CS7641 Machine Learning, Supervised Learning

Edward Cates, ecates8@gatech.edu

SUMMARY

1. Abstract
2. Bank Deposit Subscription Data
3. EEG / Eye Data
4. Data Set Comparison
5. Implementation Details

1. ABSTRACT

The classification problems describe in this report are based on 2 data sets from the University of California Irvine's Machine Learning data set Repository. The first problem: Given a bank's client information, which consists of categorical personal information as well as client contact stats (how often, how recently), predict whether or not the client has subscribed (or will subscribe) to a term deposit. The second problem: Given the data dump from an EEG (a record of electrical brain signals), determine when the patient's eyes were open (or closed).

Finally, a third problem was considered, the results of which are not included save a brief synopsis of the failure: Given categorical personal information of credit card account holders as well as recent payment history, determine whether or not the account holder will default on the next credit card payment.

Both data sets have real-world usefulness but are primarily interesting because of the behavior they elicit from the models rather than immediate application to my day-to-day life.

2. BANK DEPOSIT SUBSCRIPTION DATA

Interest and Overview

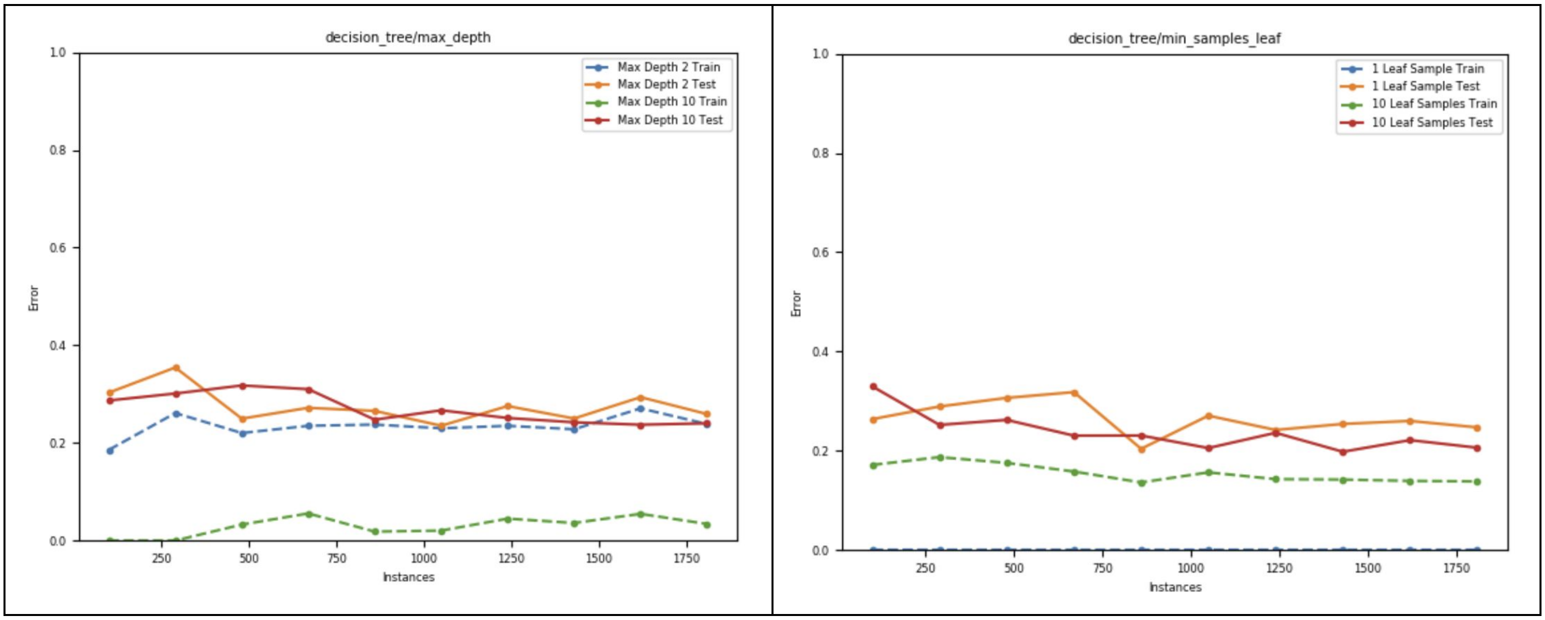
The identifying aspects of the bank subscription data set are the independent, categorical fields for personal data and the contact fields that profile the frequency of contact with a client. Models like decision trees, which can profile logically separated or conditional sets (i.e. sets whose fields are most useful in a series of "if" statements), will likely perform best on this data set.

Strong classification would allow bank representatives to better predict the likelihood of success with a given client based on known information, which would allow the bank to save a drastic amount of time in the sales process.

The attributes are: age, job, marital status, education, credit default status, housing loan status, contact communication type, month of last contact, day of week of last contact, last contact duration, number of contacts in current “campaign”, days between contact “campaigns”, number of contacts during previous “campaign”, outcome of previous contact “campaign”, employment variation rate, consumer price index, euribor 3 month rate, and number of employees.

Of the 45211 instances, 39922 had not subscribed to a long term deposit and 5289 had (11.7% were subscribed).

