CS7641 - Machine Learning

Project 1 – Supervised Learning

I. Introduction

The motivation of this paper is to survey various supervised machine learning algorithms. These algorithms are: decision trees, neural networks, boosting, support vector machines, and k-nearest neighbors. The algorithms are to be applied to classification problems. We will be exploring the performance of these algorithms by looking into training vs. testing error on two different datasets.

The first data set records the physiochemical characteristics of red wine and a final dependent variable that is the quality rating of the wine. This dataset contains information on the wines' various types of acidities, amount of citric acid, residual sugar, chlorides, free and total sulfur oxide, density, pH (potential of hydrogen – a numeric scale used to specify the acidity of basicity of an aqueous solution), amount of sulphates, alcohol content, and it's quality as a score between 0 and 10.

The second data set is the result of shape feature extractors encoding the characteristics of 3D objects, in this case cars, within 2D images into 2D silhouettes. These characteristics are then represented as various features calculated based on measurements such as average perimeter, average radius, average distance from border, various ratios between maximum and minimum radii, perpendicular and horizontal lengths, major and minor axes, kurtosis about minor axis, kurtosis about major maxis, ratio between area of hollows and area of the bounding polygon, and its classification as being manufactured by being an opel, a saab (both car manufacturers), a bus, or a van.

In the following section, these two datasets will be described more in depth. It is to be noted that these data sets were not kept as they were and were, in fact, "massaged" to be more easily "digestible" by the algorithms. The manner in which they were massaged will be further described in the following section.

II. The Datasets

The first dataset I will talk about is the red wine data set containing 1598 observations and 12 features. I found this dataset interesting, first of all, because wine is a product, and the methodology behind examining this data could be extended to any product available for mass consumption. Second of all, because there's a mixture between quantitative, measurable, features as well as a final qualitative one., the wine's rating. This, in my opinion, lends itself to many kids of real world problems such as deciding whether or not a stock is good to invest in based on actual measurable features, such as its alpha, volatility, 30 day return, historical mean, etc.

Since the assignment was concerned with classification problems, and the original wine data set had a quality rating represented within the domain of integers, changes had to be made. I decided to, first of all, normalize all features to be in the [0, 1] range, and second of all, quality would be encoded into two classes: POOR and GOOD. Anything under a 0.5 would be considered a wine of POOR quality, while anything above is considered to be GOOD. This might be an over-simplification of the quality rating system, but since the second dataset naturally lent itself to multi-class (4 different classes, to be precise) problems, I really wanted to have a problem that was simply a binary classification problem since these are the ones that are easiest for beginner students of machine learning to "wrap their head around".

The second dataset was the vehicle silhouettes dataset containing 845 observations and 18 features. As I already mentioned, this was a multi-class dataset possessing 4 different classes. All other features were numeric in nature and were also normalized, same as with the previous dataset. I found this data set interesting because of its obvious relationship to problems in the fields of robotics, and also general computer vision. These applications are very relevant today, and I was interested in the way they encoded 3D objects within images into numeric features based on measurements representative of the actual object’s physical appearance rather than trying to encode it based on color, or other features more strongly related to the images rather than the physical object itself.

The end result of the vehicles dataset is simply a set of ratios of different dimensions of the shape of the vehicle and its final classification as a saab, opel, van, or a bus. I believe, since image recognition and deep learning are such huge topics, this would be an interesting introductory, yet relevant problem to examine.

The way the numeric values were normalized in both data sets was the same for both. It was a basic normalization consisting of:

$x_{norm}=\frac{x-x_{min}}{x_{max}-x_{min}}$ for all x , where x is the value of a particular feature (so each “row” is a vector of x’s), x_{min} is the smallest value present in the entire dataset for that particular feature (the smallest value found for that entire column), and x_{max} is the largest one.

In the next section, I will detail the performance of the five algorithms when being trained, then tested on these data sets. We will look at the confusion matrices, as well as training vs. testing errors and talk about possible explanations for the results observed.

III. The Algorithms

In order to tune the trees, grid-search was performed. The way this works, when training the model, several parameters are tested, and the ones that achieve the most accuracy are the ones that are kept in the end. The accuracy is determined by performing k-fold cross-validation for several trials. For all algorithms I used 10-fold cv for 3 trials. The plots are for accuracy, that is, the metric in the range [0, 1.0] indicative of the fraction of the examples the algorithm classifies correctly.

