HW5 For Applied Data Mining
STAT W 3026-4026
Spring 2016
Columbia University

Robin Lee
rcl2136
QMSS MA

March 23, 2016

1 Instruction

For the Boston Housing Data, firstpartition the data as training (2/3) and testing (1/3). Then fit each the following models in table below (Linear Reg, Lasso, ElasticNet, PLS, Neural Networks, MARS, SVM, K-NN) and present the performance measures for both training and test data. Use cross validation to tune the parameters.

2 Result

		Training	Testing	
	RMSE	R2	RMSE	R2
Linear Regression	4.515	0.775	5.587	0.614
Lasso	4.528	0.769	5.613	0.609
ElasticNet	4.531	0.776	5.586	0.613
PLS	7.434	0.374	7.704	0.272
Neural Networks	23.578	NA	23.019	NA
MARS	3.108	0.888	4.718	0.735
SVM	3.667	0.855	4.566	0.744
K-NN	6.369	0.547	6.749	0.446

3 Step 1 - Create Data Partition

```
> library(caret)
> library(MASS)
> data("Boston")
> set.seed(569)
> train_index <- createDataPartition(Boston$medv, p = 2/3,
+                                    list = FALSE, times = 1)
> train <- Boston[train_index, ]
> test <- Boston[-train_index, ]
```

4 Set CV

```
> control <- trainControl( # 10 fold CV, repeated 10 times
+   method = 'repeatedcv', number = 10, repeats = 10
+
+ )
```

5 Linear Regression

```
> lm1 <- train(medv ~ . , data = train,
+              method = "lm",
+              trControl = control)
> lm1
> lm_test <- predict(lm1, test)
> postResample(lm_test, test$medv)
```

6 Lasso

```
> lasso1 <- train(medv ~ . , data = train,
+              method = "lasso",
+              trControl = control)
> lasso1
> lasso_test <- predict(lasso1, test)
> postResample(lasso_test, test$medv)
```

7 Elastic Net

```
> enet1 <- train(medv ~ . , data = train,
+              method = "enet",
+              trControl = control)
> enet1
> enet_test <- predict(enet1, test)
> postResample(enet_test, test$medv)
```

8 Partial Least Square

```
> pls1 <- train(medv ~ . , data = train,
+              method = "pls",
+              trControl = control)
> pls1
> pls_test <- predict(pls1, test)
> postResample(pls_test, test$medv)
```

9 Neural Net

```
> nnet1 <- train(medv ~ . , data = train,
+              method = "nnet",
+              trControl = control)
> nnet1
> nnet_test <- predict(nnet1, test)
> postResample(nnet_test, test$medv)
```

10 MARS

```
> mars1 <- train(medv ~ . , data = train,
+              method = "earth",
+              trControl = control)
> mars1

Multivariate Adaptive Regression Spline

338 samples
 13 predictors

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 305, 304, 303, 305, 306, 304, ...
Resampling results across tuning parameters:

  nprune  RMSE      Rsquared   RMSE SD    Rsquared SD
   2      5.757601  0.6157302  0.9035042  0.13191191
  11      3.369326  0.8704629  0.5822574  0.04774102
  20      3.109791  0.8895437  0.5068076  0.04044185

Tuning parameter 'degree' was held constant at a value of 1
RMSE was used to select the optimal model using  the smallest value.
The final values used for the model were nprune = 20 and degree = 1.

> mars_test <- predict(mars1, test)
> postResample(mars_test, test$medv)
```

```
     RMSE  Rsquared
4.7180742 0.7359379
```

11 SVM

```
> svm1 <- train(medv ~ . , data = train,
+             method = "svmRadial",
+             trControl = control)
> svm1
> svm_test <- predict(svm1, test)
> postResample(svm_test, test$medv)
```

12 K Nearest Neighbor

```
> knn1 <- train(medv ~ . , data = train,
+             method = "knn",
+             trControl = control)
> knn1
> knn_test <- predict(knn1, test)
> postResample(knn_test, test$medv)
```

13 Question 2

The lm.ridge function would not fit age twice. I created an age2 variable that is the same as age. Fitting it twice age with lm.ridge does not give the expected coefficient . It is not one half of the original coefficient.

```
> ridgem <- lm.ridge(medv ~ age + age + . , data = train)
> ridgem
```

```
                    age          crim            zn         indus          chas
 19.48390148  -0.01368204  -0.09892445   0.03403269   0.03776729   2.82505346
         nox           rm           dis          rad          tax      ptratio
-13.88439925   5.31382481  -1.24210651   0.26774200  -0.01342864  -0.76619515
       black        lstat
  0.01223820  -0.39446523
```

```
> train_add <- train
> train_add$age2 <- train_add$age
> ridgem2 <- lm.ridge(medv ~ age +., data = train_add)
> ridgem2

                        age          crim            zn         indus
 2.435153e+01 -8.834433e+13 -1.223741e-01  6.734561e-03  1.340861e-01
         chas           nox            rm           dis           rad
 3.968855e+00 -1.671082e+01  5.280787e+00 -1.264933e+00  2.339035e-01
          tax       ptratio         black         lstat          age2
-1.342864e-02 -7.661951e-01  1.125007e-02 -3.944652e-01  8.834433e+13

>
>
>
```