Yan Zhang's Logistic Regression Movie Prediction Project

Yan Zhang

December 1, 2015

For this project, I will use data set Film fit with three logistic model in order to see if British movies and other variables can predict if the movie is good. The dataset has 100 movie reviews so there are 100 data for each variables. The range for Cast is from 3 to 13; the range for Rating is from 1 to 4 in crements of 0.5; the range Description is 5 to 21 and the Origin 0 means USA, 1 means Great Britain, 2 means France, 3 means Italy and 4 means Canada.

First of all, let's read the data set from library Stat2Data first:

```
library(Stat2Data)
data(Film)
head(Film)
```

```
##                          Title Year Time Cast Rating Description Origin
## 1            A_Ticklish_Affair 1963   89    5    2.0           7      0
## 2 Action_in_the_North_Atlantic 1943  127    7    3.0           9      0
## 3        And_the_Ship_Sails_On 1984  138    7    3.0          15      3
## 4                Autumn_Sonata 1978   97    5    3.0          11      5
## 5           Bachelor_Apartment 1931   77    6    2.5           7      0
## 6           Benson_Murder_Case 1930   69    8    2.5          10      0
##   Time_code Good
## 1     short    0
## 2      long    1
## 3      long    1
## 4      long    1
## 5     short    0
## 6     short    0
```

```
n = nrow(Film)
```

Next, we only care if British movies affect movie rating so the next part of code is to create a new column called British only contains British and Other:

```
attach(Film)
British = NULL #create the new variable
for(i in 1:n){
  if (Origin[i] == 1){
    British[i] = "British"
  }
  else{
    British[i] = "Other"
  }
}
```

```
New.Film = data.frame(Film, British)
head(New.Film)
```

```
##                           Title Year Time Cast Rating Description Origin
## 1             A_Ticklish_Affair 1963   89    5    2.0           7      0
## 2 Action_in_the_North_Atlantic 1943  127    7    3.0           9      0
## 3         And_the_Ship_Sails_On 1984  138    7    3.0          15      3
## 4                 Autumn_Sonata 1978   97    5    3.0          11      5
## 5            Bachelor_Apartment 1931   77    6    2.5           7      0
## 6            Benson_Murder_Case 1930   69    8    2.5          10      0
##   Time_code Good British
## 1     short    0   Other
## 2      long    1   Other
## 3      long    1   Other
## 4      long    1   Other
## 5     short    0   Other
## 6     short    0   Other
```

Model 1: Do chi-square test: Good~British:

First create a contigency tabe:

```
table1 = table(New.Film$British, New.Film$Good)
table1
```

```
##
##             0  1
##   British  9  4
##   Other   60 27
```

```
chisq.test(table1)
```

```
## Warning in chisq.test(table1): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table1
## X-squared = 3.6531e-31, df = 1, p-value = 1
```

From the table above, we can see this data set looks random and independent but there is only 4 sample in one cell, which doesn't meet the assumption for Chi-square test, so the reault of the test may be unreliable.

df= (a-1)(b-1) a=2, b=2 so df = 1 X-squared vaule is very small so the p value is 1, so base on our test, those two variables are independent with high posibility since this test desn't meet assumption.

Althogh one cell has sample size 4 is acceptable but in case I will also do a Fisher test:

```
fisher.test(table1)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table1
## p-value = 1
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2540726 4.8993590
## sample estimates:
## odds ratio
##   1.012375
```

The p value for Fisher test is also 1, so we can see base on our test, those two variables are independent.

Model 2: create a logistic model: Good~British

```
model2=glm(Good~British, data = New.Film, family = binomial(link = "logit"))
summary(model2)
```

```
##
## Call:
## glm(formula = Good ~ British, family = binomial(link = "logit"),
##     data = New.Film)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.862  -0.862  -0.862   1.530   1.535
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.81093    0.60092  -1.349    0.177
## BritishOther  0.01242    0.64406   0.019    0.985
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.82  on 99  degrees of freedom
## Residual deviance: 123.82  on 98  degrees of freedom
## AIC: 127.82
##
## Number of Fisher Scoring iterations: 4
```

We are using z test in here, both coefficiing are less to one and both p value are greater than 0.05, so British doesn't predict good prediction for variable Good, which also prove the result we got from the first model. These results are consistent, in model 1, there is no association between the two variables, and in model 2, British is not a good predictor of Good.

