**[STA6714-19Spring 0001](https://webcourses.ucf.edu/courses/1320950)**:**Homework for Chapter 10**

**Mithun Mohanraj**

**Data preprocessing :**

For creating dummies , the following r code is used:

ebay <- dummy.data.frame(dat, sep = ".")

**a. Create pivot tables for the mean of the binary outcome (Competitive?) as a function**

**of the various categorical variables (use the original variables, not the dummies).**

**Use the information in the tables to reduce the number of dummies that will be**

**used in the model. For example, categories that appear most similar with respect**

**to the distribution of competitive auctions could be combined.**

R code for pivot table :

data.m <- melt(dat, id=c(1:7), measure=c(8))

pivot<- cast(data.m,Category~variable , mean,margins = TRUE)

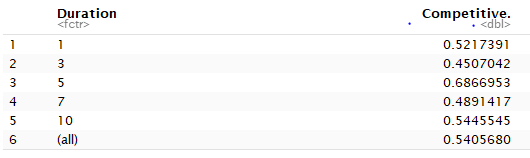
pivot

For endday , the pivot table follows:



Based on pivot table , we can combine Friday , Sunday and Wednesday together as dummies .

For Duration , the pivot table follows:



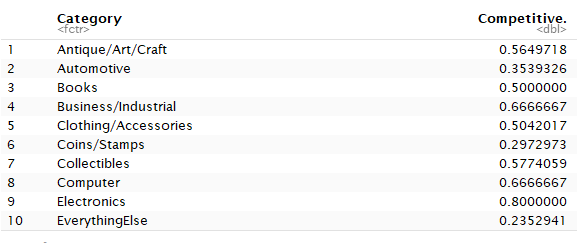
Since 5-day duration is giving high competitiveness , we can combine all the other dummies together as 5-day duratiom

For currency ,



Since all the mean are different , we cannot combine any dummies in currency.

For category,





Since we have all range of values between 0 and 1 , ideally we can divide the category variables into three ranges 0 to 0.3 , 0.3 to 0.6 , 0.6 to 1

**B,Split the data into training (60%) and validation (40%) datasets. Run a logistic model with all predictors with a cutoff of 0.5.**

**R code :**

set.seed(2)

train.index <- sample(c(1:dim(ebay)[1]), dim(ebay)[1]\*0.6)

train.df <- ebay[train.index,]

valid.df <- ebay[-train.index, ]

logit.reg <- glm(train.df$Competitive ~ ., data = train.df[,-33], family = "binomial")

options(scipen=999)

summary(logit.reg)

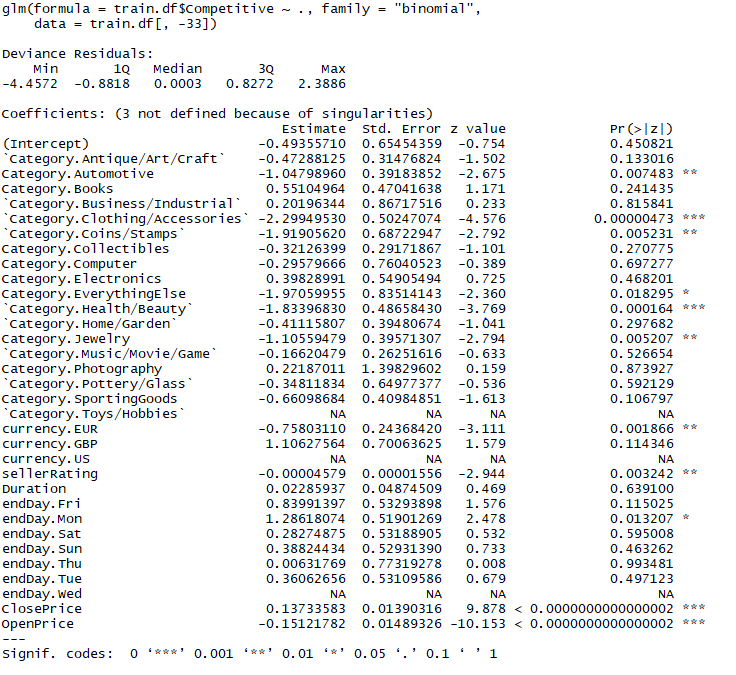
logit.reg.pred <- predict(logit.reg, valid.df[,-33], type = "response")