1. Decision Trees

The algorithm used for the decision trees portion of the survey was a more current version of the C4.5 Algorithm (Quinlan 1993) detailed in the book *Machine* Learning by Tom M. Mitchel, the

C5.0 algorithm. The original C4.5 algorithm by first allows over-fitting in training, then converts the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node, performs post-pruning of each rule by removing any preconditions that result in improving its estimated accuracy, then sorts the pruned rules by their estimated accuracy and considers them in this sequence when classifying subsequent instances (Mitchell). The improvements C5.0 has made over C4.5 are mostly an increase in speed (performance), automatic removal of unhelpful attributes (Pandya 2015), and more aggressive pruning, which results in less over-fitting and lower error rates.

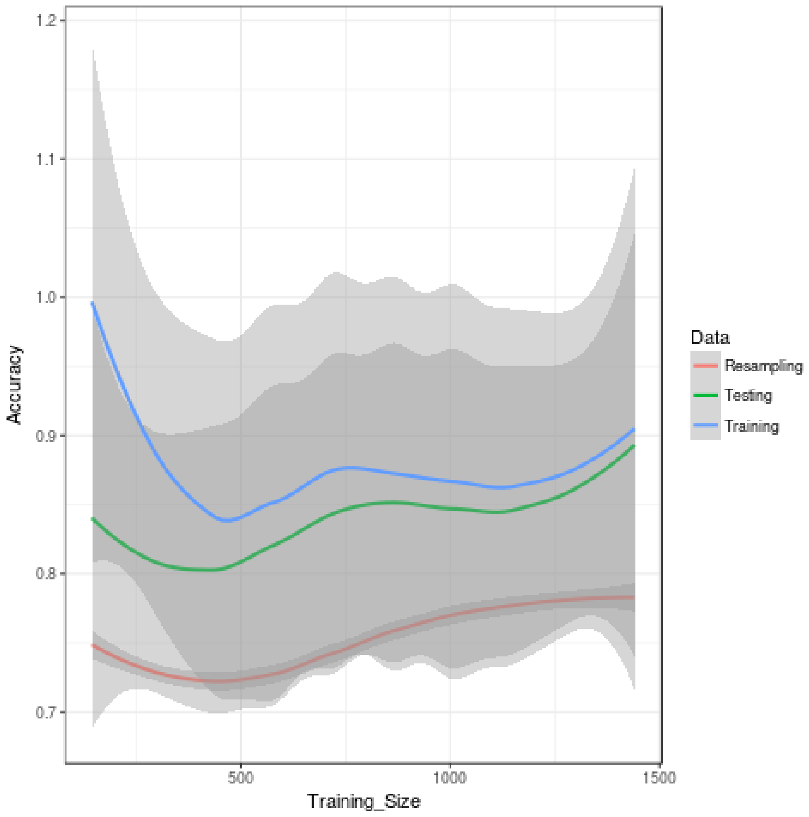

Figure 1. Wine data decision tree testing and training error vs. training data set size

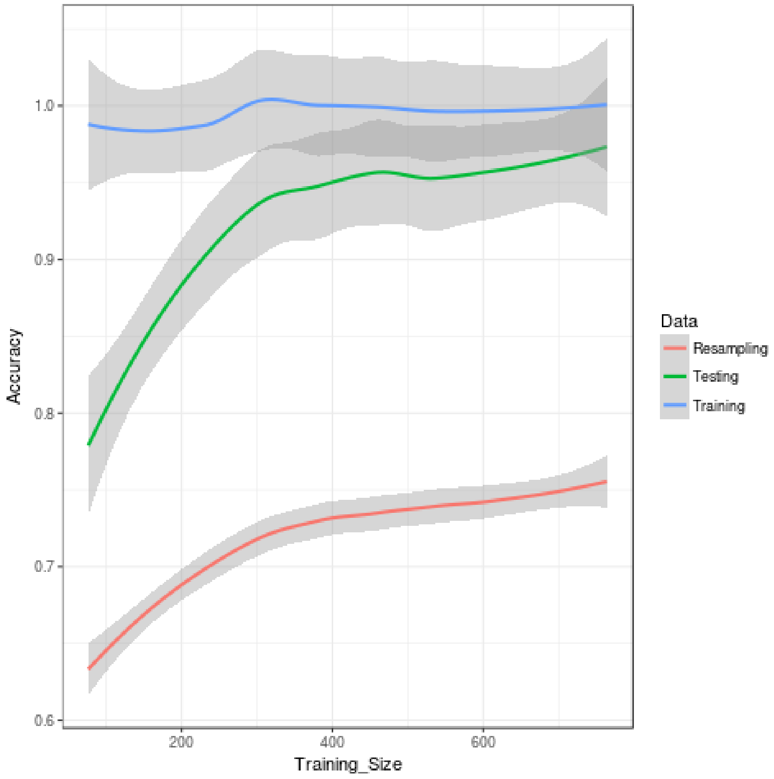

Figure 2. Vehicle data decision tree testing and training error vs training data set size

