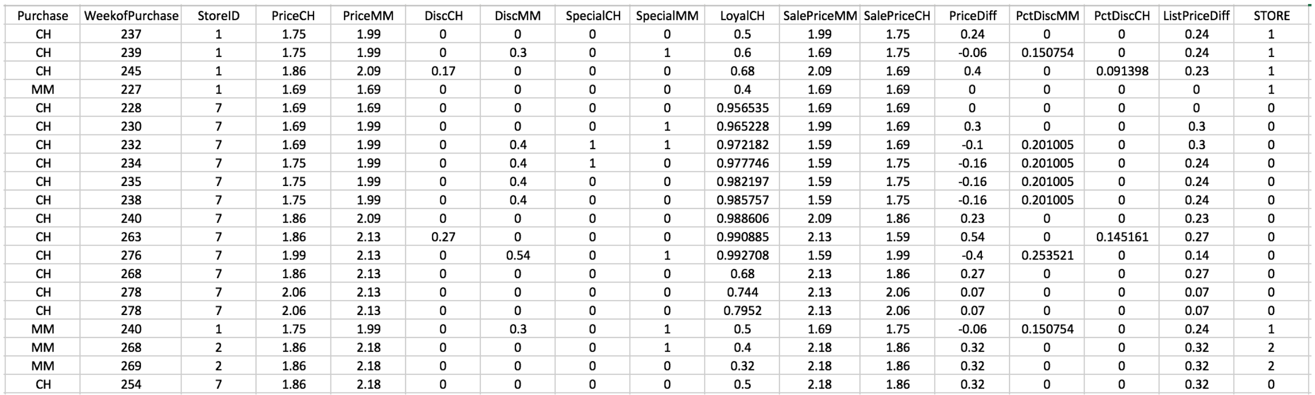
Classification Methods
Logistic Regression & Decision Tree

Introduction

In this project, we will analyze the consumer data to target a promotion for the MM brand. The screenshot of the dataset is as followed. The first variable Purchase is the brand of orange juice the consumer previously purchased, which is either the brand MM or CH. The other variables are as following:

- WeekofPurchase - Week of purchase
- StoreID - Store ID
- PriceCH/PriceMM - Price charged for CH/MM
- DiscCH/DiscMM - Discount offered for CH/MM
- SpecialCH/ SpecialMM - Indicator of special on CH/MM
- LoyalCH - A proxy for customer brand loyalty for CH
- SalePriceCH/SalePriceMM - Sale price for CH/MM
- PriceDiff - Sale price of MM less sale price of CH