Model 3: Use model selection to find the best model for predicting Good variable. Start from simplest model need use British in the model. Complex model include a 5 way interaction.

There are five variables in this model so I need to see if there's 5, 4, 3 and 2 interactions and if these interactions are good predictors for this model.

```
#the complex model:

model3w4 = glm(Good~(British+Year+Time+Cast+Description)^4, data = New.Film,
family = binomial(link = "logit"))
summary(model3w4)

##
## Call:
## glm(formula = Good ~ (British + Year + Time + Cast + Description)^4,
##     family = binomial(link = "logit"), data = New.Film)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.56517  -0.57032  -0.21511   0.05376   2.23900
##
## Coefficients: (2 not defined because of singularities)
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -6.438e+05  6.906e+09   0.000   0.9999
## BritishOther                     6.324e+05  6.906e+09   0.000   0.9999
## Year                             3.269e+02  3.502e+06   0.000   0.9999
## Time                             6.228e+03  7.206e+07   0.000   0.9999
## Cast                             5.521e+04  4.132e+08   0.000   0.9999
## Description                      3.964e+04  6.401e+08   0.000   1.0000
## BritishOther:Year               -3.211e+02  3.502e+06   0.000   0.9999
## BritishOther:Time               -6.115e+03  7.206e+07   0.000   0.9999
## BritishOther:Cast               -5.338e+04  4.132e+08   0.000   0.9999
## BritishOther:Description        -3.828e+04  6.401e+08   0.000   1.0000
## Year:Time                       -3.150e+00  3.637e+04   0.000   0.9999
## Year:Cast                       -2.824e+01  2.122e+05   0.000   0.9999
## Year:Description                -2.011e+01  3.250e+05   0.000   1.0000
## Time:Cast                       -4.492e+02  3.454e+06   0.000   0.9999
## Time:Description                -4.295e+02  7.287e+06   0.000   1.0000
## Cast:Description                -1.742e+02  6.259e+05   0.000   0.9998
## BritishOther:Year:Time           3.092e+00  3.637e+04   0.000   0.9999
## BritishOther:Year:Cast           2.731e+01  2.122e+05   0.000   0.9999
## BritishOther:Year:Description    1.942e+01  3.250e+05   0.000   1.0000
## BritishOther:Time:Cast           4.309e+02  3.454e+06   0.000   0.9999
## BritishOther:Time:Description    4.159e+02  7.287e+06   0.000   1.0000
## BritishOther:Cast:Description   -3.624e+01  6.259e+05   0.000   1.0000
## Year:Time:Cast                   2.281e-01  1.749e+03   0.000   0.9999
## Year:Time:Description            2.168e-01  3.683e+03   0.000   1.0000
## Year:Cast:Description            1.069e-01  6.135e-02   1.743   0.0813
## Time:Cast:Description            2.144e+00  1.319e+00   1.625   0.1042
## BritishOther:Year:Time:Cast     -2.188e-01  1.749e+03   0.000   0.9999
```

```
## BritishOther:Year:Time:Description -2.098e-01  3.683e+03   0.000   1.0000
## BritishOther:Year:Cast:Description         NA         NA      NA       NA
## BritishOther:Time:Cast:Description         NA         NA      NA       NA
## Year:Time:Cast:Description         -1.090e-03  6.695e-04  -1.629   0.1034
##
## (Intercept)
## BritishOther
## Year
## Time
## Cast
## Description
## BritishOther:Year
## BritishOther:Time
## BritishOther:Cast
## BritishOther:Description
## Year:Time
## Year:Cast
## Year:Description
## Time:Cast
## Time:Description
## Cast:Description
## BritishOther:Year:Time
## BritishOther:Year:Cast
## BritishOther:Year:Description
## BritishOther:Time:Cast
## BritishOther:Time:Description
## BritishOther:Cast:Description
## Year:Time:Cast
## Year:Time:Description
## Year:Cast:Description              .
## Time:Cast:Description
## BritishOther:Year:Time:Cast
## BritishOther:Year:Time:Description
## BritishOther:Year:Cast:Description
## BritishOther:Time:Cast:Description
## Year:Time:Cast:Description
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.820  on 99  degrees of freedom
## Residual deviance:  64.432  on 71  degrees of freedom
## AIC: 122.43
##
## Number of Fisher Scoring iterations: 20

model3w3 = glm(Good~(British+Year+Time+Cast+Description)^3, data = New.Film,
family = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model3w3)
```

```
