HW05

caojilin

10/24/2018

Problem 1

```
test = read.csv("test.csv")
test$Pclass = as.factor(test$Pclass)
train = read.csv("train.csv")
train$Pclass = as.factor(train$Pclass)
summary(train$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.42   20.12   28.00   29.70   38.00   80.00     177
```

We notice that Age has 177 missing values. So we have to fill in these missing values. We'd like to replace them with average Age. But we divided data into several groups. People with the "Miss." title are ususally young. So we replace their missing Age by the average Age of the people with "Miss." title. Etc.

```
#average age for "Miss."
age1 = mean(train[grep("Miss",train$Name),]$Age,na.rm = TRUE)
#average age for "Mrs."
age2 = mean(train[grep("Mrs",train$Name),]$Age,na.rm=TRUE)
#average age for "Master."
age3 = mean(train[grep("Master",train$Name),]$Age,na.rm = TRUE)
#average age for "Mr."
age4 = mean(train[grep("Mr",train$Name),]$Age,na.rm = TRUE)

train$Age[grep("Miss",train$Name)]= age1
train$Age[grep("Mrs",train$Name)]= age2
train$Age[grep("Master",train$Name)]= age3
train$Age[grep("Mr",train$Name)]= age4
train$Age[grep("Dr",train$Name)] = age4
summary(train$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   4.574  28.000  33.118  29.744  33.118  70.000
```

```
train$Title = "Other"
train$Title[grep("Miss",train$Name)]="Miss"
train$Title[grep("Mrs",train$Name)]="Mrs"
train$Title[grep("Master",train$Name)]="Master"
train$Title[grep("Mr",train$Name)]= "Mr"
```

```
#average age for "Miss."
age1 = mean(test[grep("Miss",test$Name),]$Age,na.rm = TRUE)
#average age for "Mrs."
age2 = mean(test[grep("Mrs",test$Name),]$Age,na.rm=TRUE)
#average age for "Master."
age3 = mean(test[grep("Master",test$Name),]$Age,na.rm = TRUE)
#average age for "Mr."
age4 = mean(test[grep("Mr",test$Name),]$Age,na.rm = TRUE)

test$Age[grep("Miss",test$Name)]= age1
test$Age[grep("Mrs",test$Name)]= age2
test$Age[grep("Master",test$Name)]= age3
test$Age[grep("Mr",test$Name)]= age4
test$Age[grep("Ms",test$Name)]= age1
summary(test$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.406  33.747  33.747  30.306  33.747  53.000
```

```
test$Title = "Other"
test$Title[grep("Miss",test$Name)]="Miss"
test$Title[grep("Mrs",test$Name)]="Mrs"
test$Title[grep("Master",test$Name)]="Master"
test$Title[grep("Mr",test$Name)]= "Mr"

test$Fare[which(is.na(test$Fare))] = mean(test$Fare, na.rm = TRUE)
```

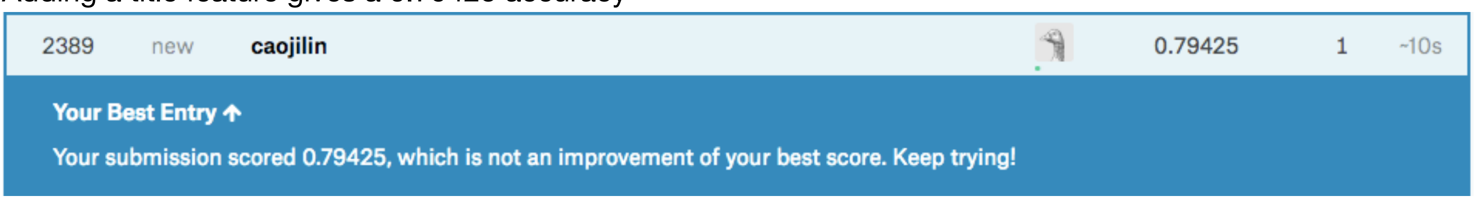
```
formula = "Survived ~ Title + Pclass + Sex + Age + Fare + SibSp + Parch"
model = glm(formula, family = "binomial", data=train)

rs =summary(regsubsets(Survived ~ Title + Pclass + Sex + Age + Fare + SibSp + Parch+E
mbarked,data=train))

fit = predict(model,test,type = 'response')

sur = rep(2, 418)
for (i in 1:418) {
  if (fit[i] > 0.6){
    sur[i] = 1
  }else{
    sur[i] = 0
  }
}
test$Survived = sur
write.csv(test[c("PassengerId","Survived")],file = "submmision.csv",row.names = FALSE
)
```

Adding a title feature gives a 0.79425 accuracy



Problem 2

For logistic regression, we estimate β_i by maximize the log likelihood $\ell(\beta)$
and its gradient is given by $\nabla\ell(\beta) = X^T(Y - \hat{p})$
in order to maximize the log likelihood, we let $\nabla\ell(\beta) = 0$
$X^T(Y - \hat{p}) = 0$, this means $Y - \hat{p}$ is orthogonal to the row of X^T,
which is also the column of X