Purchase	WeekofPurchase	StoreID	PriceCH	PriceMM	DiscCH	DiscMM	SpecialCH	SpecialMM	LoyalCH	SalePriceMM	SalePriceCH	PriceDiff	PctDiscMM	PctDiscCH	ListPriceDiff	STORE
CH	237	1	1.75	1.99	0	0	0	0	0.5	1.99	1.75	0.24	0	0	0.24	1
CH	239	1	1.75	1.99	0	0.3	0	1	0.6	1.69	1.75	-0.06	0.150754	0	0.24	1
CH	245	1	1.86	2.09	0.17	0	0	0	0.68	2.09	1.69	0.4	0	0.091398	0.23	1
MM	227	1	1.69	1.69	0	0	0	0	0.4	1.69	1.69	0	0	0	0	1
CH	228	7	1.69	1.69	0	0	0	0	0.956535	1.69	1.69	0	0	0	0	0
CH	230	7	1.69	1.99	0	0	0	1	0.965228	1.99	1.69	0.3	0	0	0.3	0
CH	232	7	1.69	1.99	0	0.4	1	1	0.972182	1.59	1.69	-0.1	0.201005	0	0.3	0
CH	234	7	1.75	1.99	0	0.4	1	0	0.977746	1.59	1.75	-0.16	0.201005	0	0.24	0
CH	235	7	1.75	1.99	0	0.4	0	0	0.982197	1.59	1.75	-0.16	0.201005	0	0.24	0
CH	238	7	1.75	1.99	0	0.4	0	0	0.985757	1.59	1.75	-0.16	0.201005	0	0.24	0
CH	240	7	1.86	2.09	0	0	0	0	0.988606	2.09	1.86	0.23	0	0	0.23	0
CH	263	7	1.86	2.13	0.27	0	0	0	0.990885	2.13	1.59	0.54	0	0.145161	0.27	0
CH	276	7	1.99	2.13	0	0.54	0	1	0.992708	1.59	1.99	-0.4	0.253521	0	0.14	0
CH	268	7	1.86	2.13	0	0	0	0	0.68	2.13	1.86	0.27	0	0	0.27	0
CH	278	7	2.06	2.13	0	0	0	0	0.744	2.13	2.06	0.07	0	0	0.07	0
CH	278	7	2.06	2.13	0	0	0	0	0.7952	2.13	2.06	0.07	0	0	0.07	0
MM	240	1	1.75	1.99	0	0.3	0	1	0.5	1.69	1.75	-0.06	0.150754	0	0.24	1
MM	268	2	1.86	2.18	0	0	0	1	0.4	2.18	1.86	0.32	0	0	0.32	2
MM	269	2	1.86	2.18	0	0	0	0	0.32	2.18	1.86	0.32	0	0	0.32	2
CH	254	7	1.86	2.18	0	0	0	0	0.5	2.18	1.86	0.32	0	0	0.32	0