##
## Call:
## glm(formula = Good ~ (British + Year + Time + Cast + Description)^3,
##     family = binomial(link = "logit"), data = New.Film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8339  -0.5931  -0.2125   0.1168   2.5435
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -9.995e+04  1.054e+07  -0.009    0.992
## BritishOther                   1.006e+05  1.054e+07   0.010    0.992
## Year                           5.181e+01  5.448e+03   0.010    0.992
## Time                           8.037e+02  9.494e+04   0.008    0.993
## Cast                           4.204e+03  1.209e+06   0.003    0.997
## Description                    3.376e+02  4.033e+05   0.001    0.999
## BritishOther:Year             -5.214e+01  5.448e+03  -0.010    0.992
## BritishOther:Time             -8.247e+02  9.494e+04  -0.009    0.993
## BritishOther:Cast             -4.331e+03  1.209e+06  -0.004    0.997
## BritishOther:Description      -1.984e+02  4.033e+05   0.000    1.000
## Year:Time                     -4.299e-01  5.015e+01  -0.009    0.993
## Year:Cast                     -2.069e+00  6.273e+02  -0.003    0.997
## Year:Description              -2.146e-01  2.073e+02  -0.001    0.999
## Time:Cast                      4.844e+00  2.073e+02   0.023    0.981
## Time:Description               3.212e+00  3.797e+02   0.008    0.993
## Cast:Description              -4.995e+01  3.843e+03  -0.013    0.990
## BritishOther:Year:Time         4.403e-01  5.015e+01   0.009    0.993
## BritishOther:Year:Cast         2.131e+00  6.273e+02   0.003    0.997
## BritishOther:Year:Description  1.430e-01  2.073e+02   0.001    0.999
## BritishOther:Time:Cast        -1.410e+00  2.073e+02  -0.007    0.995
## BritishOther:Time:Description -3.424e+00  3.797e+02  -0.009    0.993
## BritishOther:Cast:Description  3.397e+01  3.843e+03   0.009    0.993
## Year:Time:Cast                -1.698e-03  1.191e-03  -1.426    0.154
## Year:Time:Description          1.286e-04  3.877e-04   0.332    0.740
## Year:Cast:Description          8.245e-03  9.158e-03   0.900    0.368
## Time:Cast:Description         -5.986e-03  1.152e-02  -0.520    0.603
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.82  on 99  degrees of freedom
## Residual deviance:  67.99  on 74  degrees of freedom
## AIC: 119.99
##
## Number of Fisher Scoring iterations: 18
```

```r
model3 = glm(Good~(British+Year+Time+Cast+Description)^2, data = New.Film, fa
mily = binomial(link = "logit"))
summary(model3)
```

```
##
## Call:
## glm(formula = Good ~ (British + Year + Time + Cast + Description)^2,
##     family = binomial(link = "logit"), data = New.Film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0007  -0.6749  -0.3465   0.4743   2.2061
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                1.484e+02  2.735e+02   0.543   0.5874
## BritishOther              -8.934e+01  1.561e+02  -0.572   0.5670
## Year                      -7.541e-02  1.434e-01  -0.526   0.5991
## Time                      -2.220e+00  1.935e+00  -1.148   0.2511
## Cast                       2.990e+00  2.572e+01   0.116   0.9074
## Description                1.840e+01  1.533e+01   1.200   0.2302
## BritishOther:Year          3.906e-02  8.165e-02   0.478   0.6324
## BritishOther:Time          7.702e-02  6.218e-02   1.239   0.2155
## BritishOther:Cast          8.003e-01  7.891e-01   1.014   0.3105
## BritishOther:Description   9.283e-02  4.991e-01   0.186   0.8524
## Year:Time                  1.024e-03  9.769e-04   1.048   0.2945
## Year:Cast                 -1.879e-03  1.357e-02  -0.138   0.8899
## Year:Description          -8.332e-03  8.097e-03  -1.029   0.3035
## Time:Cast                  3.586e-02  1.697e-02   2.113   0.0346 *
## Time:Description          -1.486e-04  8.711e-03  -0.017   0.9864
## Cast:Description          -3.006e-01  1.251e-01  -2.403   0.0163 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.820  on 99  degrees of freedom
## Residual deviance:  83.842  on 84  degrees of freedom
## AIC: 115.84
##
## Number of Fisher Scoring iterations: 6
```

```r
anova(model2,model3)
```

```
## Analysis of Deviance Table
##
## Model 1: Good ~ British
## Model 2: Good ~ (British + Year + Time + Cast + Description)^2
##   Resid. Df Resid. Dev Df Deviance
## 1        98    123.820
## 2        84     83.842 14   39.978
```

```r
pchisq(39.978, 14, lower.tail=F)
```

```
## [1] 0.0002571456
```

