# Test your knowledge of logit & big data analytics

1)
```
heat <- read.csv("Heating.csv")
head(heat)
```
This helps us visualise the first few observations in the dataset.

a)
```
install.packages("mlogit")
library(mlogit)
```
(If mlogit not installed before)

```
data <- mlogit.data(heat, choice = "depvar", shape = "wide", varying = c(3:12))
```
This creates a data object for mlogit to run where depvar is the choice, wide indicates that each row is one choice situation & varying indicates the indexes of variables that are alternative specific

```
model1 <- mlogit(depvar ~ ic + oc -1, data)
```
Here ic & oc are used to predict choice where -1 indicates that no intercepts are needed

i) Both the coefficients of ic & oc are negative which makes sense since as the installation cost & operating cost for a system increases, the probability of choosing that system goes down.

ii) p value for $H_0: \beta_{ic} = 0$ is $< 2.2 \times 10^{-16}$
p value for $H_0: \beta_{oc} = 0$ is $< 2.2 \times 10^{-16}$

These p-values are very close to 0 indicating that we can reject the null hypothesis that the coefficients are 0.

iii) pred1 <- predict (model1, newdata = data)
This predicts the choice probabilities for each choice situation

table (heat $ depvar)/900

| ec | er | gc | gr | hp |
|---|---|---|---|---|
| 0.071 | 0.093 | 0.636 | 0.143 | 0.055 |

These are the observed shares in the data set

apply (pred1, 2, mean)
This computes the average of the choice probabilities & gives:

| ec | er | gc | gr | hp |
|---|---|---|---|---|
| 0.104 | 0.051 | 0.516 | 0.24 | 0.087 |

While the model captures the data reasonably well, there are still differences in the results

0.636/0.516 0.143/0.24 .

iv) model1$coef["oc"] / model1$coef["ic"]

This computes the ratio of the $\beta_{oc}/\beta_{ic} = 0.739$

This implies that the decision makers are willing to pay $0.73 higher in installation cost for 1 $ reduction in operating cost. Note that it seems unreasonable for the decision-maker to pay only 73 cents for a one time payment for a 1$ reduction per year economically.

b) data$lcc <- data$ic + data$oc/0.12

This defines the new set of lifecycle costs in the data object for mlogit

model2 <- mlogit(depvar ~ lcc - 1, data)

The log likelihood = -1248.7 & the variable lcc is significant.

c)

```
model3 <- mlogit(depvar ~ ic + oc, data,
                 reflevel = "hp")
```

This forces hp to be the reference level & the other alternative specific constants are relative to this

```
pred3 <- predict(model3, newdata = data)
apply(pred3, 2, mean)
table(heat$depvar)/900
```

In this case both match exactly

| ec | er | gc | gr | hp |
|---|---|---|---|---|
| 0.0711 | 0.0933 | 0.636 | 0.143 | 0.055 |

The presence of alternative specific constants ensures that the average probabilities equal the average share.

ii) model3 $ coef ["oc"] / model3 $ coef ["ic"]

In this case = 4.56 $ for 1 $ saving in

~~annual~~ operating costs. This seems more reasonable

since operating cost is annual while installation

cost is one time.

iii) model4 ← mlogit (depvar ~ ic + oc, data,
reflevel = "gr")

Here gr is the reference level. Note that

in model 3, the intercept for gr is 0.308.

Hence in the new model, we would reduce

all the alternative specific constants downward

by 0.308. Note this does not change quality

of fit since probabilities are not modified

by adding a constant to all alternatives.

d)

i) data\$iic <- data\$ic/data\$income

models <- mlogit( depvar ~ oc + iic, data )

Note that by summary (models), we note that the new log likelihood value = -1010.2 in comparison to -1008.2 is lower. This is worse in terms of fit since we want to maximize log likelihood. Also in the new model, installation cost divided by income is no longer significant while it was significant in the model in part c).

ii) model6 <- mlogit (depvar ~ oc + ic | income, data )

summary (model6)

This also estimates a coefficient for each alternative that is income dependent. As income rises, probability of choosing heat pump increases (zero coefficient) relative to other. As income rises, probability of choosing gas room drops relative to others (most -ve)

None of these variables are significant however

e) heat1 <- heat
i)
heat1$ic.hp <- 0.9 heat1$ic.hp

data1 <- mlogit.data (heat1, choice = "depvar",
shape = "wide", varying = c(3:12))

We will use model 3 to make predictions

pred3a <- predict(model3, newdata = data1)

apply(pred3a, 2, mean)

| hp | ec | er | gc | gr |
|---|---|---|---|---|
| 0.064 | 0.704 | 0.092 | 0.63 | 0.142 |

These are the predicted shares under the new installation costs.