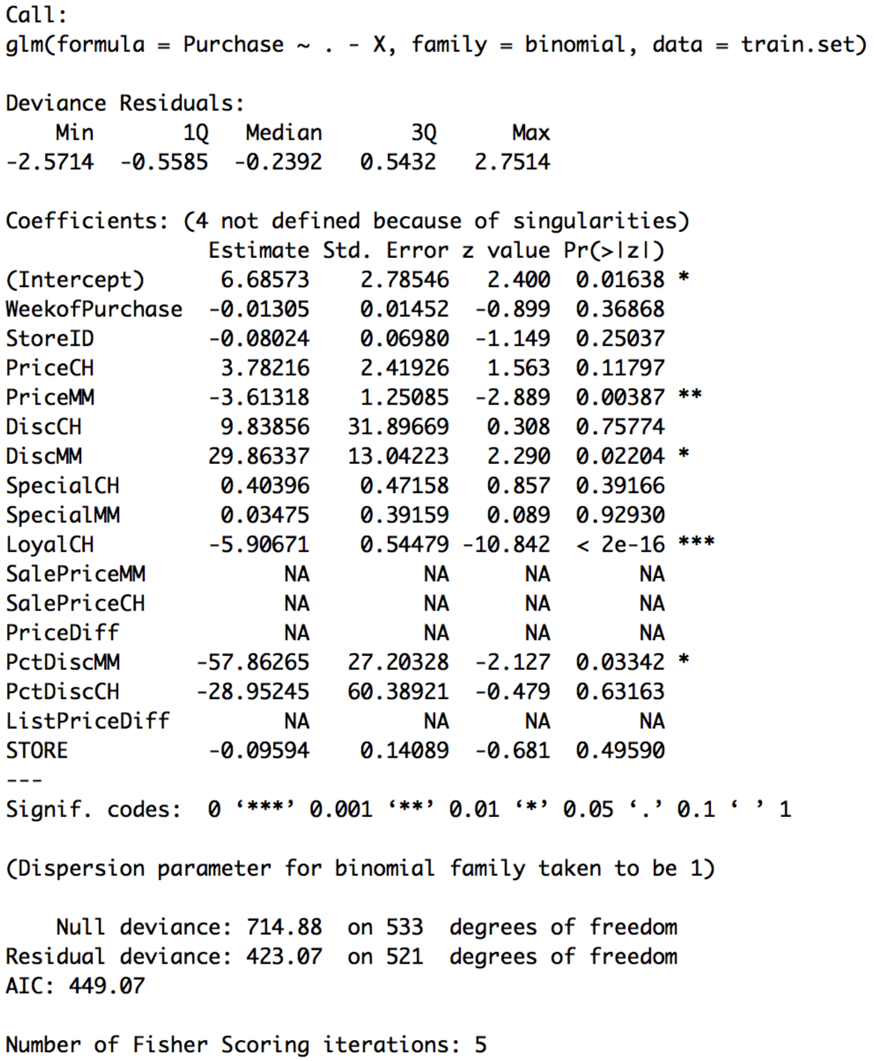
Objective

We will use logistic regression and decision tree to fit three models and select the best model. Based on the best model selected, we will discuss a case for an optimization problem regarding the decision threshold.

Solution

Step 1
Split the data into training data (50%), validation data (25%) and test data (25%).

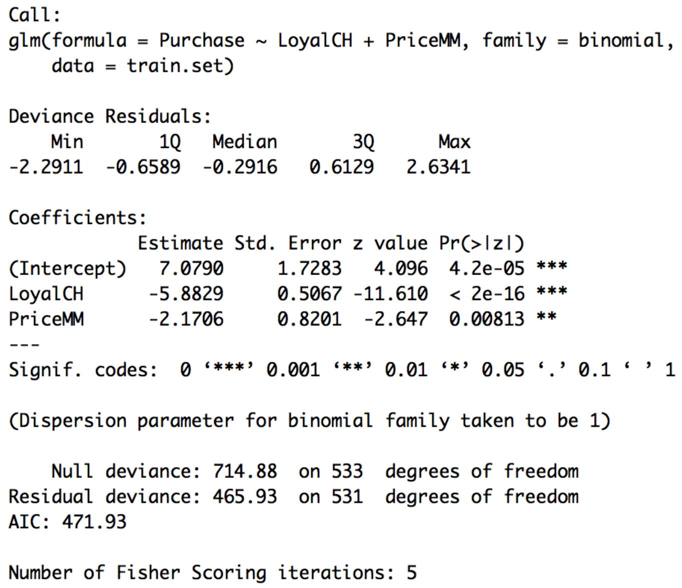
Step 2
Use all the variables to fit a logistic regression mode. The summary is as followed:
(Model A)



```
Call:
glm(formula = Purchase ~ . - X, family = binomial, data = train.set)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.5714  -0.5585  -0.2392   0.5432   2.7514  

Coefficients: (4 not defined because of singularities)
                Estimate Std. Error z value Pr(>|z|)    
(Intercept)      6.68573    2.78546   2.400  0.01638 *  
WeekofPurchase  -0.01305    0.01452  -0.899  0.36868    
StoreID         -0.08024    0.06980  -1.149  0.25037    
PriceCH          3.78216    2.41926   1.563  0.11797    
PriceMM         -3.61318    1.25085  -2.889  0.00387 ** 
DiscCH           9.83856   31.89669   0.308  0.75774    
DiscMM          29.86337   13.04223   2.290  0.02204 *  
SpecialCH        0.40396    0.47158   0.857  0.39166    
SpecialMM        0.03475    0.39159   0.089  0.92930    
LoyalCH         -5.90671    0.54479 -10.842  < 2e-16 ***
SalePriceMM           NA         NA      NA       NA    
SalePriceCH           NA         NA      NA       NA    
PriceDiff             NA         NA      NA       NA    
PctDiscMM      -57.86265   27.20328  -2.127  0.03342 *  
PctDiscCH      -28.95245   60.38921  -0.479  0.63163    
ListPriceDiff         NA         NA      NA       NA    
STORE           -0.09594    0.14089  -0.681  0.49590    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 714.88  on 533  degrees of freedom
Residual deviance: 423.07  on 521  degrees of freedom
AIC: 449.07

Number of Fisher Scoring iterations: 5
```

The result shows that only a few of covariates are significant in the model. For some covariates, say SalePriceMM/CH, PriceDiff, ListPriceDiff, all of them can be computed by other covariates. Therefore, their coefficients could not be solved and we need to remove them from the model.

