Assignment 2 - KLVTHO001

November 17, 2017

1 Introduction

Many differnet learning algorithms are used in Statistical Sciences in order to classify hand-written digits. This project is focusing on a number of these learning approaches, namely Support Vector Machine, Artificial Neural Nework, Concolutional Neural Network, Random Forest, Bagging and Boosting. The dataset provided for this classification problem contains a response variable which indicated whether a digit is even or odd and 784 potential predictor variables which represents various digit properties. All the aforementioned approches are fit to the dataset and the best predictive algorithm is decided.

2 Theory

2.1 Tree Based Methods

Bagging, Boosting and Random Forests are all ensemble methods and meta learners. The key difference between Bagging and Boosting lies in how the two approches use the training set. Bagging is simply just bootstrap aggregation, which is all about choosing a random sample with replacement, train the algorithm on each sample seperately and average the predictions in the end. Furthermore, the key difference between Bagging and Random Forest is that Random Forest has the ability to improve variance by reducing the correlation between each tree in the forest. This is accomplished by randomly selecting a feature-subset for each split at each node. This is the main reason why Random Forests generally will generalize better as the number of trees grows. Furthermore, The key difference between these methods is that a Random Forest and Bagging is trained in parallel, i.e. each model is build independantly. In contrast, Boosting builts the models in a sequential way, making each model dependant on the previous ones.

2.2 Support Vector Machine

Support Vector Machine is a supervised binary classification algorithm. It attempts to find a hyperplane in a high dimensional space that can seperate the two classes of data by the largest margin. In order to achieve this, the Support Vector Machine will use a kernel to find the hyperplane that separates the data best.

2.3 Deep Learning

A neural network is within the class of Deep Learning and it is a supervised classifier. The network consists of several components interconnected and organized in layers. Each layer consists of neurons, which itself is a simple classifier. The input data is fed to the network and will pass through in a forward-feed manner. Furthermore, the training part is often done with the *Back Propagation* algorithm which helps find the optimal of each neuron between the layers of the network.

Convolutional Neural Networks are somewhat similar to the network described above. However, instead of learning single global weights matrix between two layers, this network is focusing on defining a set of locally connected nurons.

3 Method

3.1 Tree Based Methods

The `randomForest` and `gbm`-packages is used for training the Tree Based Methods. Furthermore, the models trained can be viewed in Table 1

Method	Number of trees	Mtry	Interaction Depth	Learning Rate	Bag Fraction	CV Folds
Random Forest	500					10
Bagging	500	32				10
Boosting	10.000		2	0.001	1	10

Table 1: Trained Models. If a field is empty, the default package settings are used.

Here, *Bag Fraction* is the fraction of the training set observations randomly selected to propose the next tree in the expansion, *Interaction Depth* is the maximum depth of variable interactions, *CV Folds* is the number of cross-validation folds to perform and *Mtry* is the number of variables randomly sampled as candidates at each split.

3.2 Support Vector Machine

A radial based kernel is used for the Support Vector Machine. In addition, Principal Component Analysis is performed in order to reduce the dimensionality of the problem. Furthermore, the `caret`-package in `R` is used to fit a model to the dataset. For the Support Vector Machine a number of different soft constraints where tested, thereof $C \in \{0, 0.5, 1, 1.5, 2, 3\}$

3.3 Deep Learning

3.3.1 Artificial Neural Network

For the Neural Network, the `h2o`-package is used in order to classify the digits. A number of different hyperparameters are tested using a grid and the best trained model is then used for further analysis. The trained models for the Neural Network the models can be seen Table 2.

Parameters	Value
Epochs	5, 10
Hidden	[512, 128], [218,42]
Rate	0.005, 0.01
Input Dropout Ratio	0.1
Nfolds	10
Stopping Rounds	3
Stopping Metric	Misclassification
Stopping Tolerance	0.02

Table 2: Trained Models

Where *Ephocs* is the number of times to iterate (stream) the dataset, *Hidden* is the hidden layer sizes, *Rate* is the the learning rate, *Input Droput Ratio* is specifying the input layer dropout ratio to improve generalization, $Nfolds$ is the the number of folds for cross-validation. Furthermore, the network stops training when misclassification rate, *Stopping Metric*, does not improve for the specified number of training rounds, based on a simple moving average. Lastly, the $Stopping\ Tolerance$ specifies the relative tolerance for the metric-based stopping to stop training if the improvement is less than this value.