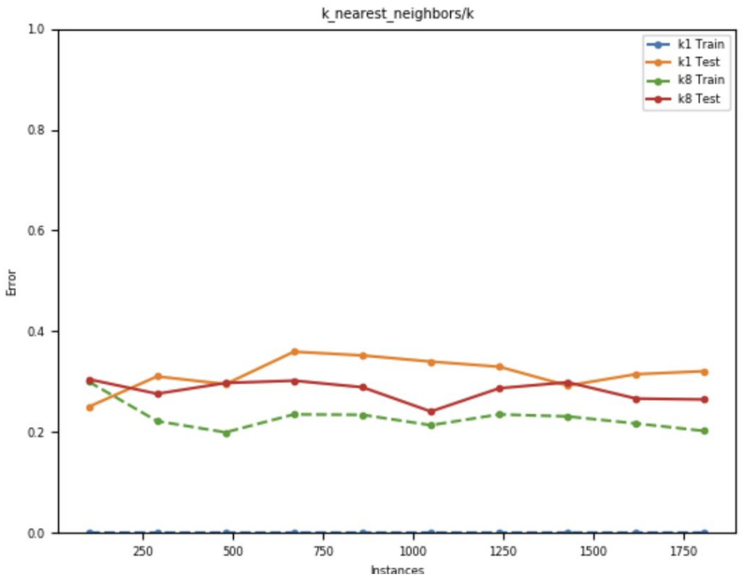
Decision Tree

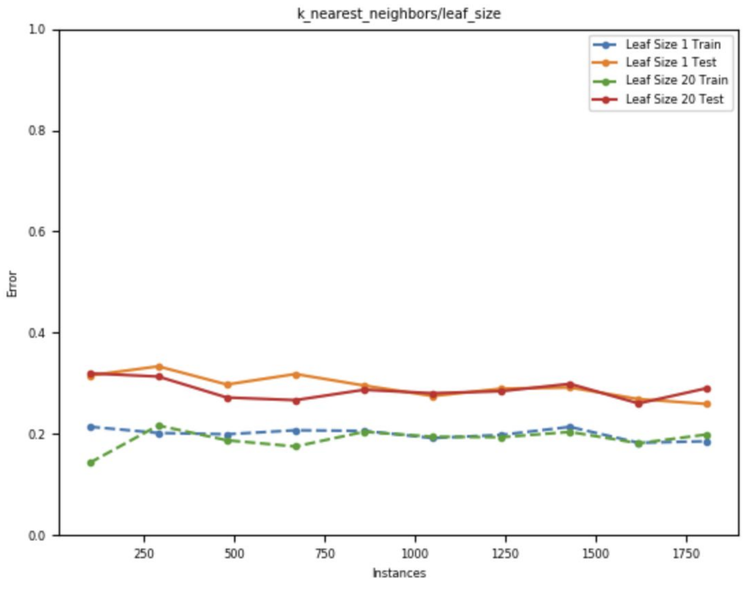


Overfitting	Error vs. Instances	Wall Clock Speed
samples per leaf = 1	Inverse relationship	Negligible time

Tree depth and number of features used (graph not shown) did not have a significant effect on performance.

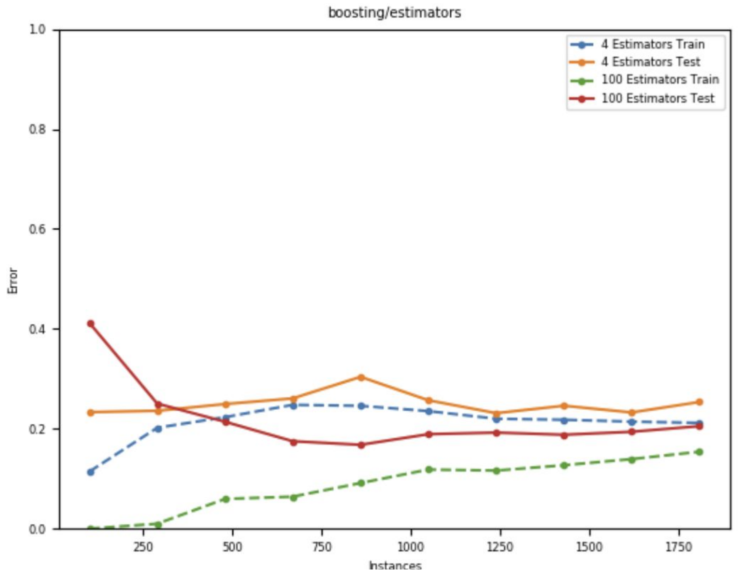
K Nearest Neighbors

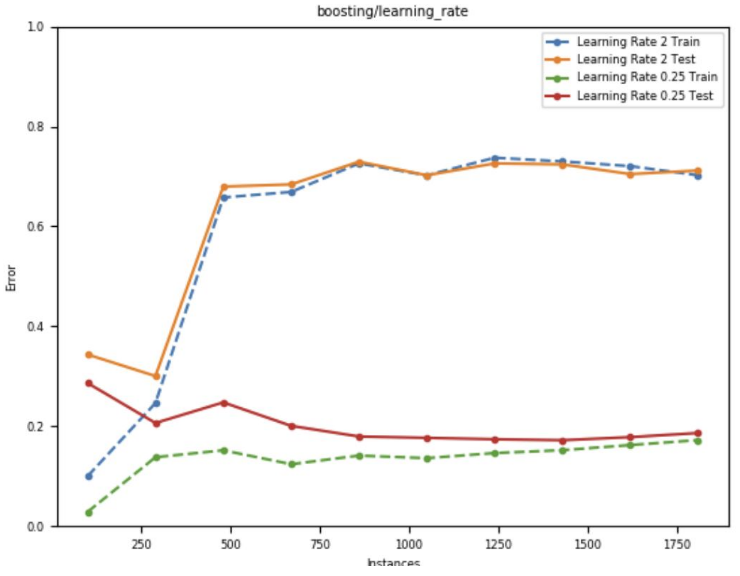




Overfitting	Error vs. Instances	Wall Clock Speed
k = 1	No trend	About 15s for cross validation

Boosting

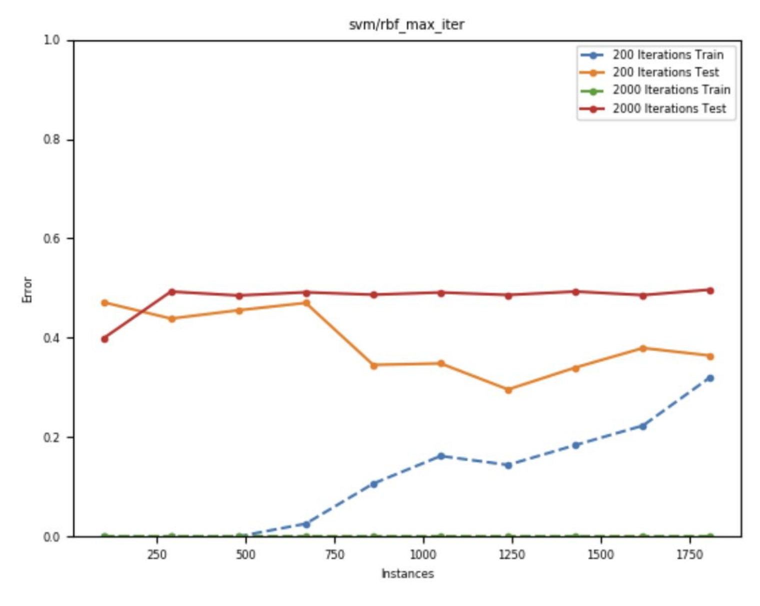


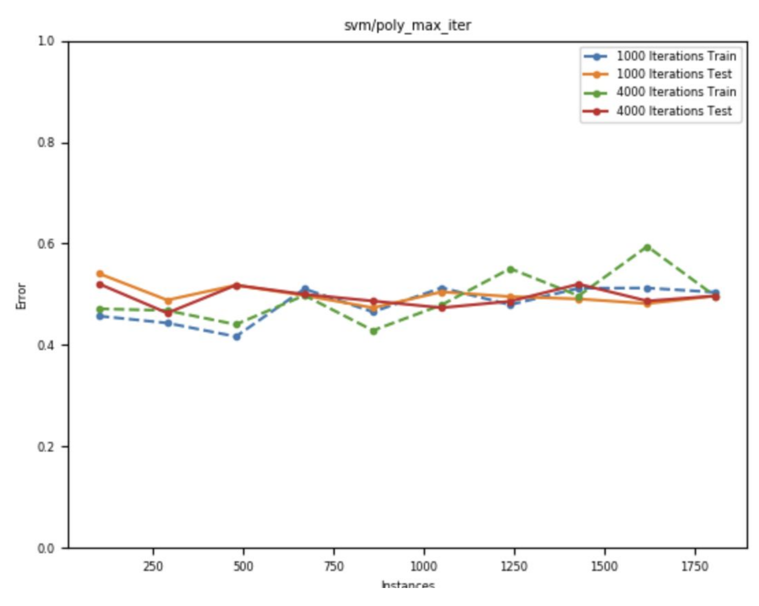


Overfitting	Error vs. Instances	Wall Clock Speed
k = 1	Slight trend for learning rate=0.25	About 10s for cross validation