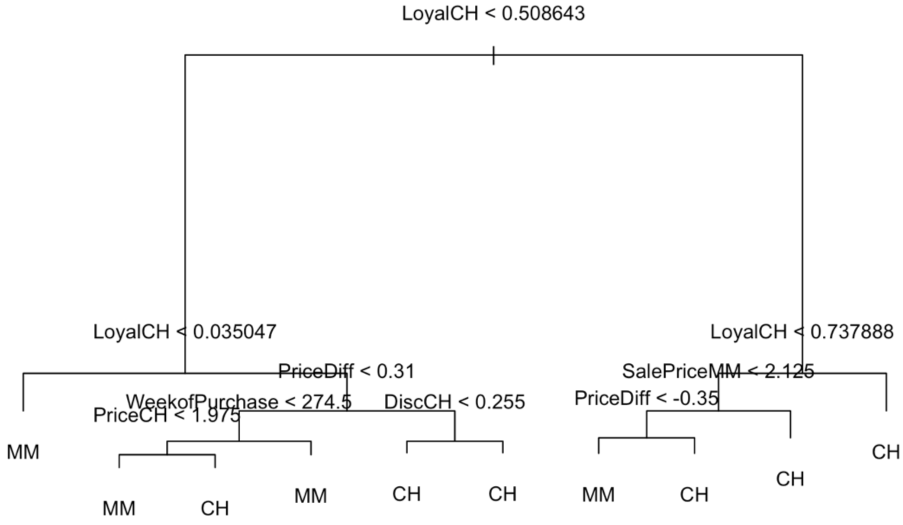
Step 3
Based on the result of the previous model, we will choose the most two significant covariates—LoyalCH and PriceMM to fit a new logistic model. (Model B)



```
Call:
glm(formula = Purchase ~ LoyalCH + PriceMM, family = binomial,
    data = train.set)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2911  -0.6589  -0.2916   0.6129   2.6341

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   7.0790     1.7283   4.096  4.2e-05 ***
LoyalCH      -5.8829     0.5067 -11.610  < 2e-16 ***
PriceMM      -2.1706     0.8201  -2.647  0.00813 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 714.88  on 533  degrees of freedom
Residual deviance: 465.93  on 531  degrees of freedom
AIC: 471.93

Number of Fisher Scoring iterations: 5
```

The result above shows that both the loyalty level of CH and the price of MM have negative relationship with the customers’ possibility to purchase MM.

Step 4
Next, we fit a decision tree to the data. (Model C)



Step 5
Then we compute the error rate for the three models based on the validation data:

Model	A	B	C
Error Rate	0.1716	0.1903	0.2201

In comparison, model A performed the best.

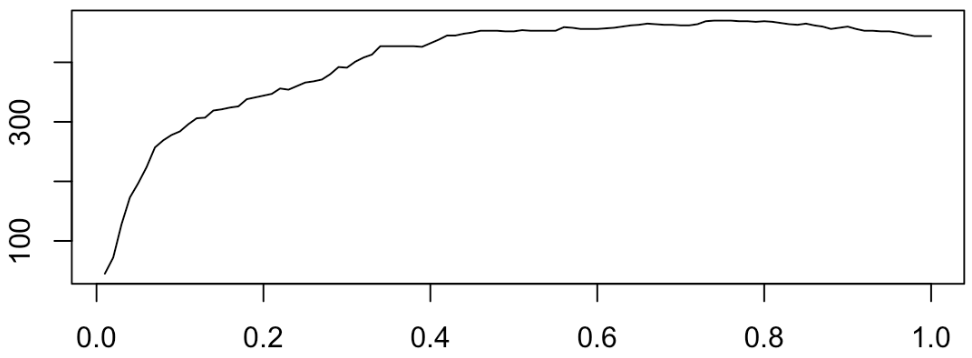
Step 6
Use all the training and validation data, we retrain model A and evaluate it by the test data, the correct prediction rate is: 0.8507.