The market share of heat pumps goes to 0.0645 from 0.055 (6.45 % from 5.5 %)

ii)

```
df <- subset (heat, select = c(3:12))
```

Take only the installation & operating costs for the 5 original alternatives.

Define a new alternative with 2 new costs:

```
df $ ic.eci <- df $ ic.ec + 200
df $ oc.eci <- 0.75 * df $ oc.ec
```

This new alternative (eci) has 200$ more installation cost & 75 % of the operating cost.

We will use model 3 to estimate new choice probabilities.

model 3 $ coef (Provides all coefficients)

We now compute new choice probabilities as follows:

```
df $ hpexp <- exp(model3 $ coef["oc"] * df $ oc.hp
                  + model3 $ coef["ic"] * df $ ic.hp)

df $ ecexp <- exp(model 3 $ coef["oc"] x df $ oc.ec
                  + model 3 $ coef["ic"] x df $ ic.ec
                  + model 3 $ coef["ec:(intercept)"])
```

⋮

Repeat for all 6 alternatives

We now normalize by the sum of exponentials to create the probabilities

```
df$sumexp <- apply(subset(df, select=c(13:17)), 1,
                                        sum)

df$sumnewexp <- apply (subset (df, select= c(13:18)),
                              1, sum)


df$hp <- df$hpexp/( df$sumexp
   :                                          5
df$gr <- df$grexp / df$sumexp

df$hpnew <- df$hpexp / df$sumnewexp
                                              6
df$ecinew <- df$eciexp / df$sumnewexp

df2 <- subset (df, select= c(26:31))
```

This contains new probabilities

```
mktsharenew <- apply(df2, 2, mean)

mktshareold <- apply(predict(model2, data), 2,
                               mean)
```

New: market shares

| hpnew | ec new | ernew | gcnew | grnew | ecinew |
|---|---|---|---|---|---|
| 0.049 | 0.063 | 0.083 | 0.57 | 0.128 | 0.103 |

Old market shares

| hp | ec | er | gc | g |
|---|---|---|---|---|
| 0.055 | 0.071 | 0.093 | 0.63 | 0.143 |

The most market share is drawn from gc from 0.63 to 0.57. This is the gas central system.

Note that from the independence of irrelevant alternatives property, the ratio of market shares remains the same irrespective of other alternatives in the set.

Note that in terms of % drop it draws roughly 10% from each system due to IIA. Note that you would assume the electric system should possibly draw more from electric systems rather than gas systems but this is not the case here.

2)

a) electricity <- read.csv("Electricity.csv")

str(electricity)

This dataframe has 4308 observations with 26 variables.

```
data1 <- mlogit.data(electricity, id.var = "id",
         choice = "choice", varying = c(3:26),
         shape = "wide", sep = "")
```

In this case there is one row for each choice situation.

```
model1 <- mlogit(choice ~ pf + cl + loc + wk
          + tod + seas - 1, data = data1,
          rpar = c(pf = 'n', loc = 'n', wk = 'n',
          tod = 'n', seas = 'n'), panel = TRUE)
```

We use the default settings to solve the mixed logit model with panel data to indicate multiple observations per individual.

summary(model1)

i) The mean coefficient of contract length is around −0.18 indicating consumers prefer shorter contracts. Mean price coefficient is −0.84. Therefore a customer will pay $0.18/0.84 \approx 0.21$ cents per KWh to reduce contract length by 1 year. The numbers could be slightly different for you ii)

The coefficient of length (cl) is normally distributed with mean $\underbrace{-0.18}_{cl}$ and standard deviation $\underbrace{0.31}_{sd.cl}$.

Share of people with negative coefficients $= P(\mu + \sigma z \leq 0)$

$\mu \downarrow -0.18$, $\sigma \downarrow 0.31$, $z \sim N(0,1)$

$= \text{pnorm}\left(-\dfrac{\text{model1\$coef["cl"]}}{\text{model1\$coef["sd.cl"]}}\right)$

$= \underbrace{0.719}$

Roughly 72% of the population dislike long term contracts.

iii) pnorm(-model1$coef["pf"]/model1$coef["sd.pf"])

0.9999998 (very close to 1)

Thus most of the population has negative price coefficients as should be expected.

```
model2 <- mlogit(choice ~ pf + cl + loc + wk +
    tod + season -1, data = data1, rpar =
    c(cl='n', loc='n', wk='n', tod='n',
    seas='n'), panel = TRUE)

summary(model2)
```

The old model has log likelihood of $-4089.6$ while the new model has a log likelihood of $-4110$ (smaller as should be expected)

The estimated value of the price coefficient (pf) is $-0.81$.

c) model3 <- mlogit (choice ~ pf + cl + loc + wk + tod + seas - 1, data = data1, rpar = c(cl = 'n', loc = 'n', wk = 'u', tod = 'n', seas = 'n'), panel = TRUE)

Summary (model3)

From the results we see that, the Uniform distribution of wk is from 0.133 to 2.58 with mean = 1.36.