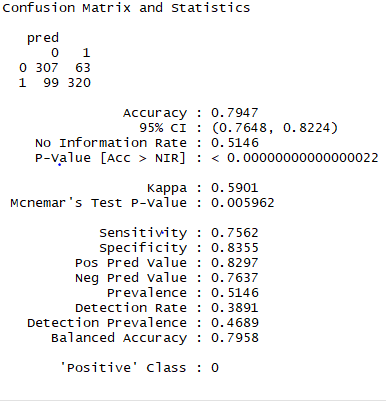
pred=ifelse(logit.reg.pred>0.54,"1","0")

conf=table(valid.df$Competitive.,pred)

confusionMatrix(conf)

**the results are :**

****

****

**c,** **If we want to predict at the start of an auction whether it will be competitive, we**

**cannot use the information on the closing price. Run a logistic model with all**

**predictors as above, excluding price. How does this model compare to the full**

**model with respect to predictive accuracy?**

**R code:**

train.df <- ebay[train.index,-31]

valid.df <- ebay[-train.index, -31]

logit.reg <- glm(train.df$Competitive ~ ., data = train.df[,-32], family = "binomial")

options(scipen=999)

summary(logit.reg)

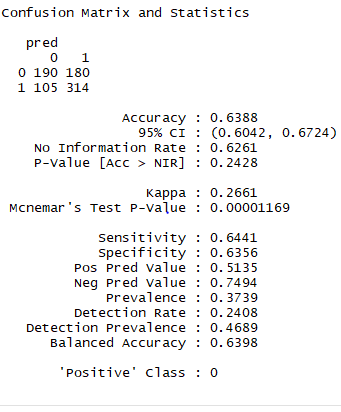
logit.reg.pred <- predict(logit.reg, valid.df[,-32], type = "response")

pred=ifelse(logit.reg.pred>0.54,"1","0")

conf=table(valid.df$Competitive.,pred)

confusionMatrix(conf)

results :



As we see the predictive accuracy for full model was 79% whereas after closing price reduces to 63%.

**d. Interpret the meaning of the coefficient for closing price. Does closing price have**

**a practical significance? Is it statistically significant for predicting competitiveness**

**of auctions? (Use a 10% significance level.**

The coefficient for closing price is 1.37 which means that the odds of higher closing price will have high competitiveness than lower closing price

Closing price is not practically significant because in application the closing price is always unknown before the auction.

Closing price is statistically significant because we can see the importance of variable in the summary with three\*\*\* with value of prob = 0.00002 which is well less than significance level 0.1

**e. Use stepwise selection (use function step() in the stats package or function**

**stepAIC() in the MASS package) and an exhaustive search (use function glmulti() in**

**package glmulti) to find the model with the best fit to the training data. Which**

**predictors are used?**

R code for stepwise:

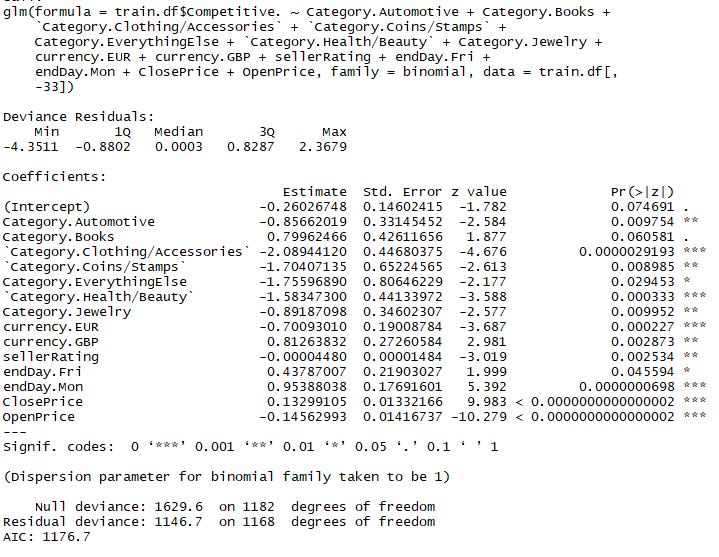
mod1 = glm(train.df$Competitive.~ .,data=train.df[,-33],family=binomial)