The max features parameter (graph not shown) did not have a significant impact.

Support Vector Machine

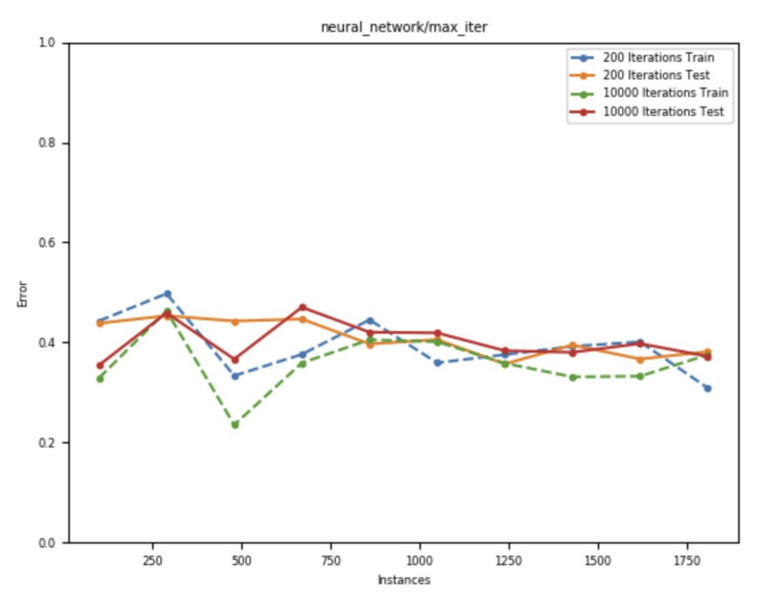


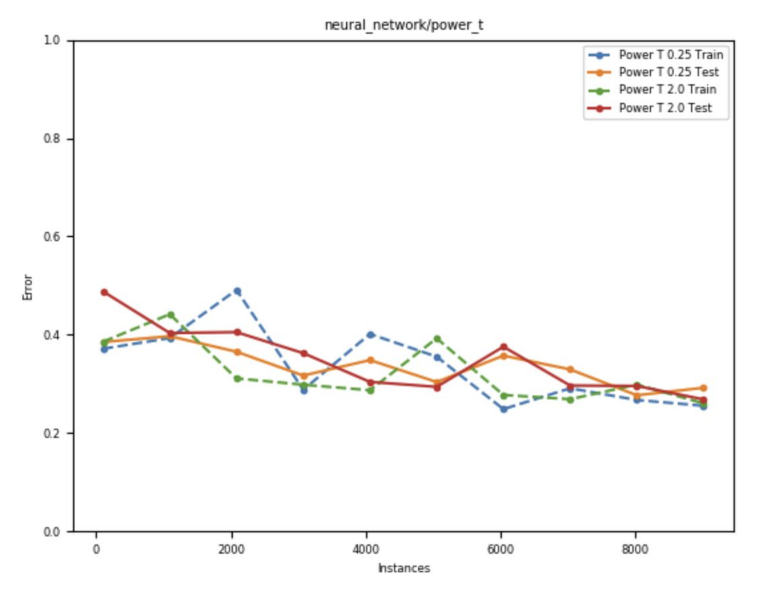


Overfitting	Error vs. Instances	Wall Clock Speed
2000 iterations	Unclear	About 5s for cross validation

Neither the RBF nor the polynomial kernel seemed to perform particularly well. Neither varying the number of iterations nor changing the penalty parameter (graph not shown) had any significant effect on the polynomial kernel classifier, implying that the polynomial classifier was fundamentally unable to separate the data points. This claim is supported by the fact that the classification rate was about 0.5 for all variations of the learner (clarification in section 6).

Neural Network





Overfitting	**Error vs. Instances**	**Wall Clock Speed**
None	Slight inverse relationship	About 120s for cross validation

The structure of the hidden layers of the neural network seemed to have no impact (graph not shown). The Power T parameter is describe by SciKit Learn as “the exponent for inverse scaling learning rate”).

3. EEG / EYE DATA

Interest and Overview

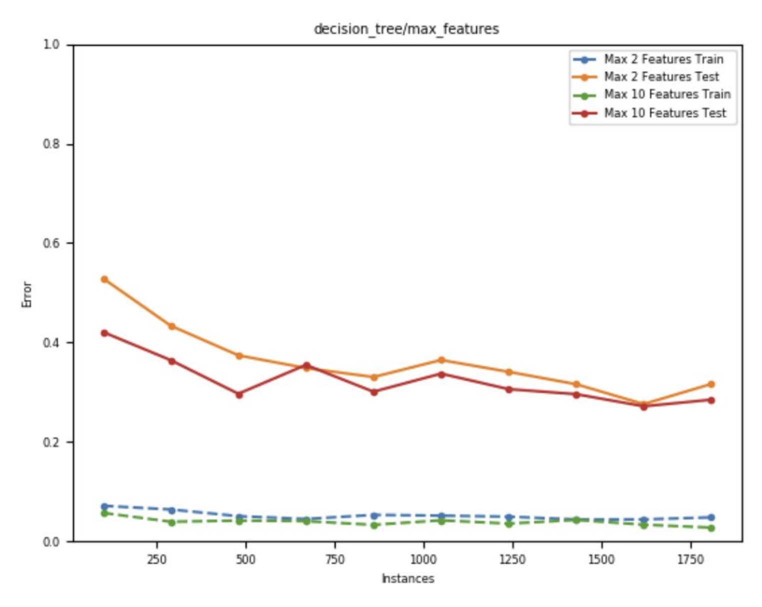
The EEG data is a dump of various monitor values that were recorded over a 117 second period. The y value of “eyes open / eyes closed” was added to the data later using a video recording of the person’s face. Since all of the EEG values are simultaneous and of the same human’s brain, they are very strongly related. For instance, if predicting whether not an automobile accelerator is floored or not based on various measurements of the engine, exhaust and RPM would be strongly related (not indepenent). K Nearest Neighbors should perform very well, since values that are close in proximity are very strong indicators for their neighbors. Learners like decision tree learners will likely work fine since they can represent complex functions, but won’t be able to represent the model as directly or efficiently. One would guess that a continuous function could be formed to describe the relationship between brain waves and eyelids.

