Logistic Regression Tutorial

A fundamental understanding of logistic regression models is assumed, please seek resources to improve understanding and use this tutorial as a computational example.

Read data

More information can be found in the Data Import and Export tutorial notebook.

In [1]:
```
train = read.csv("train.csv")
test = read.csv("test.csv")
```

Error Metric ¶

R-Bloggers has a nice article on explaining the Log Loss function which can be read here (http://www.r-bloggers.com/making-sense-of-logarithmic-loss/).

In [2]:
```
LogLossBinary = function(actual, predicted, eps = 1e-15) {
    predicted = pmin(pmax(predicted, eps), 1-eps)
    - (sum(actual * log(predicted) + (1 - actual) * log(1 - predicted)))
/ length(actual)
}
```

One Predictor Model

Fit a single predictor logistic regression model and inspect its coefficients.

In [3]:
```
model = glm(Response ~ Var1, data = train, family = "binomial")
```

In [4]:
```
coef(model)
```

Out[4]:

(Intercept)	-1.01342180494271
Var1	-0.228121321366388

In [5]:
```
summary(model)
```

Out[5]:
```
Call:
glm(formula = Response ~ Var1, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9143  -0.8267  -0.7963   1.5474   1.7307

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.013422   0.008732 -116.06   <2e-16 ***
Var1        -0.228121   0.015172  -15.04   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 118824  on 99999  degrees of freedom
Residual deviance: 118595  on 99998  degrees of freedom
AIC: 118599

Number of Fisher Scoring iterations: 4
```

Make predictions with the single predictor model on the training data.

In [6]:
```
trainingPredictions = predict(model, type = "response")
```

In [7]:
```
LogLossBinary(train$Response, trainingPredictions)
```

Out[7]: 0.59297692097586

Create another model that predicts every observation as the mean of the training set's response variable. Compare its log loss to the first model's log loss.

In [8]:
```
responseMean = rep(mean(train$Response), nrow(train))
```

In [9]:
```
LogLossBinary(train$Response, responseMean)
```

Out[9]: 0.594120639781108

In [10]:
```
LogLossBinary(train$Response, trainingPredictions)
```

Out[10]: 0.59297692097586

Three Predictor Model

Fit another logistic regression model with more predictors and inspect results.

In [11]:
```
multipleModel = glm(Response ~ Var1 + Var2 + NVVar1, data = train, famil
y = "binomial")
```

In [12]:
```
coef(multipleModel)
```

Out[12]:

(Intercept)	-0.990706038509847
Var1	-0.0170514509093614
Var2	-0.198554853852551
NVVar1	0.0421376519120142

In [13]:
```
summary(multipleModel)
```

Out[13]:
```
Call:
glm(formula = Response ~ Var1 + Var2 + NVVar1, family = "binomial",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0335  -0.8382  -0.7782   1.5042   2.2484

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -0.990706   0.008792 -112.681  < 2e-16 ***
Var1        -0.017051   0.019062   -0.895    0.371
Var2        -0.198555   0.011143  -17.819  < 2e-16 ***
NVVar1       0.042138   0.006965    6.050 1.45e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 118824  on 99999  degrees of freedom
Residual deviance: 118231  on 99996  degrees of freedom
AIC: 118239

Number of Fisher Scoring iterations: 4
```

In [14]:
```
multiplePredictions = predict(multipleModel, type = "response")
```

Compare the log loss of all 3 models fit.

In [15]:
```
print(paste(LogLossBinary(train$Response, responseMean), "Log loss of re
sponse mean model"))
print(paste(LogLossBinary(train$Response, trainingPredictions), "Log los
s of single predictor model"))
print(paste(LogLossBinary(train$Response, multiplePredictions), "Log los
s of multiple predictor model"))
```

```
[1] "0.594120639781108 Log loss of response mean model"
[1] "0.59297692097586 Log loss of single predictor model"
[1] "0.591155408804644 Log loss of multiple predictor model"
```

As expected, the single predictor logistic regression model has a lower log loss than the response mean model, and the multiple predictor logistic regression model has a lower log loss than both.

Create baseline submission

Create predictions on the test set using our three predictor model. These predictions are scored as the 'GLM Benchmark' on the competition leaderboard.

In [16]:
```
testPredictions = predict(multipleModel, newdata = test, type = "respons
e")
```

In [17]:
```
outputDataSet = data.frame("RowID" = test$RowID,
                           "ProbabilityOfResponse" = testPredictions)
```

In [18]:
```
write.csv(outputDataSet, "glmBenchmarkSubmission.csv", row.names = FALS
E)
```