Business Analytics

Assignment-3

Name-Subham Kedia
UNI-sk4355

Answer 1

Part a.

```
> summary(data.train)
 Purchase WeekofPurchase     StoreID      PriceCH         PriceMM         DiscCH            DiscMM
 CH:317   Min.   :227.0   1: 74   Min.   :1.690   Min.   :1.690   Min.   :0.00000   Min.   :0.0000
 MM:218   1st Qu.:239.5   2:116   1st Qu.:1.790   1st Qu.:2.090   1st Qu.:0.00000   1st Qu.:0.0000
          Median :256.0   3:108   Median :1.860   Median :2.090   Median :0.00000   Median :0.0000
          Mean   :254.0   4: 68   Mean   :1.866   Mean   :2.084   Mean   :0.04723   Mean   :0.1191
          3rd Qu.:267.0   7:169   3rd Qu.:1.990   3rd Qu.:2.180   3rd Qu.:0.00000   3rd Qu.:0.2000
          Max.   :278.0           Max.   :2.090   Max.   :2.290   Max.   :0.50000   Max.   :0.8000
   SpecialCH        SpecialMM         LoyalCH          SalePriceMM     SalePriceCH      PriceDiff
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000017   Min.   :1.190   Min.   :1.390   Min.   :-0.6700
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.320000   1st Qu.:1.690   1st Qu.:1.750   1st Qu.: 0.0000
 Median :0.0000   Median :0.0000   Median :0.600000   Median :2.090   Median :1.860   Median : 0.2400
 Mean   :0.1402   Mean   :0.1421   Mean   :0.560588   Mean   :1.965   Mean   :1.819   Mean   : 0.1459
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.854584   3rd Qu.:2.180   3rd Qu.:1.890   3rd Qu.: 0.3000
 Max.   :1.0000   Max.   :1.0000   Max.   :0.999947   Max.   :2.290   Max.   :2.090   Max.   : 0.6400
```

The only qualitative variable that we have in our data set is StoreID. It has been set up as an ordered categorical variable. We use as.factor command to i.e. data$StoreID = as.factor(data$StoreID) to inform R that StoreID is a qualitative variable.

Part b.

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5085  -0.5741  -0.2363   0.5150   2.7720

Coefficients: (3 not defined because of singularities)
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.70964    2.69235   0.635  0.52543
WeekofPurchase 0.01310    0.01625   0.806  0.42034
StoreID2      -0.05185    0.39117  -0.133  0.89455
StoreID3       0.34486    0.54369   0.634  0.52589
StoreID4      -0.70485    0.61150  -1.153  0.24906
StoreID7      -0.58866    0.41574  -1.416  0.15680
PriceCH        3.70726    2.66414   1.392  0.16406
PriceMM       -4.31098    1.32790  -3.246  0.00117 **
DiscCH        -1.84703    1.55225  -1.190  0.23408
DiscMM         2.46123    0.75908   3.242  0.00119 **
SpecialCH     -0.75005    0.52243  -1.436  0.15109
SpecialMM      0.06780    0.41082   0.165  0.86892
LoyalCH       -6.26312    0.58487 -10.709  < 2e-16 ***
SalePriceMM         NA         NA      NA       NA
SalePriceCH         NA         NA      NA       NA
PriceDiff           NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 723.24  on 534  degrees of freedom
Residual deviance: 409.75  on 522  degrees of freedom
AIC: 435.75