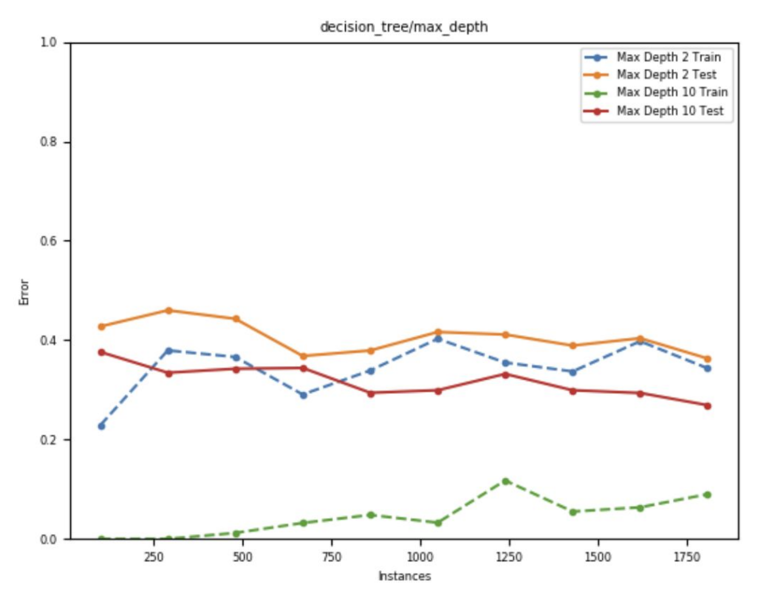
A strong classifier from this data set would allow researchers to determine if a person’s eyes were open or closed based only on information from the brain’s electrical signals. While this doesn’t seem immediately useful, I have a feeling it will somehow be useful in the field of robotics, augmented reality, or something similar. Also, learning how the brain more about the brain typically useful and interesting.

For the attributes, there are 15 recorded values, all assigned labels that, without an MD, might as well be random.

Of the 14979 instances, 8256 corresponded to eyes being closed and 6723 to eyes being open, meaning that the eyes were open about 55% of the time.

Decision Tree

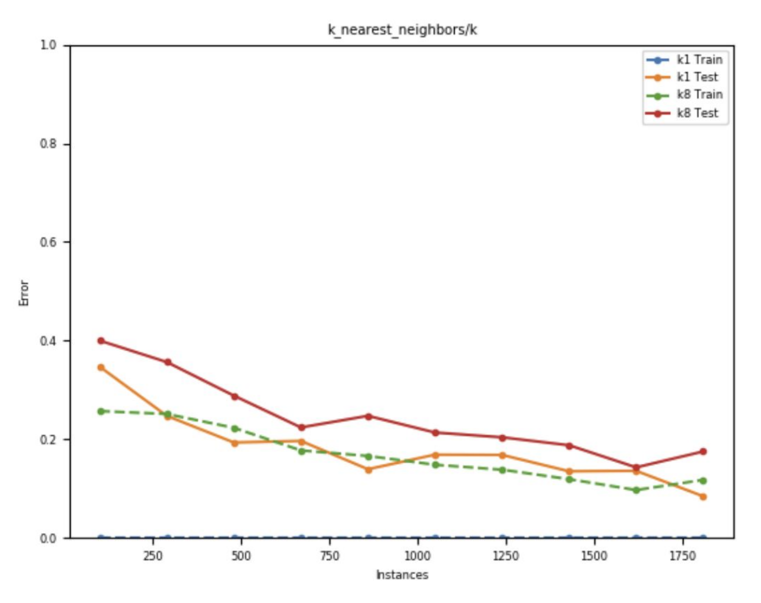


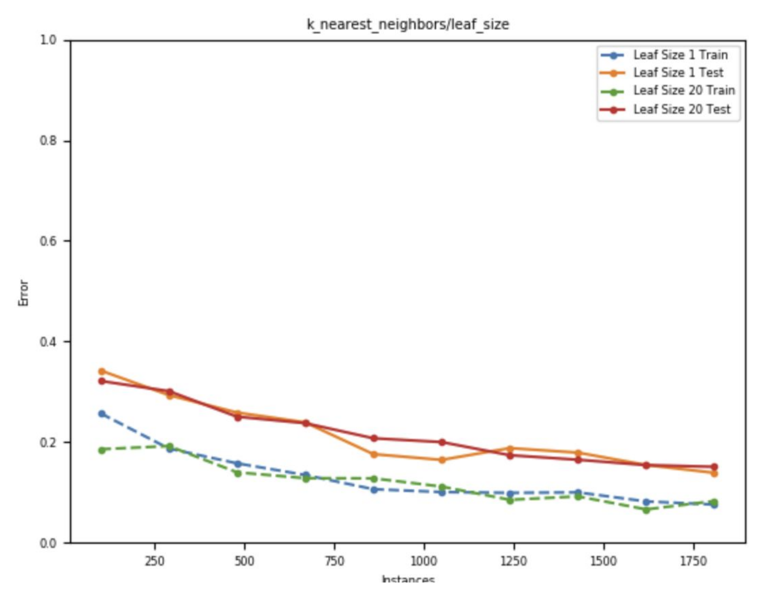


Overfitting	Error vs. Instances	Wall Clock Speed
None	Slight inverse relationship	Negligible

Interestingly, this data set did not overfit for 1 sample per leaf (graph not shown). A direct relationship existed for both tree depth and number of features. More values were tried for each parameter, but only the graphs above are shown in order to cleanly and visibly demonstrate the relationship.