#stepwise = step(mod1, direction = "both")

stepw1=step(mod1,trace = 0)

summary(stepw1)

**result:**



**R code for exhaustive :**

prednames <- c("a","b","c","d", "e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z","aa","bb","cc","dd","ee","ff")

names(train.df)[1:32] <- prednames

res <- glmulti(y="Competitive.",xr=prednames, data=train.df, crit="aicc")

summary(res)

**result :** exhaustive search takes all the time to run.

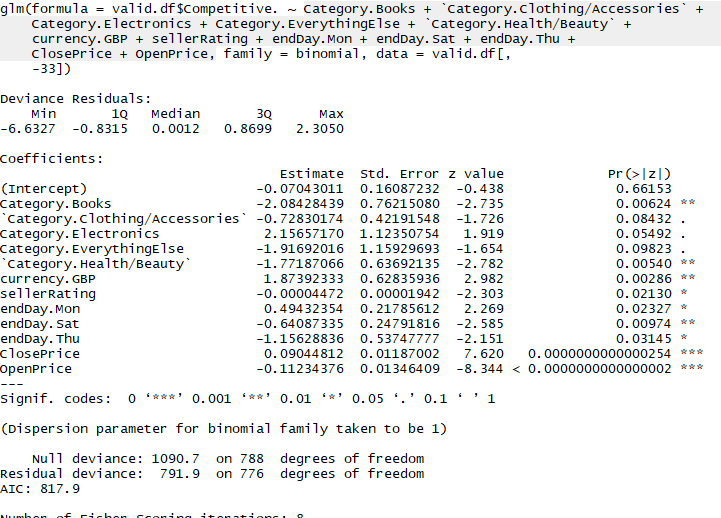
Predictors used :

Category.Automotive + Category.Books + `Category.Clothing/Accessories` + `Category.Coins/Stamps` + Category.EverythingElse +`Category.Health/Beauty` + Category.Jewelry + currency.EUR + currency.GBP + sellerRating + endDay.Fri + endDay.Mon +closeprice+ OpenPrice

**f. Use stepwise selection and an exhaustive search to find the model with the lowest**

**predictive error rate (use the validation data). Which predictors are used?**

**Result for stepwise:**

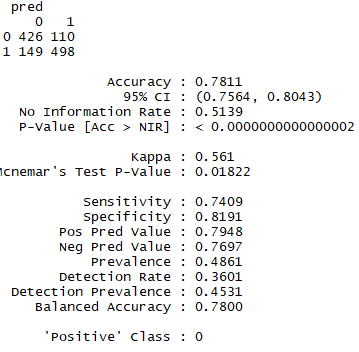


**Predictors used:**

Category.Books + `Category.Clothing/Accessories` +

Category.Electronics + Category.EverythingElse + `Category.Health/Beauty` + currency.GBP + sellerRating + endDay.Mon + endDay.Sat + endDay.Thu + ClosePrice+ OpenPrice,

**Results are :**



The highest predictive accuracy is 78.11%

**Exhaustive search :**

It takes forever to run.

**g. What is the danger of using the best predictive model that you found?**

As we see in the above results, the resulting accuracy is very low for validation set compared to the training set, which means the model fits closely to the data meaning overfitting.

**H,Explain why the best-fitting model and the best predictive models are the same or different.**

They both are different

Best -fitting model : tries to fit the model closely to data using predictor significance .

Best predictor model : tries to achieve high predictive accuracy by evaluating the model on new data unlike just considering predictor significance.

**i. If the major objective is accurate classification, what cutoff value should be used?**

For accurate classification , the optimal cutoff value is 0.4679 which increases the accuracy to 79%

Since the task is finding all the competitive auctions , cut-off should be low because we cant afford to lose even a single competitive auction.

**j. Based on these data, what auction settings set by the seller (duration, opening price,**

**ending day, currency) would you recommend as being most likely to lead to a**

**competitive auction?**

Duration : 5 days

Endday : Monday , Tuesday, Thursday

Currency : GBP