3.3.2 Convolutional Neural Network

For the Convolutional Neural network the `mxnet`-package in `R` in order to predict the digits. A number of differnet model were trained and the most accurate model parameters can be seen in Table 3.

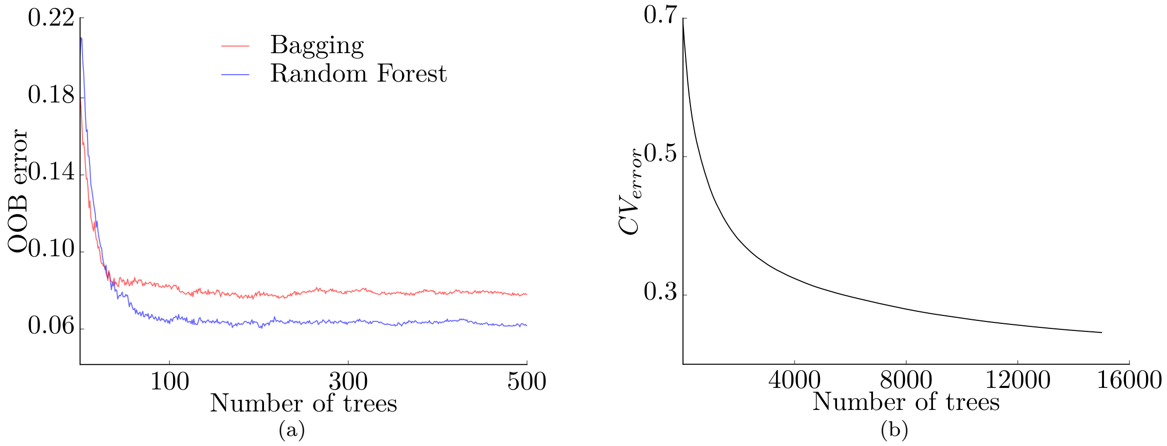
Parameters	Value
Num Rounds	50
Layers	2
Rate	0.01
Array Batch Size	40
Momentum	0.9
Stopping Metric	Misclassification
Activation	Sigmoid

Table 3: CNN Model

Where *Num Rounds* is the number of iterations over training data to train the model, *Layers* is the number of layers in the network, *Rate* is the learning rate, *Array Batch Size* is the batch size used for `R` array traininga and *Activation* is the activation function used.

4 Results

4.1 Tree Based Methods



(a)

(b)

Class	Even	Odd	Total	Error Rate
Even	222	17	239	0.034
Odd	20	241	261	0.04
Total	242	258	500	0.074

Table 4: Random Forest confusion matrix

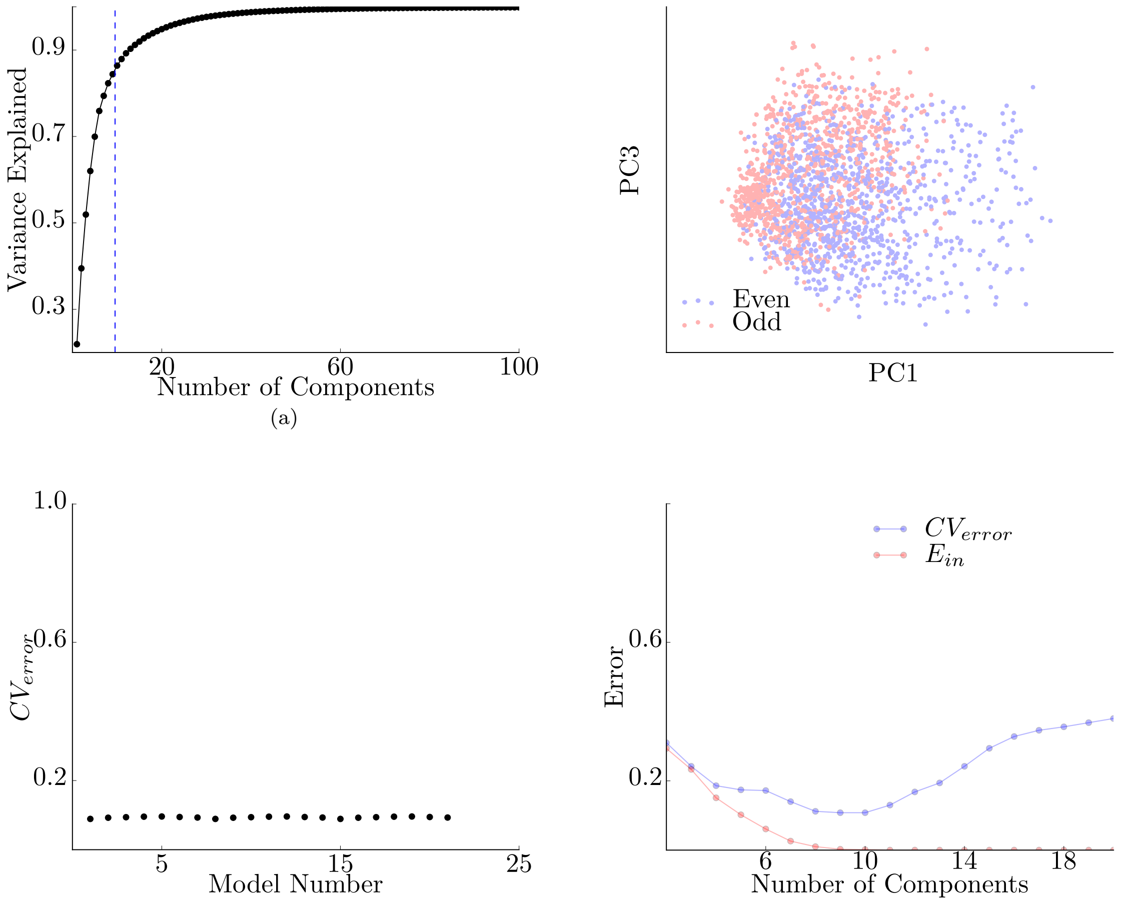
Class	Even	Odd	Total	Error Rate
Even	214	32	246	0.064
Odd	28	226	254	0.056
Total	242	258	500	0.12