The estimate price coefficient = $-0.811$

3)

a) Clearly for every value of k, the best subset selection method will have the smallest training sum of squared errors. This finds the global optimum set for each k.

b) This cannot be determined since a low training sum of squared errors does not necessarily translate to low test sum of squared errors.

c)

1) True. Since in forward stepwise selection at each step, we add only 1 variable to the previous set in an optimal manner.

2) True. Since you drop a variable from the set in backward stepwise selection

3) False

4) False

5) False

4)

a) College <- read.csv("college.csv")

Set.seed(1)

trainid <- Sample(1: nrow(College), nrow(College)*0.8)

testid <- -trainid

train <- College[trainid, ]

test <- College[testid, ]

Str(train) }
Str(test) }

621 observations in training set and 156 observations in test set.

b) model1 <- lm(Apps~. , data = train)

Summary(model1)

The total sum of squared error for the training set is obtained as

Sum(model1$residual$^2$)

The average sum of squared error is

mean(model1$residual$^2$) = 1061946.

predict1 <- predict (model1, newdata = test)

msetest1 <- mean ( (test$Apps - predict1)$^2$ )

This gives the test mean squared error

= 1075064.

c) library (leaps)

model2 <- regsubsets (Apps~., data = train,
~~nvmax~~ nvmax = 17,
method = "backward")

Summary (model2)

We see from the result that the subset of size 16 is obtained by dropping <u>P.Undergrad</u>.

~~[illegible]~~

d) Summary (model2) $adjr2

plot (summary (model2) $adjr2)

which.max (summary (model2) $ adjr2)  12

Summary (model2) $ which

In addition to the intercept, we would include Private Yes, Accept, Enroll, Top10perc, Top25perc, F.Undergrad, Outstate, Room.Board, PhD, Terminal, S.F.Ratio, Expend, Grad.Rate

We would drop P.Undergrad, Books, Terminal, perc.alumni, Personal.

e) model3 ~ lm(Apps ~ Private + Accept + Enroll + Top10perc + Top25perc + F.Undergrad + Outstate + Room.Board + PhD + S.F.Ratio + Expend + Grad.Rate, data = train)

Summary(model3)

This gives all the coefficient estimated values.

Predict3 <- predict(model3, newdata = test)

mse3 <- mean((test$Apps - predict3)^2)

This gives a test mean squared error = 1070293.

Since this error is less than 1075064, it does improve on the accuracy.

f) library (glmnet)

```
grid <- 10^ seq (10, -2, length = 100)
x <- model.matrix (Apps~. , College)
y <- College $ Apps
```

~~set.seed~~(1)

```
model4 <- glmnet (x[train.id, ], y[train.id],
                  lambda = grid )

plot(model4, xvar = "lambda")
```

g) set. seed (1)

```
cvmodel 4 <- cv.glmnet ( x[train.id, ], y [train.id],
                         nfolds = 10, lambda = grid)

cvmodel 4 $ lambda.min
```

This gives $\lambda = 0.497$

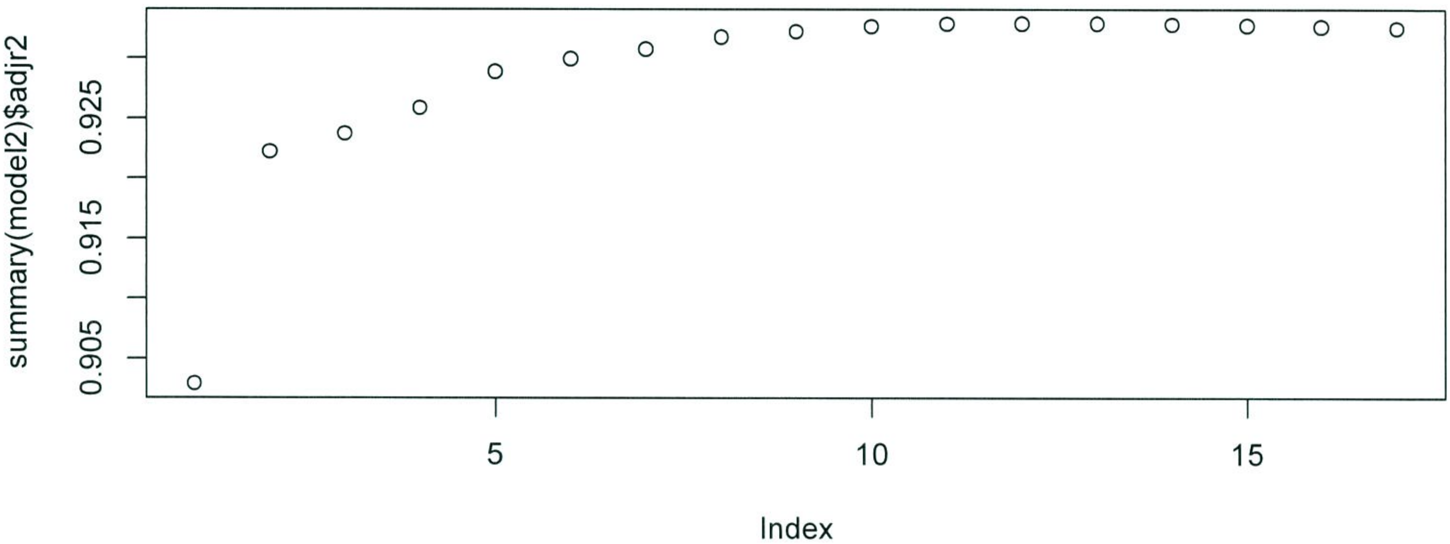
signif (cvmodel4 $ lambda, 4) Only plots to 4 significant digits