K Nearest Neighbors

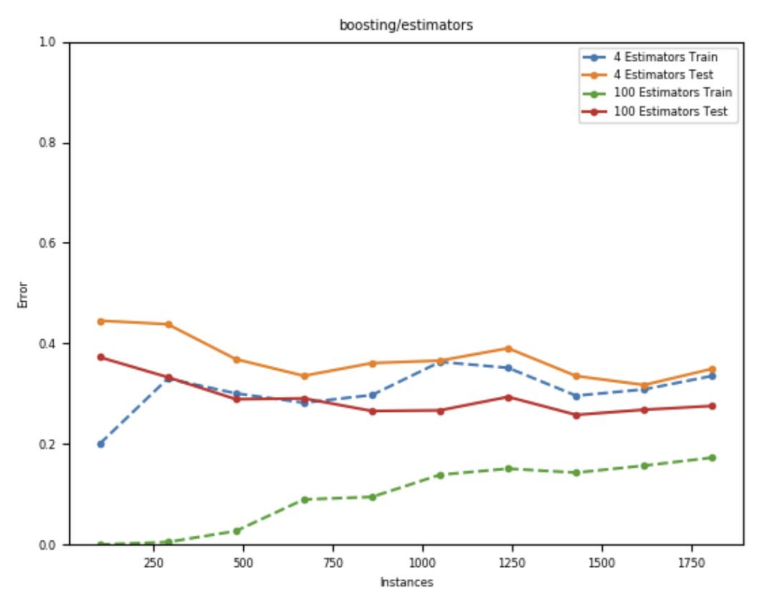


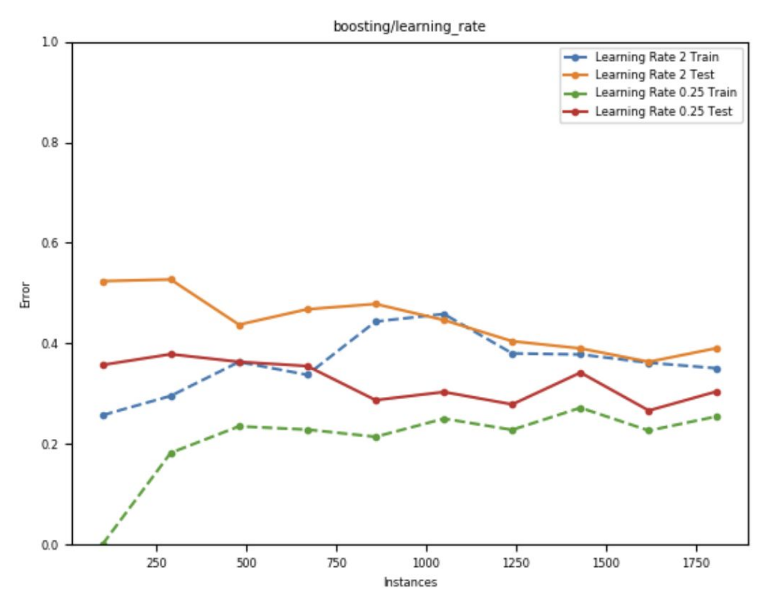


Overfitting	Error vs. Instances	Wall Clock Speed
None	Strong inverse relationship	10 seconds for cross validation

Surprisingly, the kNN classifier for k=1 did not overfit; in fact, it performed outstandingly well. kNN demonstrates that similar data points work very well as indicators for this data set.

Boosting

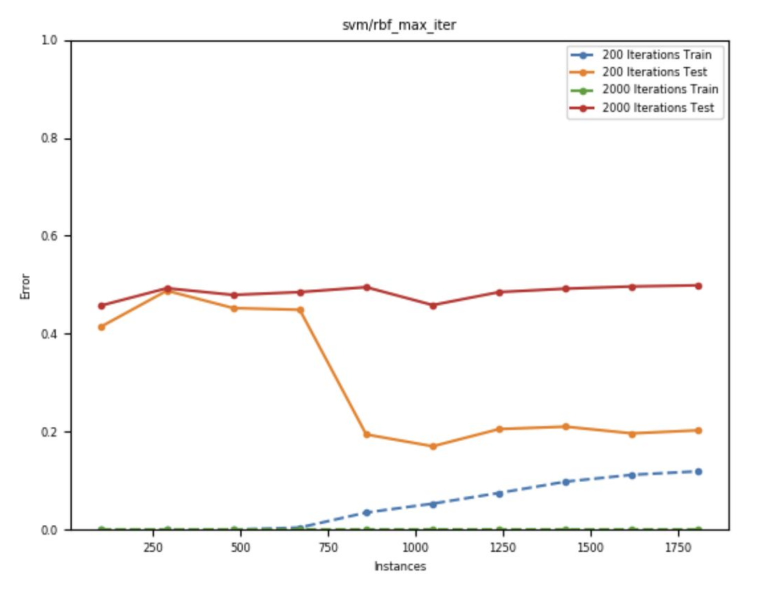


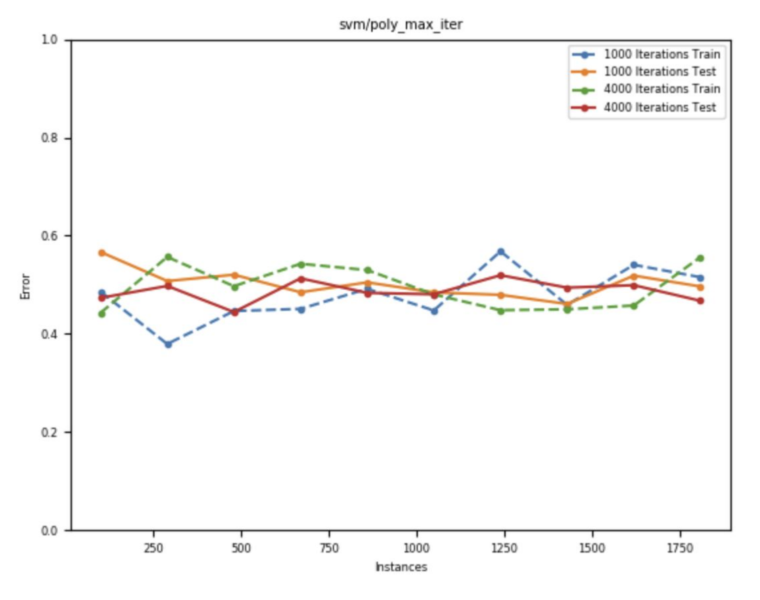


Overfitting	**Error vs. Instances**	**Wall Clock Speed**
None	Inverse relationship	15 seconds for cross validation

Boosting performs well, with the parameters working as expected - decreased error for more classifiers and lower learning rates.