From the test above, we can see there complex model doesn't have 4, 3, 2 way interaction so the model with only 2 way interaction will be my complex model for the next model selection.

```r
both = step(model3, scope = list(lower=model2, upper=model3), direction = "bo
th")
```

```
## Start:  AIC=115.84
## Good ~ (British + Year + Time + Cast + Description)^2
##
##                       Df Deviance    AIC
## - Time:Description     1   83.842 113.84
## - Year:Cast            1   83.861 113.86
## - British:Description  1   83.877 113.88
## - British:Year         1   84.085 114.08
## - British:Cast         1   84.840 114.84
## - Year:Time            1   84.970 114.97
## - Year:Description     1   85.071 115.07
## - British:Time         1   85.351 115.35
## <none>                     83.842 115.84
## - Time:Cast            1   89.506 119.51
## - Cast:Description     1   91.182 121.18
##
## Step:  AIC=113.84
## Good ~ British + Year + Time + Cast + Description + British:Year +
##     British:Time + British:Cast + British:Description + Year:Time +
##     Year:Cast + Year:Description + Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
## - Year:Cast            1   83.862 111.86
## - British:Description  1   83.877 111.88
## - British:Year         1   84.090 112.09
## - British:Cast         1   84.848 112.85
## - Year:Time            1   84.993 112.99
## - British:Time         1   85.352 113.35
## <none>                     83.842 113.84
## - Year:Description     1   86.139 114.14
## + Time:Description     1   83.842 115.84
## - Time:Cast            1   89.844 117.84
## - Cast:Description     1   91.377 119.38
##
## Step:  AIC=111.86
## Good ~ British + Year + Time + Cast + Description + British:Year +
##     British:Time + British:Cast + British:Description + Year:Time +
##     Year:Description + Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
```

```
## - British:Description  1   83.898 109.90
## - British:Year         1   84.091 110.09
## - British:Cast         1   84.860 110.86
## - Year:Time            1   84.993 110.99
## - British:Time         1   85.393 111.39
## <none>                     83.862 111.86
## - Year:Description     1   86.333 112.33
## + Year:Cast            1   83.842 113.84
## + Time:Description     1   83.861 113.86
## - Cast:Description     1   91.383 117.38
## - Time:Cast            1   91.543 117.54
##
## Step:  AIC=109.9
## Good ~ British + Year + Time + Cast + Description + British:Year +
##     British:Time + British:Cast + Year:Time + Year:Description +
##     Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
## - British:Year         1   84.283 108.28
## - British:Cast         1   85.048 109.05
## - Year:Time            1   85.091 109.09
## - British:Time         1   85.428 109.43
## <none>                     83.898 109.90
## - Year:Description     1   86.358 110.36
## + British:Description  1   83.862 111.86
## + Year:Cast            1   83.877 111.88
## + Time:Description     1   83.897 111.90
## - Time:Cast            1   91.543 115.54
## - Cast:Description     1   91.689 115.69
##
## Step:  AIC=108.28
## Good ~ British + Year + Time + Cast + Description + British:Time +
##     British:Cast + Year:Time + Year:Description + Time:Cast +
##     Cast:Description
##
##                       Df Deviance    AIC
## - Year:Time            1   85.615 107.61
## - British:Cast         1   85.797 107.80
## <none>                     84.283 108.28
## - Year:Description     1   86.638 108.64
## - British:Time         1   87.504 109.50
## + British:Year         1   83.898 109.90
## + British:Description  1   84.091 110.09
## + Time:Description     1   84.267 110.27
## + Year:Cast            1   84.283 110.28
## - Cast:Description     1   91.972 113.97
## - Time:Cast            1   92.564 114.56
##
## Step:  AIC=107.61
## Good ~ British + Year + Time + Cast + Description + British:Time +
```

```
##     British:Cast + Year:Description + Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
## - British:Cast         1   87.244 107.24
## <none>                     85.615 107.61
## - Year:Description     1   87.665 107.67
## - British:Time         1   87.913 107.91
## + Year:Time            1   84.283 108.28
## + British:Year         1   85.091 109.09
## + British:Description  1   85.248 109.25
## + Time:Description     1   85.542 109.54
## + Year:Cast            1   85.586 109.59
## - Cast:Description     1   92.578 112.58
## - Time:Cast            1   93.726 113.73
##
## Step:  AIC=107.24
## Good ~ British + Year + Time + Cast + Description + British:Time +
##     Year:Description + Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
## - Year:Description     1   89.222 107.22
## <none>                     87.244 107.24
## + British:Cast         1   85.615 107.61
## + Year:Time            1   85.797 107.80
## + British:Year         1   86.275 108.28
## + British:Description  1   86.305 108.31
## + Year:Cast            1   86.992 108.99
## + Time:Description     1   87.226 109.23
## - British:Time         1   91.989 109.99
## - Cast:Description     1   93.428 111.43
## - Time:Cast            1   93.742 111.74
##
## Step:  AIC=107.22
## Good ~ British + Year + Time + Cast + Description + British:Time +
##     Time:Cast + Cast:Description
##
##                       Df Deviance    AIC
## <none>                     89.222 107.22
## + Year:Description     1   87.244 107.24
## + British:Cast         1   87.665 107.67
## + Year:Time            1   88.074 108.07
## + British:Year         1   88.472 108.47
## + British:Description  1   88.516 108.52
## + Time:Description     1   88.640 108.64
## - British:Time         1   93.013 109.01
## + Year:Cast            1   89.193 109.19
## - Time:Cast            1   95.303 111.30
## - Year                 1   95.603 111.60
## - Cast:Description     1   96.878 112.88
```

```r
formula(both)
```

```
## Good ~ British + Year + Time + Cast + Description + British:Time +
##     Time:Cast + Cast:Description
```