For the wine data set, a higher accuracy of 0.719 (vs. 0.716) was achieved by using winnowing (a form of selecting only the attributes with the highest information gain). As far as the vehicles data set goes, the best results were achieved by not using winnowing, 0.715 (vs. 0.689).

2. Neural Networks

A classic feed-forward neural network (using backpropagation/gradient descent) algorithm was used for the purposes of this survey of machine learning algorithms. Once again grid search was performed using 10 fold cross validation for 3 trials. For the size of the network, the range 1 to 20 hidden layers was tested, and for decay it was 0 to 1.0, where decay is a multiplier for the learning rate used to, hopefully, prevent getting stuck at local minima.

For the wine dataset grid search determined the best accuracy was obtained by having a 19 hidden layers and 0.1 as decay factor. As far as the vehicles dataset is concerned, grid search settled on a decay rate of 0.1 and a size of 20 hidden layers.

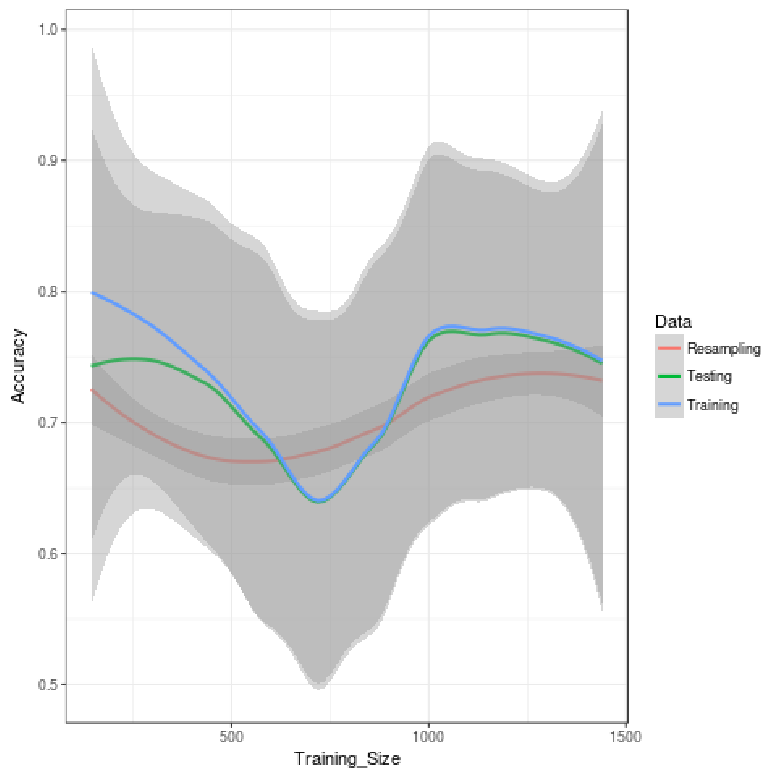

Figure 3. accuracy for neural network with the wine data set

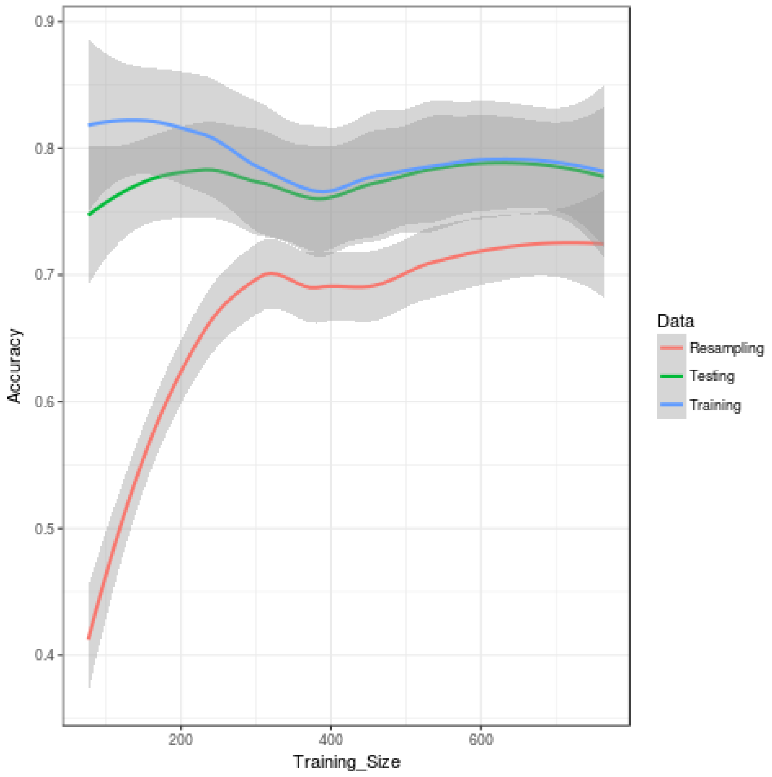

figure 4. accuracy for neural networks with the vehicles data set

The accuracy was decently high for both datasets, but not even reaching the 0.9 with either However, it is good to observe the testing accuracy being rather close to the training accuracy, meaning no obvious traces of over-fitting. Perhaps given more data, better performance would have been observed.

3. Boosting

The learner chosen for boosting was the C5.0 algorithm due to it’s high accuracy in testing. For boosting, Rob Schapire and Yoav Freund’s adaptive boosting (Adaboost) was used. The numbers of trials used were 1, 5, 10, 15, and 20. For the wine data set, the best parameters were found to be 20 trials and no winnowing for 0.80 accuracy (80%). For the vehicles data set, the best results were, once again, achieved by not using winnowing and using all 20 trials to achieve 0.743 accuracy (74.3%).

