Practical 2

Jumping Rivers

Cross validation

- Fit a linear regression model to the `cars2010` data set with `FE` as the response, using `EngDispl`, `NumCyl` and `NumGears` as predictors. Load the data like so

```
data("FuelEconomy", package = "jrPred")
```

```
mLM = train(FE~EngDispl+NumCyl+NumGears, method = "lm", data = cars2010)
```

- What is the training error rate (RMSE) for this model? Hint: The training error can be found by taking the square root of the average square residuals. The `sqrt` and `resid` functions may be useful.

```
res = resid(mLM)
(trainRMSE = sqrt(mean(res*res)))
```

```
## [1] 4.589728
```

- We can re-train the model using the validation set approach to estimate a test RMSE. That is, splitting up the data into a training and testing set.

```
## pick an index for samples
trainIndex = createDataPartition(cars2010$FE, p = 0.5, list = FALSE)
## set up validation set approach
tcVS =  trainControl(method = "cv",
                     number = 1,
                     index = list(Fold1 = trainIndex))

mLMVS = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
              data = cars2010, trControl = tcVS)
getTrainPerf(mLMVS)
```

```
##   TrainRMSE TrainRsquared TrainMAE method
## 1  4.702351     0.6017011 3.543246     lm
```

Doing it this way we can see that the RMSE is larger. Validation set estimates tend to over estimate the test RMSE as models perform worse with less data. Furthermore, if we ran this code again, we'd get a completely different test RMSE estimate. For validation set estimates, the test RMSE is highly variable depending on which observations are chosen for each group.

- We can use k-fold cross validation to solve these two issues. Re-run this model but using 10-fold cross-validation instead.

```
# set up train control objects
tcKFOLD = trainControl(method = "cv", number = 10)
# run model
mLMKFOLD = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tcKFOLD)
```

- How do these estimates compare with the validation set approach?

```
getTrainPerf(mLMVS)
```

```
##   TrainRMSE TrainRsquared TrainMAE method
## 1  4.702351     0.6017011 3.543246     lm
```

```
getTrainPerf(mLMKFOLD)
```

```
##   TrainRMSE TrainRsquared TrainMAE method
## 1  4.597432     0.6302621 3.498251     lm
```

```
# 10-fold is lower than validation set, we mentioned it tended to
# over estimate test error
```

- The object returned by `train` also contains timing information that can be accessed via the `times` component of the list. Which of the methods is fastest?
  Hint: The `$` notation can be used pick a single list component.

```
mLMVS$times$everything
```

```
##    user  system elapsed
##   0.432   0.000   0.432
```

```
mLMKFOLD$times$everything
```

```
##    user  system elapsed
##   0.489   0.004   0.493
```

- Using k-fold cross validation to estimate test error investigate how the number of folds effects the resultant estimates and computation time.

```
# a number of trainControl objects
tc2 = trainControl(method = "cv", number = 2)
tc5 = trainControl(method = "cv", number = 5)
tc10 = trainControl(method = "cv", number = 10)
tc15 = trainControl(method = "cv", number = 15)
tc20 = trainControl(method = "cv", number = 20)
# train the model using each
mLM2 = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tc2)
mLM5 = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tc5)
mLM10 = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tc10)
mLM15 = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tc15)
mLM20 = train(FE~EngDispl+NumCyl+NumGears, method = "lm",
    data = cars2010, trControl = tc20)
# use a data frame to store all of the relevant information
(info = data.frame("Folds" = c(2,5,10,15,20),
    "Time" = c(mLM2$times$everything[1],
        mLM5$times$everything[1],
        mLM10$times$everything[1],
        mLM15$times$everything[1],
        mLM20$times$everything[1]),
    "Estimate" = c(mLM2$results$RMSE,
                   mLM5$results$RMSE,
                   mLM10$results$RMSE,
                   mLM15$results$RMSE,
                   mLM20$results$RMSE)))
```

```
##   Folds  Time Estimate
## 1     2 0.434 4.583831
## 2     5 0.445 4.598761
```

```
## 3    10 0.467 4.586105
## 4    15 0.490 4.571732
## 5    20 0.517 4.564702
```

```
# as there are more folds it takes longer to compute,
# not an issue with such a small model but something
# to consider on more complicated models.
# Estimates are going down as the number of folds increases.
# This is because for each held out fold we are using a greater
# proportion of the data in training so expect to get a better
# model.
```

- Experiment with adding terms to the model, transformations of the predictors and interactions say and use cross validation to estimate test error for each. What is the best model you can find?