From the model selection above, we can see the model R slect the model with least AIC, so the model Good ~ British + Year + Time + Cast + Description + British:Time + Time:Cast + Cast:Description will be my best model.

```r
#use the selected model for prediction
model4 = glm(Good ~ British + Year + Time + Cast + Description + British:Time
 + Time:Cast + Cast:Description, data=New.Film, family = binomial(link = "log
it"))
summary(model4)
```

```
##
## Call:
## glm(formula = Good ~ British + Year + Time + Cast + Description +
##     British:Time + Time:Cast + Cast:Description, family = binomial(link =
"logit"),
##     data = New.Film)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7730  -0.7440  -0.3728   0.5621   2.2292
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       84.40417   38.41277   2.197  0.02800 *
## BritishOther      -8.57315    4.87575  -1.758  0.07869 .
## Year              -0.04543    0.01910  -2.378  0.01740 *
## Time              -0.18813    0.10469  -1.797  0.07235 .
## Cast               0.51575    1.16679   0.442  0.65847
## Description        1.91584    0.73283   2.614  0.00894 **
## BritishOther:Time  0.09487    0.05168   1.836  0.06639 .
## Time:Cast          0.02678    0.01268   2.112  0.03471 *
## Cast:Description  -0.26650    0.10831  -2.460  0.01388 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 123.820  on 99  degrees of freedom
## Residual deviance:  89.222  on 91  degrees of freedom
## AIC: 107.22
##
## Number of Fisher Scoring iterations: 5
```

From the test above, we can see Predicted Good = 84.40417 - 8.57315*BritishOther* - *0.04543*Year - 0.18813*Time* + *0.51575*Cast + 1.91584*Description* + *0.09487*BritishOther:Time + 0.02678*Time:Cast* - *0.26650*Cast:Description

In the last step, we will plug in all values we know to predict if the movie is good:

```
newx=data.frame(British=as.factor('British'), Year=1982, Time=75, Cast=13, De
scription=5)
#logodds = predict(model4, newdata = newx) log(p/(1-p))=beata0+beta1*x1+beata
2*x2+++
#exp(logodds)/(1+exp(logodds)) # probability
predict(model4, newdata = newx, type = 'response')
```

```
##        1
## 0.995193
```

From the result above, we can see the movie is predicted to be a good movie since it's very close to 1