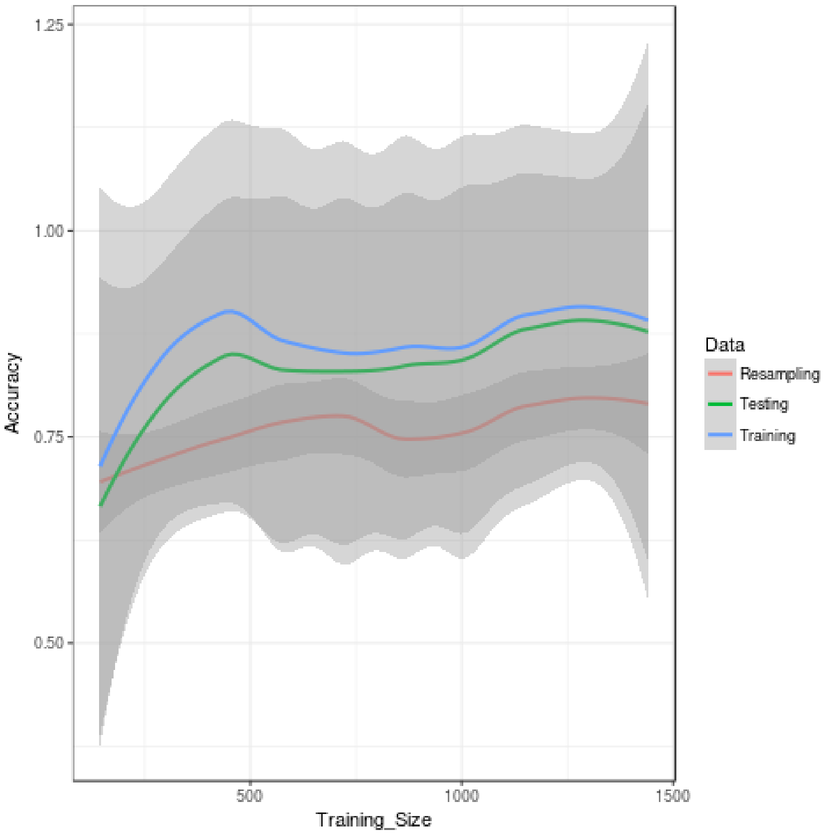

figure 5. boosting with C5.0 trees and the wine data set

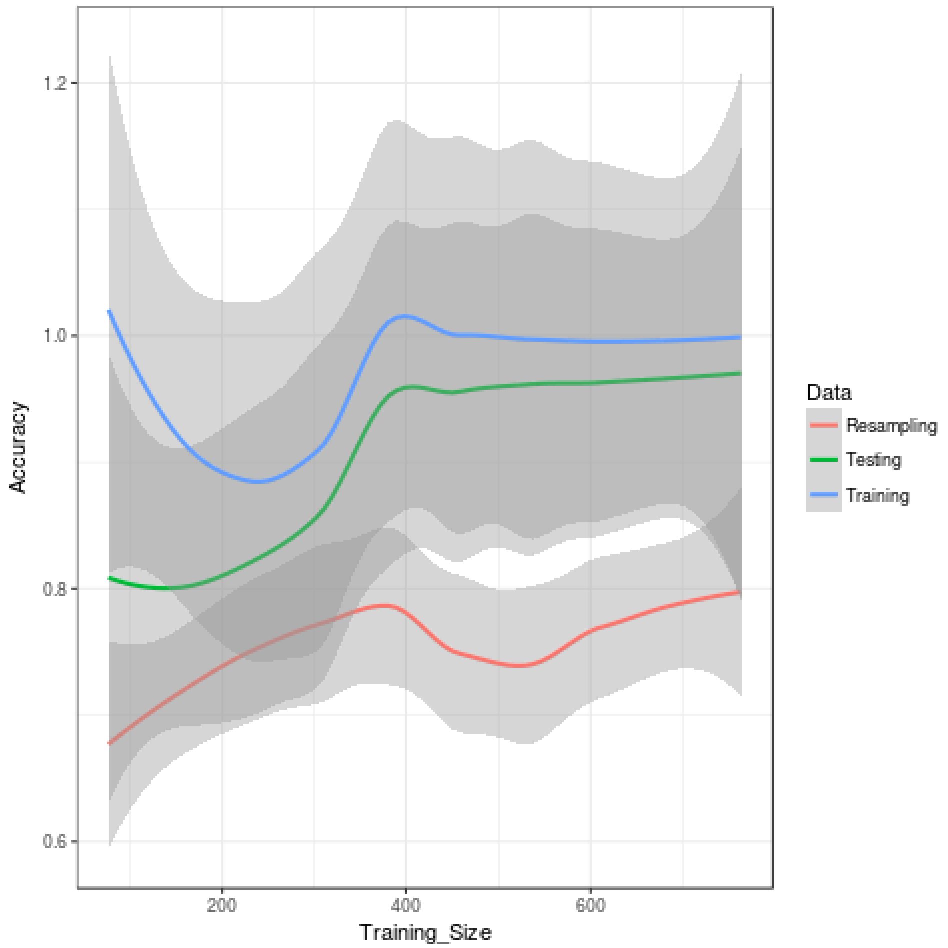

Figure 6. Boosting with C5.0 and the vehicles dataset

There was a noticeable improvement in accuracy, especially for the vehicles dataset.

4. Support Vector Machines

Support vector machines were examined using both radial basis function (RBF) and a linear function (LF) for kernels. For the wine dataset, the parameters used were, as determined best by grid search, 3.05E-5 for sigma and cost=1 with RBF, and a cost of 246 for the Linear kernel. For vehicles 9.53E-7 for sigma and cost=8 with RBF and a cost of 8 for the Linear kernel.