Number of Fisher Scoring iterations: 5
```

The coefficients of SalePriceMM, SalePriceCH, PriceDiff are zero which indicate that these columns are linearly dependent and will not have any effect on the models to provide us with predictions. The p-value of LoyalCH is very low which indicates that the 'LoyalCH' feature is very important for our models.

Part c.

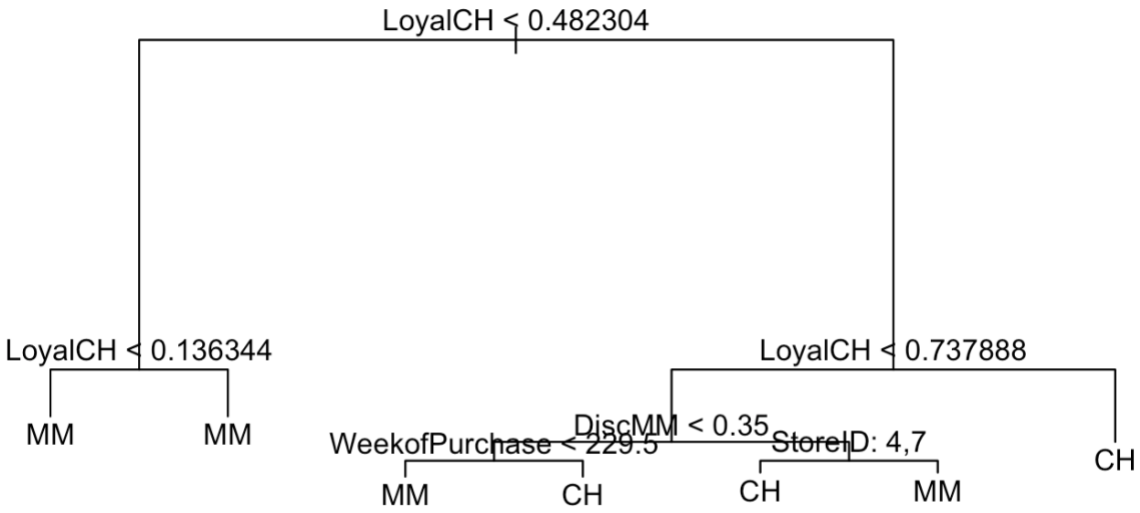
```
> bestlam
[1] 0.01
> lasso.mod = glmnet(x, y, alpha=1, lambda=bestlam, family="binomial")
> lasso.mod

Call:  glmnet(x = x, y = y, family = "binomial", alpha = 1, lambda = bestlam)

     Df  %Dev Lambda
[1,] 10 0.423   0.01
```

The best lambda is 0.01.

Part d.

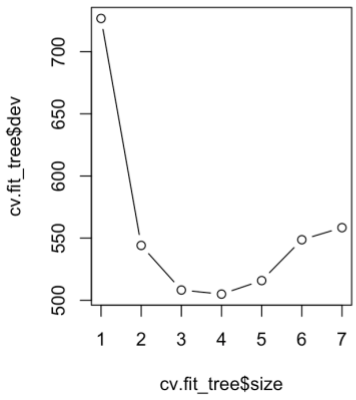
Decision Tree before cross validation and pruning.

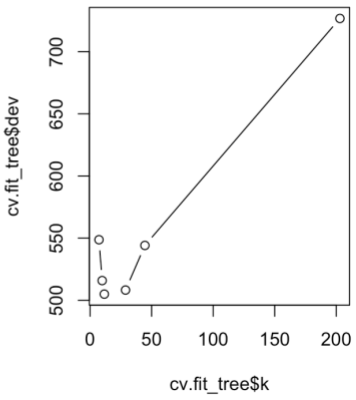
fit_tree = tree(Purchase~., data.train)

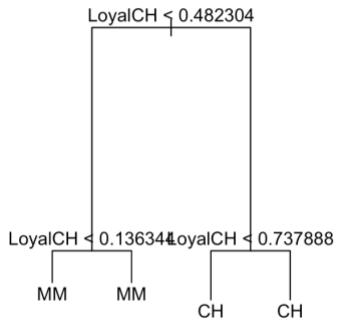


Performing cross validation considering deviance. Best Decision Tree.

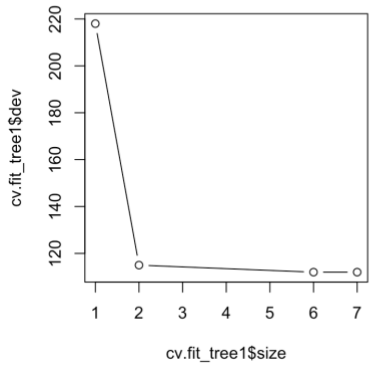
cv.fit_tree = cv.tree(fit_tree)

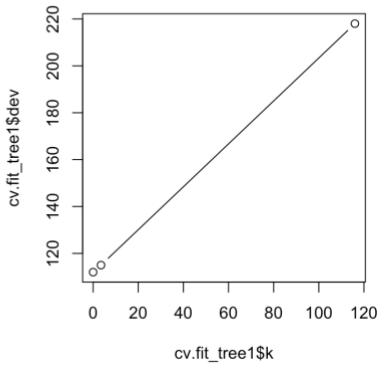


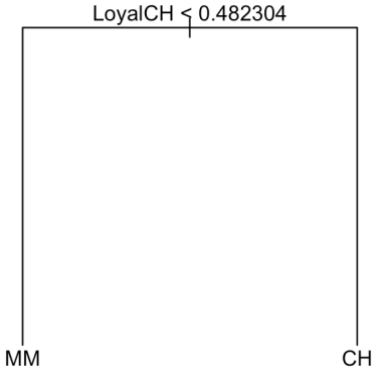




Performing cross validation considering classification error rate. Best Decision Tree.
cv.fit_tree1 = cv.tree(fit_tree, FUN=prune.misclass)







Part e.