Table 5: Bagging confusion matrix

Class	Even	Odd	Total	Error Rate
Even	216	26	242	0.052
Odd	26	232	258	0.052
Total	242	258	500	0.10

Table 6: Boosting confusion matrix

4.2 Support Vector Machines

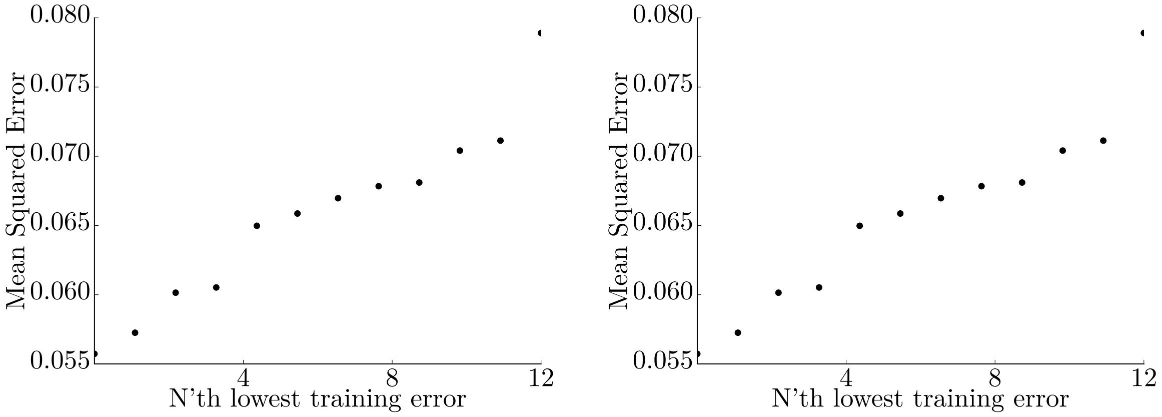


(a)

Class	Even	Odd	Total	Error Rate
Even	233	41	266	0.082
Odd	17	217	234	0.034
Total	242	258	500	0.11

Table 7: Support Vector Machine confusion matrix

4.3 Artificial Neural Network



(a) (b)

4.3.1 Convolutional Neural Network

A number of different models are were manually trained in order to obtain the best model. Table 8 shows the confusion matrix for the final model and it is observed that this model is able to correctly classify 97.8% of the digits in the test set, thus making this the most accurate prediction algorithm.

Class	Even	Odd	Total	Misclassification Rate
Even	234	8	242	0.0014
Odd	5	253	258	0.0018
Total	239	261	500	0.022

Table 8: Confusion matrix for Concolutional Neural Network

5 Discussion

6 Appendix