SVM

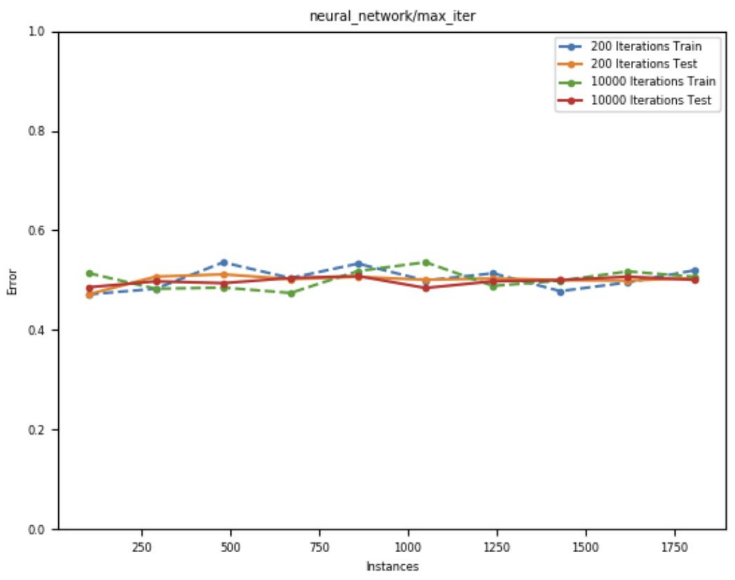


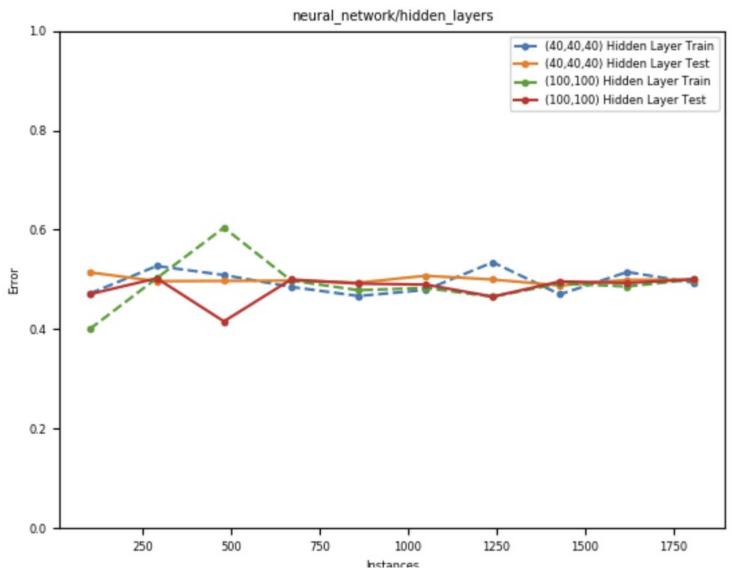


Overfitting	**Error vs. Instances**	**Wall Clock Speed**
2000 iterations	Inverse relationship	5 seconds for cross validation

The RBF kernel was able to separate the data, but more iterations cause the learner to level off. The polynomial kernel wasn't able to separate the data at all, as the classifier's error hovered right around random guessing for all tested parameter combinations.

Neural Network



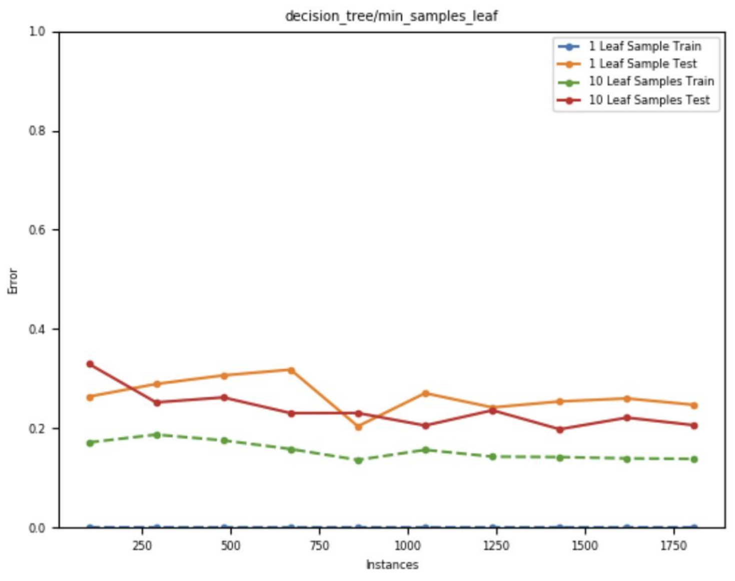


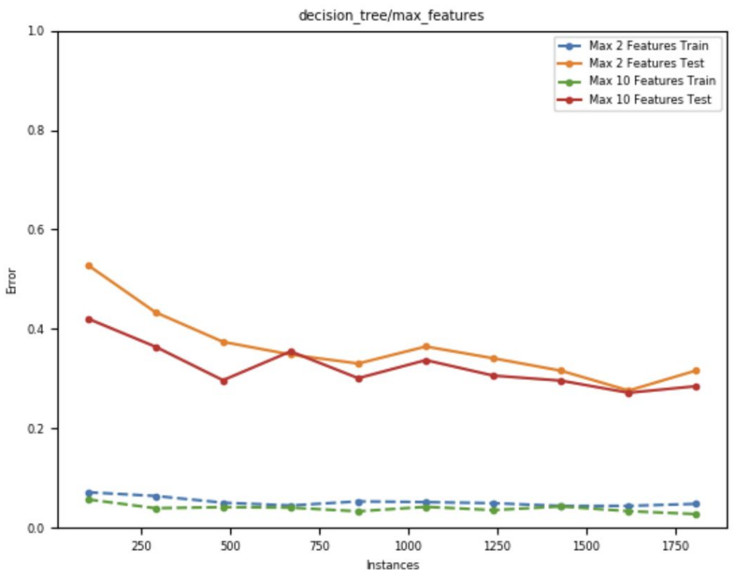
Overfitting	Error vs. Instances	Wall Clock Speed
None	None	25 seconds for cross validation

Regardless of the parameters, the neural network classifier performed about as well as random guessing.