```
> table(lda.class,data.train$Purchase)

lda.class  CH  MM
       CH 279  50
       MM  38 168
> mean(lda.class!=data.train$Purchase)
[1] 0.164486
```

Classification Error on the training data = 16.4486%

Part f.

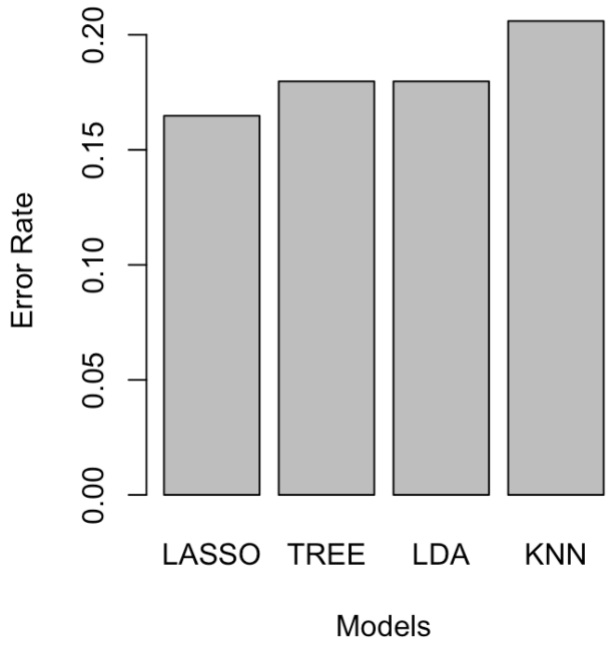
```
> print(best_k)
[1] 7
> orange_avg <- knn(train=orange_norm, test=orange_norm, cl=data.train[,1], k=best_k)
> avg_err=mean(orange_avg!=data.train[,1])
> print(avg_err)
[1] 0.1607477
```

Classification error on the training data = 16.07477%
Best k-value = 7

Part g.

```
> table(pred, y1)
    y1
pred  CH  MM
  CH 146  20
  MM  24  77
> lasso_error = mean(pred!=y1)
> lasso_error
[1] 0.164794
> table(tree.pred1,data.valid$Purchase)

tree.pred1  CH  MM
        CH 143  21
        MM  27  76
> tree_error = mean(tree.pred1!=data.valid$Purchase)
> tree_error
[1] 0.1797753
> table(lda.class1,data.valid$Purchase)

lda.class1  CH  MM
        CH 145  23
        MM  25  74
> lda_error = mean(lda.class1!=data.valid$Purchase)
> lda_error
[1] 0.1797753
> table(orange_avg1,data.valid$Purchase)

orange_avg1  CH  MM
         CH 143  28
         MM  27  69
> knn_avg_err1 = mean(orange_avg1!=data.valid[,1])
> knn_avg_err1
[1] 0.2059925
```

The LASSO model has the lowest error rate.



Part h.

```
> table(final_pred, y3)
          y3
final_pred  CH  MM
        CH 150  25
        MM  16  77
> final_error = mean(final_pred!=y3)
> final_error
[1] 0.1529851
```

Classification Error Rate on the test data using LASSO model = 15.29851%

Part i.

We can calculate the threshold probability using the following equation:

E[Profit] = $3.50*p - $0.50*(1-p) > 0
We get p > 0.125

If you consider the current model output where the threshold p=0.5, the number of predictions of CH using our model on our test data is (150+25=175) and we have a prediction error of 15.30%. We have 150 correct CH predictions and we have 25 wrong CH predictions which are actually MM.
Therefore, the maximum attainable payoff: 150*$3.50 – 25*$0.50 = $512.50