Problem 3

se = 0.6864/0.313 =2.193

```
Coefficients:
                Estimate  Std.Error z value Pr(>|z|)
(Intercept)      0.6864    XXXXX     0.313   0.754146
log(distance)   -0.9050    XXXXX    -4.349   1.37e-05 ***
log(NoOfPools)   0.5027    0.2004    2.509   0.012102 *
meanmin          1.1153    0.3131    3.562   0.000369 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ

(Dispersion parameter for binomial family taken to be 1)
```

se=-0.905/-4.349=0.208

ybar=79/212 mll.null =212*ybar*log(ybar)+ 212*(1-ybar)*log(1-ybar)= -139.99 null deviance = -2*mll.null=279.98

```
Null deviance: XXXXX  on XXX  degrees of freedom
Residual deviance: XXXXX  on XXX  degrees of freedom
AIC: 222.18

Number of Fisher Scoring iterations: 5
```

df=212-1=211

214.18 df=n-p-1=208

AIC = residual deviance plus 2*(p+1)
p=3
222.18-8=214.18

Also consider the following R code:

```
X = model.matrix(frogs.glm)
W = diag(frogs.glm$fitted.values*(1 - frogs.glm$fitted.values))
solve(t(X) %*% W %*% X)
```

which gave me the output

```
               (Intercept) log(distance) log(NoOfPools)     meanmin
(Intercept)      4.8038479  -0.363947754  -0.255928180 -0.49698440
log(distance)   -0.3639478   0.043313307   0.008053415  0.01562971
log(NoOfPools)  XXXXXXXXXX   0.008053415   0.040141698  0.02678507
meanmin         -0.4969844   0.015629708   0.026785069  XXXXXXXXX
```

-0.255928180, same as (1,3) entry 0.3131^2 = 0.09803161

b. plug in data we get

$$p_i = \frac{e^{\beta_0 + X\beta}}{1 + e^{\beta_0 + X\beta}} = 0.7646$$

c. the null deviance won't get affected, because it has not involved any explanatory variables
the residual deviance could increase or decrease, because it is related to AIC, which could increase or decrease when we add more parameters.

Problem 4

a.

$$\nabla \ell(\beta) = X^T(Y - \hat{p}) = 0$$

b)

$$\hat{p} = \frac{e^{\hat{\beta}_0 + X\hat{\beta}}}{1 + e^{\hat{\beta}_0 + X\hat{\beta}}}$$

c)

$$Y - \hat{p} \text{ is orthogonal to column of } X$$
$$X \text{ has 1 as its first column, so}$$
$$\sum_{i=1}^{n} y_i - \hat{p_i} = 0$$
$$\text{the number of } y_i \text{ that are equal to 1}$$
$$\sum_{i=1}^{n} y_i = \sum_{i=i}^{n} \hat{p_i}$$

d)

$$\text{residual deviance } = -2 * \sum_{i=1}^{n} [y_i \cdot log(\hat{p_i}) + (1 - y_i) \cdot log(1 - \hat{p_i})]$$

Problem 5

```
Coefficients:
                 Estimate  Std. Error z value
(Intercept)       4.11947   0.36342   XXXXXX   z=4.11947/0.36342=11.335
log(crl.tot)      0.30228   0.03693    8.185
log(dollar + s)   0.32586   0.02365   13.777
log(bang + s)     0.40984   0.01597   25.661
log(money + s)    XXXXXXX   0.02800   12.345   estimate=0.028*12.345=0.34566
log(n000 + s)     0.18947   0.02931    6.463
log(make + s)    -0.11418   0.02206   -5.177
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ

(Dispersion parameter for binomial family taken to be 1)
mll.null =4601*0.394*log(0.394)+ 4601*(1-0.394)*log(1-0.394)= −3084.99
Null deviance: XXXXXX  on XXXX  degrees of freedom   null deviance=-2*mll.null=6169.97
                                                      df = n-1=4600
Residual deviance: 3245.1  on XXXX  degrees of freedom
AIC: XXXXXX   AIC = residual deviance + 2*(p+1)=3245.1+2*7= 3259.1
              df = n-p-1=4601-7=4594

Number of Fisher Scoring iterations: 6
```

n=4601
ybar=1813/4601
=0.394
p=6

b. plug in data we get

$$\beta_0 + X\beta = 4.11947 + 0.30228 * log(157) + 0.32586 * log(0.868 + 0.001) + 0.40984 * log(2.894 + 0.001) = 6.037771$$

$$p_i = \frac{e^{(\beta_0 + X\beta)}}{1 + e^{(\beta_0 + X\beta)}} = \frac{e^{(6.037771)}}{1 + e^{(6.037771)}} = 0.9976188$$

c. I would use M1, since AIC is smaller for M1 than M2. Perhaps the original data are skewed, so taking logarithms transformation is better.