4. DATA SET COMPARISON (Left: Bank data, Right: EEG data)

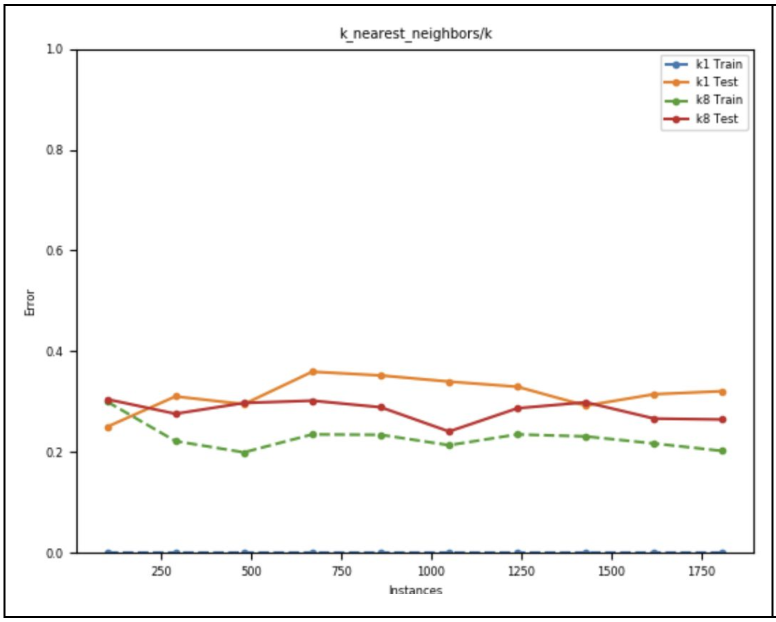
Decision Tree

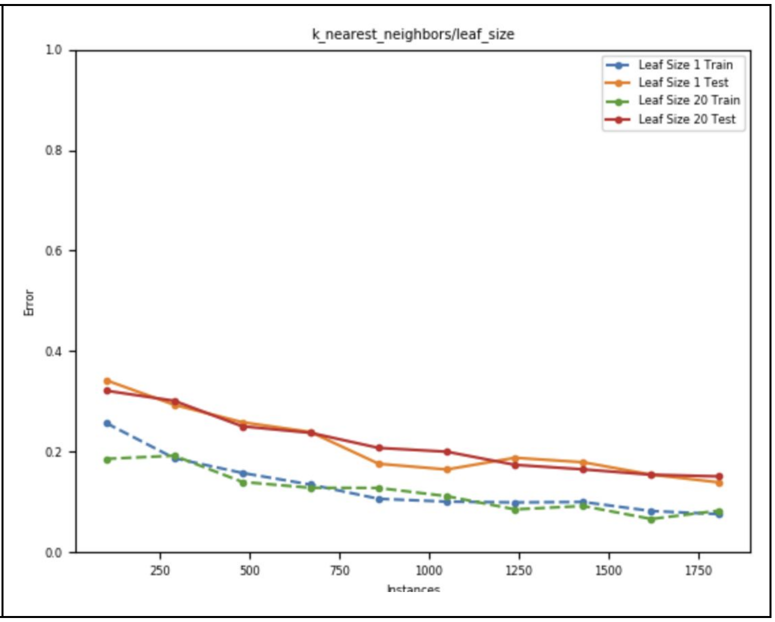




As explained in the overview sections, decision trees performed better on bank data than EEG data.

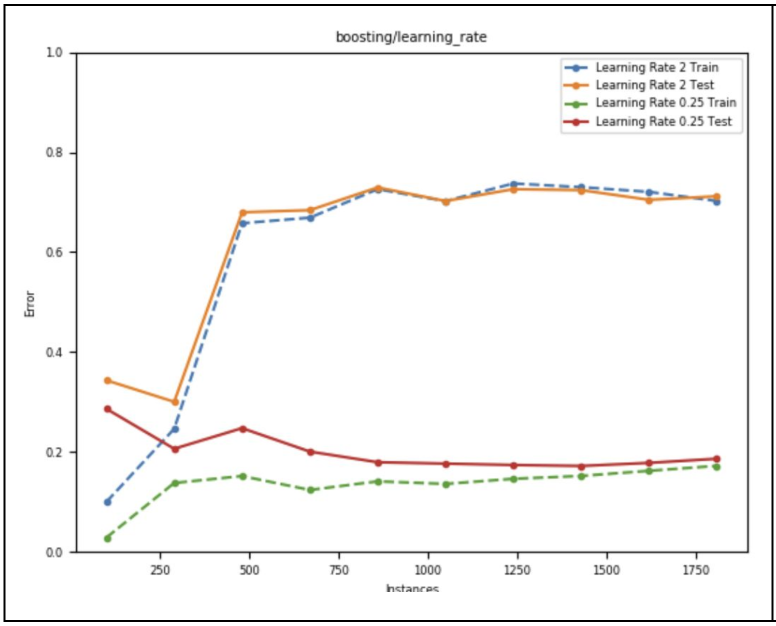
K Nearest Neighbors

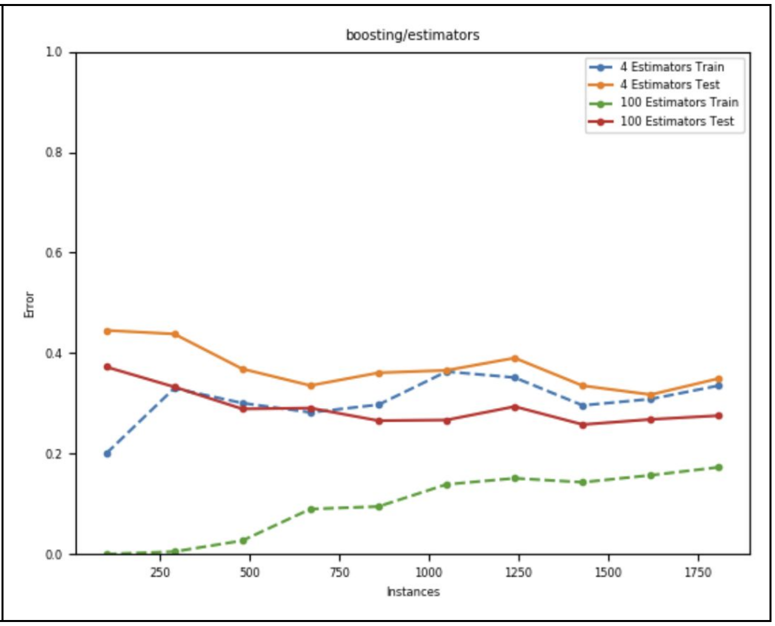




As explained in the overview sections, kNN performed better on EEG data than bank data.