cvmodel4 $nzero

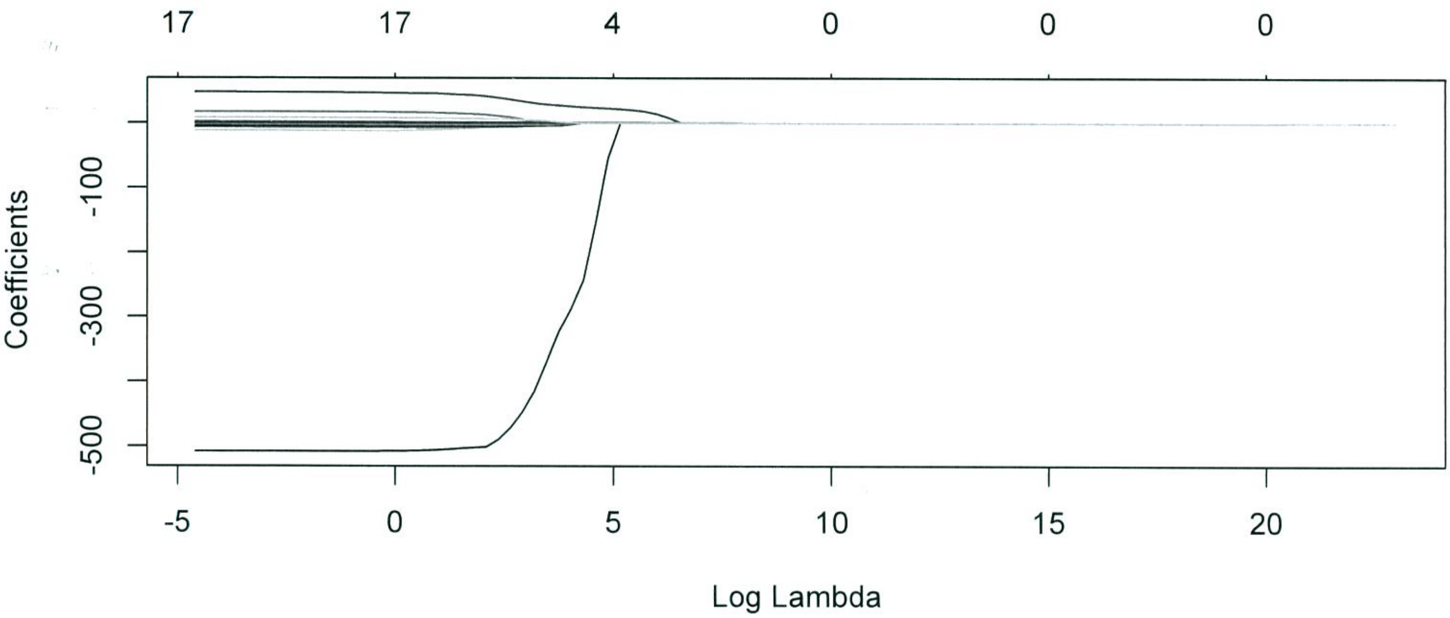
There are 17 nonzeros complete model.

Here LASSO gives the full model.

Test error will be same as model 1.



summary(model2)$adjr2
0.905
0.915
0.925
5
10
15
Index



17 17 4 0 0 0
Coefficients
-100
-300
-500
-5 0 5 10 15 20
Log Lambda