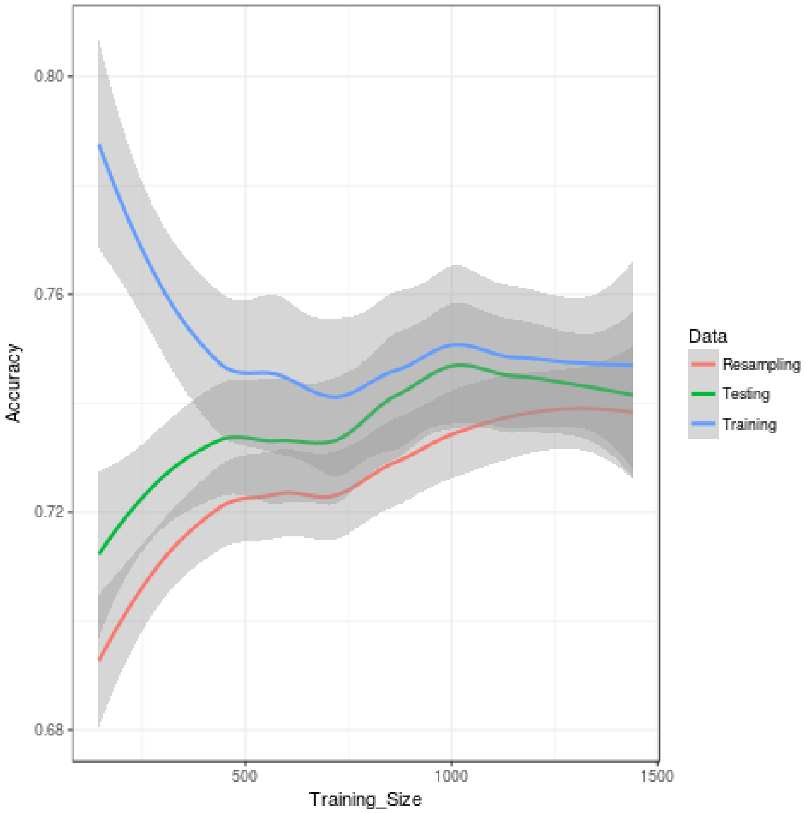

Figure 7. SVM with linear kernel applied to wine dataset

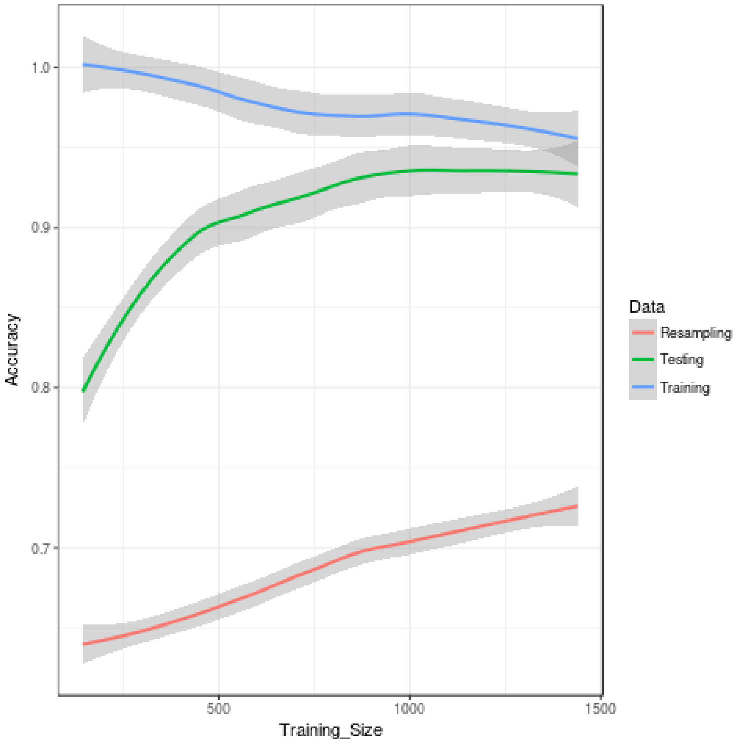

Figure 8. SVM with Radial basis function kernel applied to wine dataset

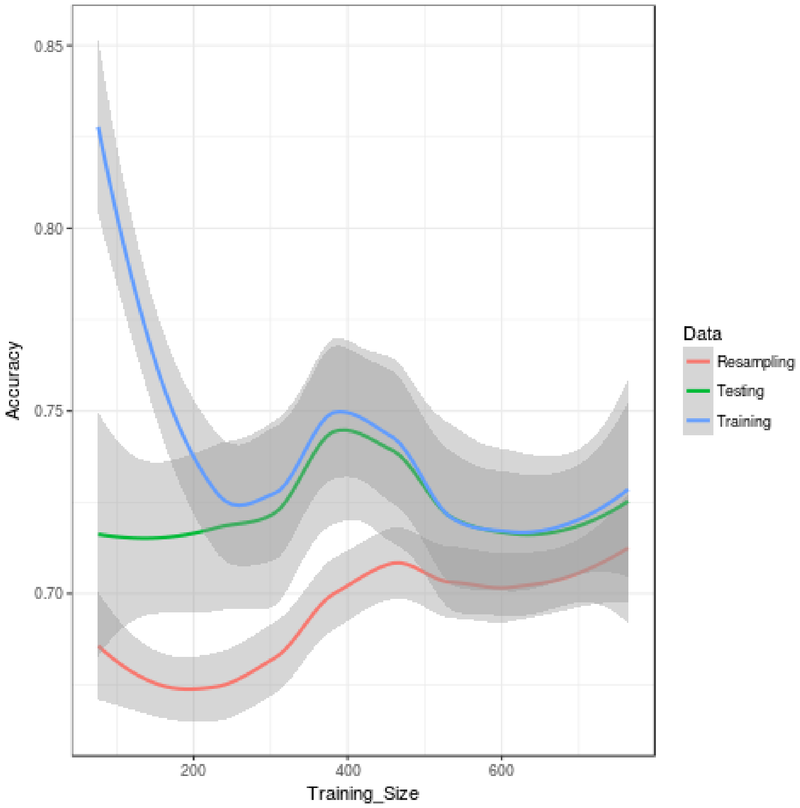

Figure 9. SVM with Linear function for kernel applied to vehicles dataset

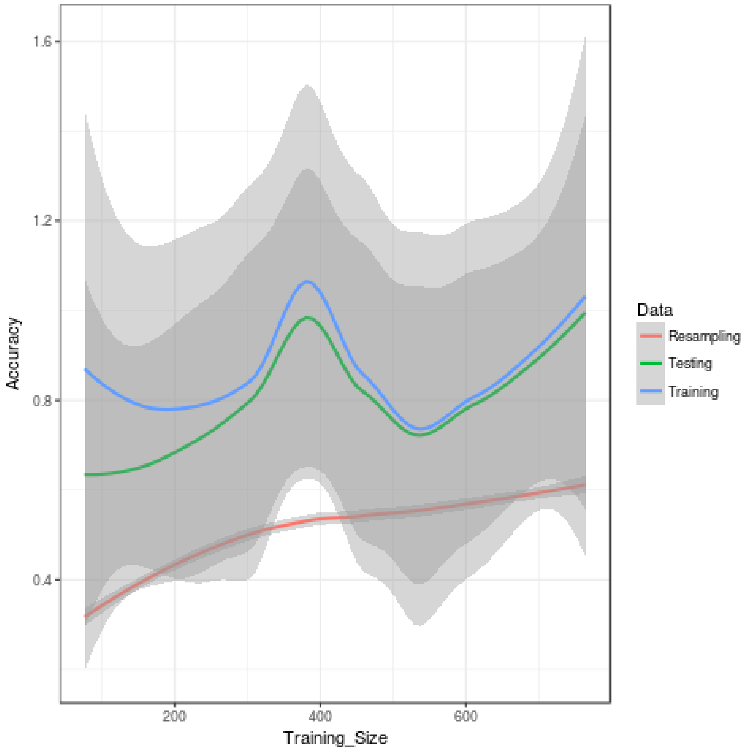

Figure 10. SVM with Radial Basis Function kernel applied to vehicles dataset

5. K-Nearest Neighbors

Once again grid search was performed to find the k that produced th most accuracy testing with k = [1, 20] with increments of 2. For the wine data set the optimal k=1 for an accuracy of 0.738. For the vehicles data set k=3, which produced 0.705 for accuracy.