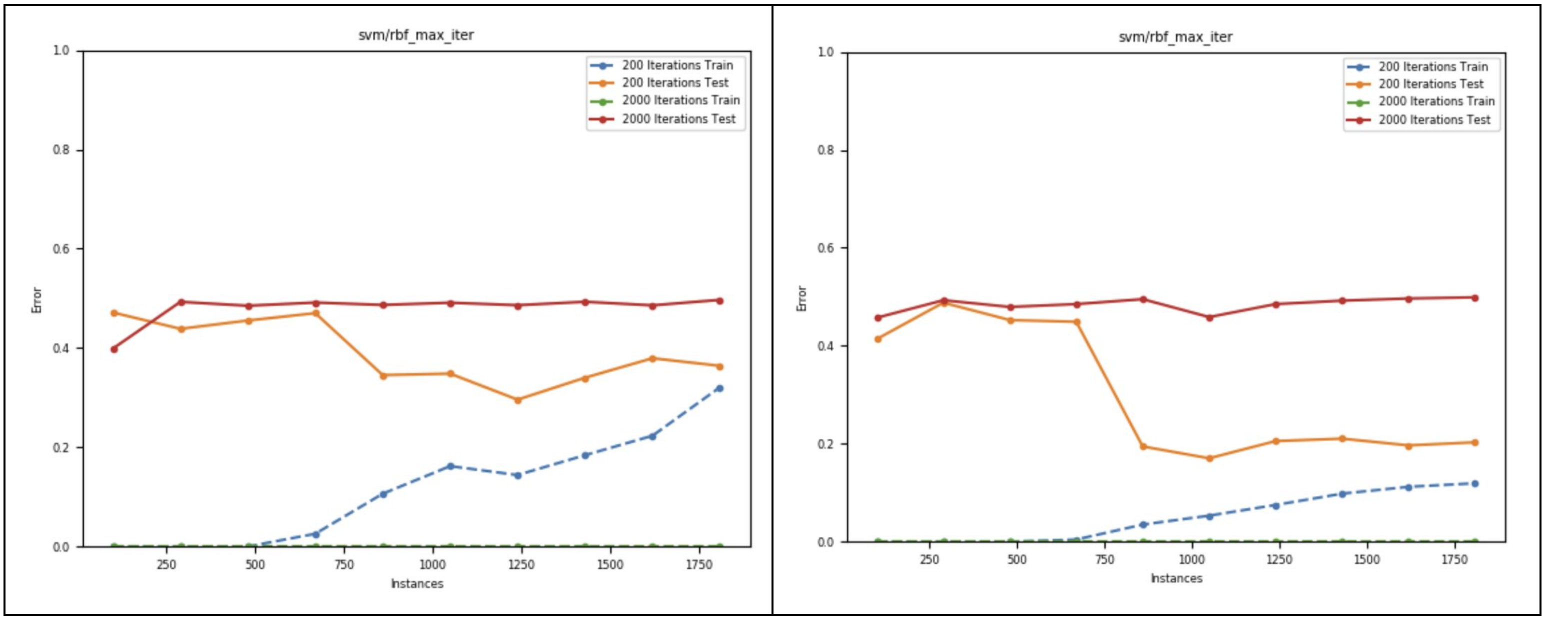
Boosting





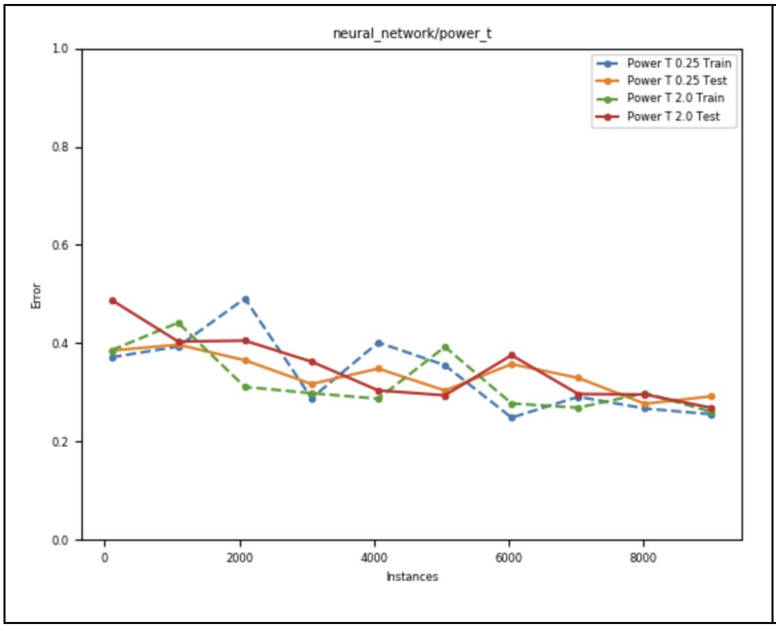
As decision trees performed better on the bank data than the EEG data, so did boosting with decision trees as the underlying weak learner.

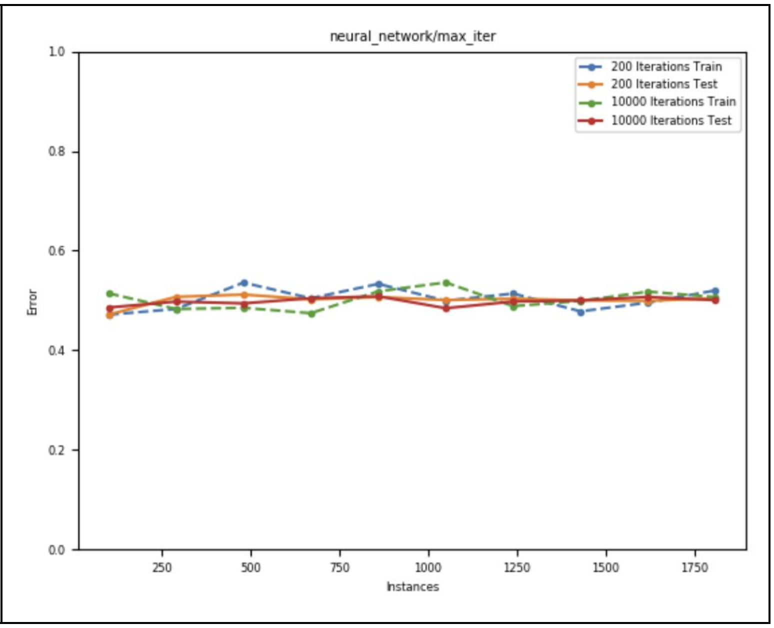
SVM



SVM was better able to separate the y classes for the continuous EEG data than the categorical bank data, which makes sense as the EEG values are related and are therefore more susceptible to being categorized by a continuous function. For both data sets, SVM overfit for too many iterations.

Neural Network





Neural networks did nothing for the EEG data but were able to classify the bank data samples a little bit better than random for a lot of data. I would have thought that neural networks would have performed better on the continuous, related (EEG) data than the categorically-separable bank data. Potentially, the neural network was able to create an accurate profile of the client contact information in the bank data set.

5. IMPLEMENTATION DETAILS

- To remove bias in the data, both data sets were down-sampled such that y was half true and half false. Therefore, random guessing always has a 0.5 probability of correct classification.
- Error is simply the fraction of y that was misclassified.
- Test error is the average error found during cross validation. Testing showed this value to be almost exactly the same as true test error, only with fewer outlier points. Therefore, cross validation error is show as test error in all of the graphs.
- Graphs that aren’t shown are not shown because of the 12 page limit.
- All classifiers were based on (“stolen”) from SciKit Learn. Classfiers used include sklearn.tree.DecisionTreeClassifier, sklearn.neighbors.KNeighborsClassifier, sklearn.ensemble.AdaBoostClassifier, sklearn.svm.SVC, sklearn.neural_network.MLPClassifier
- Unless otherwise specified, default parameters were used for all classifiers
- Adaptive learning rate and stochastic gradient descent were always used for neural network classifiers
- The RBF (radial basis function) and Polynomial kernels were used for the SVM classifiers
- For neural networks, many different activation functions were tried, all with the same results (e.g. sigmoid, tangent)