Step 7
If we are running a promotion aimed to convince customers to sample the MM orange juice. We wish to target customers who bought CH orange juice and give them a coupon for MM. Our marketing team tells you that a coupon handed to a customer who bought CH will help convert the customer and will generate a $3 profit, however a coupon that is handed to a customer who already bought MM will be a waste and will generate a loss of $1.

Using one of the previous models, we need to decide who receives coupons, namely, find the optimal threshold to convert the probability to a decision.

Here we use a for-loop with 100 steps to find the best threshold for the decision (by computing the estimated reward), increasing the threshold from 0.01 to 1.00.

Using the validation data, the best threshold is 0.77 (below this value I will assume that customer has bought a CH orange juice and therefore give him/her a coupon for MM) and the corresponding payoff is 471 dollars (for 268 observations). The result is as followed:



Using the test data to assess the model, the best attainable payoff is 387. (for 268 observations)

Attachment: R code

```
Orange=read.csv('OrangeJuice.csv')

#Step 1
set.seed(4650)
train=sample(nrow(Orange),0.75*nrow(Orange))
train.set=Orange[train[1:floor(2/3*length(train))],]
valid.set=Orange[train[(floor(2/3*length(train))+1):length(train)],]
test.set=Orange[-train,]
summary(train.set)

#Step 2
attach(Orange)
log.fit.b=glm(Purchase~.-X, family=binomial, data=train.set)
summary(log.fit.b)

#Step 3
log.fit.c=glm(Purchase~LoyalCH+PriceMM, family=binomial, data=train.set)
summary(log.fit.c)

#Step 4
library(tree)
OrangeTree=tree(Purchase~.-X, data=train.set)
summary(OrangeTree)
plot(OrangeTree)
text(OrangeTree,pretty=0)
OrangeTree

#Step 5
prob.b=predict(log.fit.b, newdata=valid.set, type="response")
pred.b=ifelse(prob.b>=0.5,'MM','CH')
error.b=mean(pred.b!=valid.set['Purchase'])
prob.c=predict(log.fit.c, newdata=valid.set, type="response")
pred.c=ifelse(prob.c>=0.5,'MM','CH')
error.c=mean(pred.c!=valid.set['Purchase'])
prob.d=predict(OrangeTree, newdata=valid.set)
pred.d=ifelse(prob.d[,'MM']>=0.5,'MM','CH')
error.d=mean(pred.d!=valid.set['Purchase'])
```

```
#Step 6
train_valid=rbind(train.set, valid.set)
log.fit.b=glm(Purchase~.-X, family=binomial, data=train_valid)
prob.test=predict(log.fit.b, newdata=test.set, type="response")
pred.test=ifelse(prob.test>=0.5,'MM','CH')
correct.rate=mean(pred.test==test.set['Purchase'])

#Step 7
prob=predict(log.fit.b, newdata=valid.set, type="response")
fact=valid.set['Purchase']
best.payoff=0

payoff.list=list()
for (i in 1:100)
{
  threshold=0.01*i
  pred=ifelse(prob>=threshold,'MM','CH')
  TruePositive=(pred=='CH'&fact=='CH')
  FalsePositive=(pred=='CH'&fact=='MM')
  payoff=3*sum(TruePositive)-1*sum(FalsePositive)
  payoff.list[i]=payoff
  if (payoff>best.payoff) {
    best.payoff=payoff
    best.threshold=threshold
  }
}
x.list=c(1:100)*0.01
plot(x=x.list,y=payoff.list, type = 'line')

#use the test data to assess the best.threshold:0.77
prob2=predict(log.fit.b, newdata=test.set, type="response")
pred2=ifelse(prob2>=best.threshold,'MM','CH')
fact2=test.set['Purchase']
TruePositive2=(pred2=='CH'&fact2=='CH')
FalsePositive2=(pred2=='CH'&fact2=='MM')
payoff.test=3*sum(TruePositive2)-1*sum(FalsePositive2)
```