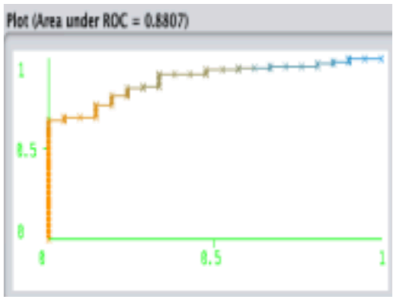
Conclusions: *ROC Curves*

After running all the Supervised algorithms described in this report, and after plotting all the model complexity and learning curves, we are going to finally test the best models built with the best hyper parameters against the testing set for each data. Now, we are going to find the best models that can accurately generalize. The ROC curve is a graphical plot that illustrates the performance of a binary classifier as it's a discrimination threshold is varied. This graph is a visual representation of how many false positives you are willing to take to increase the number of true positives. The real power of ROC curves is behind the area under the curve. This area represents the accuracy of a model; it tests how well the model separates the group being tested into, for instance, those with annual income greater than 50k or lower.

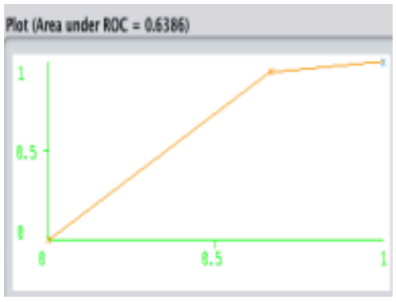
Adult ROC curves



Best: AdaBoost



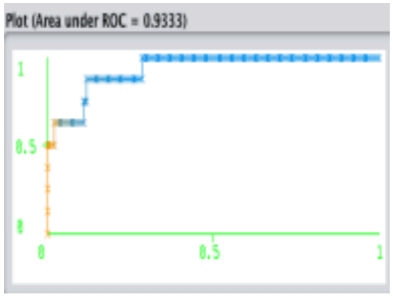
Good: Neural Network



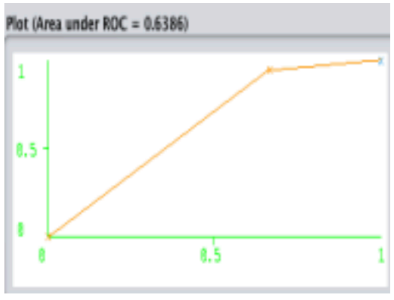
Worst: Support Vector Machine

Bank ROC curves

Best: Decision Tree



Good: Neural Network



Worst: Support Vector Machine

*I plotted only the best, the worst, and a good ROC curves out of the 5 possible per data set.

For the adult dataset, the best classifier is AdaBoost, using REPT trees as the weak learner, with a maximum depth of 3 as the complexity. This algorithm was also the fastest learner out the 5 algorithms used in this assignment. Hence, boosting is the best in terms of accuracy and time. It has high area under the ROC curve, close to 1. This means that the number of false positives we must take by increasing the number of true positives is the lowest. For the bank data, the best classifier is the Decision Tree with a maximum depth of 5. As we noticed throughout the report, this data has high variability. This previous fact affected the performance significantly of different algorithms like Neural Networks and k-Nearest Neighbors more than it affected Decision Trees. The ROC is extremely good since the number of false positives we must take by increasing the number of true positives is low. The worst algorithm, in both datasets, is the Support Vector Machine. This classifier took by far the longest amount of time to train and it yielded the worst results. Even though I tried two different kernels and I varied the complexity of the models, I still got the worst results in terms of accuracy and time. The SVM ROC curves for both adult and bank data support the fact that this algorithm performs poorly with my datasets. The area under these curves is about 0.60, which is close to 0.5. This means that the True Positive rate is almost equal to the False Positive rate.