Quiz 4

5/5 points
earned (100%)

Quiz passed!

Continue Course (/learn/practical-machine-learning/supplement/PvInj/course-project-instructions-read-first)

Back to Week 4 (/learn/practical-machine-learning/home/week/4)

---

1 / 1
points

1.
For this quiz we will be using several R packages. R package versions change over time, the right answers have been checked using the following versions of the packages.

AppliedPredictiveModeling: v1.1.6

caret: v6.0.47

ElemStatLearn: v2012.04-0

pgmm: v1.1

rpart: v4.1.8

gbm: v2.1

lubridate: v1.3.3

forecast: v5.6

e1071: v1.6.4

If you aren't using these versions of the packages, your answers may not exactly match the right answer, but hopefully should be close.

Load the vowel.train and vowel.test data sets:

-code--code-library(ElemStatLearn)

data(vowel.train)

data(vowel.test)

-/code--/code-

Set the variable y to be a factor variable in both the training and test set. Then set the seed to 33833. Fit (1) a random forest predictor relating the factor variable y to the remaining variables and (2) a boosted predictor using the "gbm" method. Fit these both with the train() command in the caret package.

What are the accuracies for the two approaches on the test data set? What is the accuracy among the test set samples where the two methods agree?

RF Accuracy = 0.9987

GBM Accuracy = 0.5152

Agreement Accuracy = 0.9985

RF Accuracy = 0.6082

GBM Accuracy = 0.5152

Agreement Accuracy = 0.5152

RF Accuracy = 0.9881

GBM Accuracy = 0.8371

Agreement Accuracy = 0.9983

RF Accuracy = 0.6082

GBM Accuracy = 0.5152

Agreement Accuracy = 0.6361

Correct Response

1 / 1
points

2.
Load the Alzheimer's data using the following commands

-code-library(caret)

library(gbm)

set.seed(3433)

library(AppliedPredictiveModeling)

data(AlzheimerDisease)

adData = data.frame(diagnosis,predictors)

inTrain = createDataPartition(adData$diagnosis, p = 3/4)[[1]]

training = adData[inTrain,]

testing = adData[-inTrain,]

-/code-

Set the seed to 62433 and predict diagnosis with all the other variables using a random forest ("rf"), boosted trees ("gbm") and linear discriminant analysis ("lda") model. Stack the predictions together using random forests ("rf"). What is the resulting accuracy on the test set? Is it better or worse than each of the individual predictions?

Stacked Accuracy: 0.80 is better than random forests and lda and the same as boosting.

Correct Response

- Stacked Accuracy: 0.80 is better than all three other methods
- Stacked Accuracy: 0.88 is better than all three other methods
- Stacked Accuracy: 0.80 is worse than all the other methods.

---

1 / 1 points

3.

Load the concrete data with the commands:

-code-

```
set.seed(3523)

library(AppliedPredictiveModeling)

data(concrete)

inTrain = createDataPartition(concrete$CompressiveStrength, p = 3/4)[[1]]

training = concrete[ inTrain,]

testing = concrete[-inTrain,]
```

-/code-

Set the seed to 233 and fit a lasso model to predict Compressive Strength. Which variable is the last coefficient to be set to zero as the penalty increases? (Hint: it may be useful to look up ?plot.enet).

- Water
- Age
- Cement

Correct Response

- CoarseAggregate

1 / 1
points

4.
Load the data on the number of visitors to the instructors blog from here:

https://d396qusza40orc.cloudfront.net/predmachlearn/gaData.csv
(https://d396qusza40orc.cloudfront.net/predmachlearn/gaData.csv)

Using the commands:

-code-library(lubridate) # For year() function below

dat = read.csv("~/Desktop/gaData.csv")

training = dat[year(dat$date) < 2012,]

testing = dat[(year(dat$date)) > 2011,]

tstrain = ts(training$visitsTumblr)

-/code-

Fit a model using the bats() function in the forecast package to the training time series. Then forecast this model for the remaining time points. For how many of the testing points is the true value within the 95% prediction interval bounds?

96%

Correct Response

- 94%
- 100%
- 92%

---

1 / 1 points

5.

Load the concrete data with the commands:

-code-

```
set.seed(3523)

library(AppliedPredictiveModeling)

data(concrete)

inTrain = createDataPartition(concrete$CompressiveStrength, p = 3/4)[[1]]

training = concrete[ inTrain,]

testing = concrete[-inTrain,]
```

-/code-

Set the seed to 325 and fit a support vector machine using the e1071 package to predict Compressive Strength using the default settings. Predict on the testing set. What is the RMSE?

- [x] 6.72

Correct Response

- [ ] 6.93
- [ ] 45.09
- [ ] 11543.39