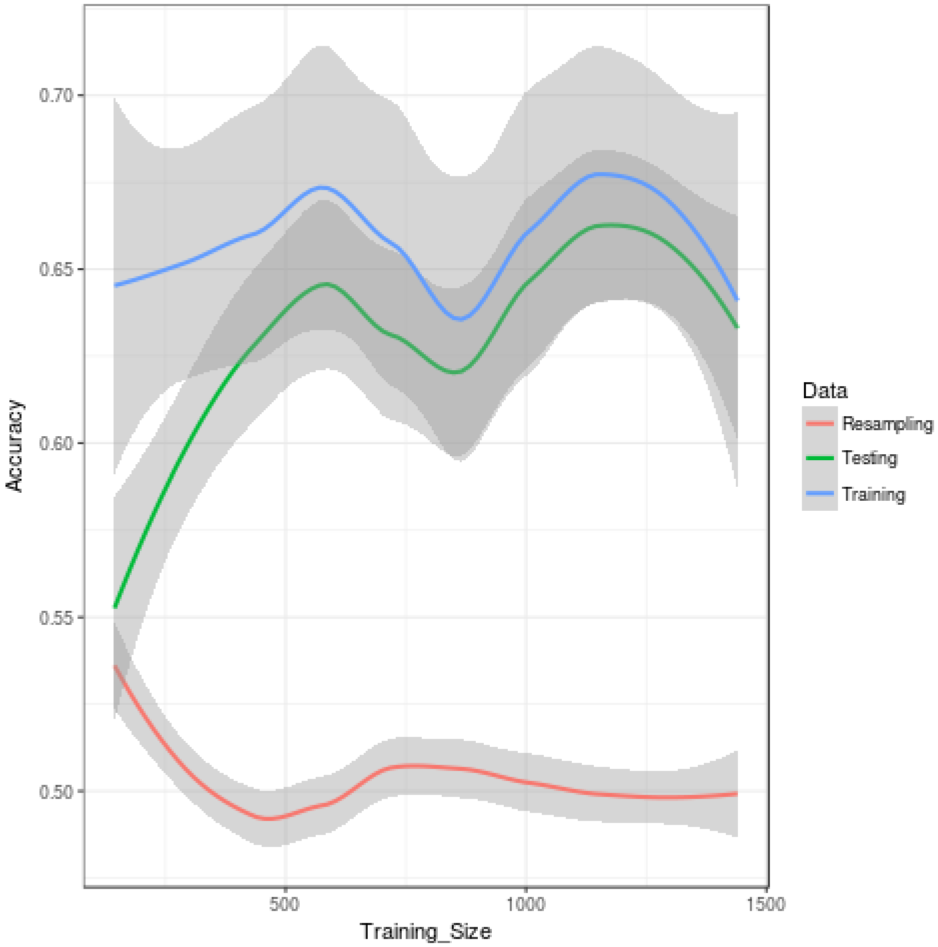

Figure 11. K-NN for the wine data set

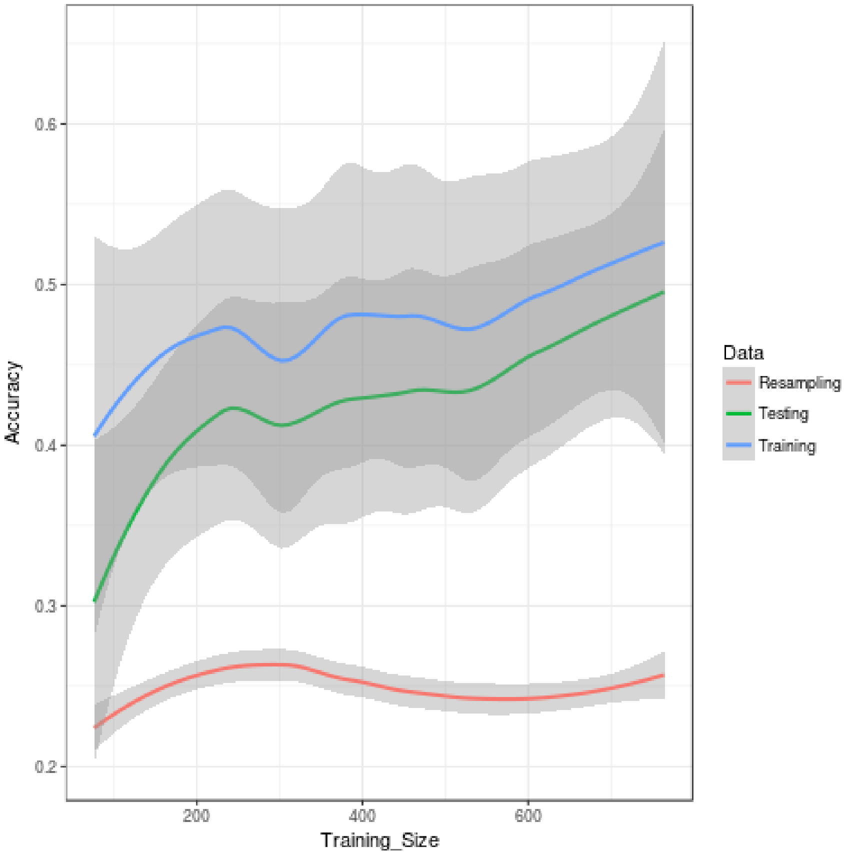

Figure 12. K-NN applied to the vehicles dataset

Conclusions

The wine dataset had a peculiar behavior because it was artificially induced to be a binary classification problem when in reality it would, perhaps, have benefited more from allowing for more classes (e.g. BAD, POOR, FAIR, GOOD), since the original quality rating system was on a scale of 1-10. Most wines would probably fall in the 5-6 range, being ok, with 1-4's and 7-10s being rarer and including some outliers. This may have resulted in the "POOR" [1-5) category being underrepresented in the data set compared to the GOOD category [5-10]. As can be seen by the superior performance with the C5.0 (both pure and boosted), and SVM with RBF kernel (but worst performance bar none with the Linear function kernel). Interestingly, for K-NN gridsearch yielded an N that was equal to the # of possible classes – 1. The vehicles dataset had similar results but overall performed better with higher accuracy with all algorithms.