Final

Cheng

6/20/2019

d = read.csv("final 1.csv", header = TRUE, sep = ",")   
# take a look of data  
summary(d)

## age job marital   
## Min. :17.00 admin. :10362 divorced: 4598   
## 1st Qu.:32.00 blue-collar: 9207 married :24798   
## Median :38.00 technician : 6707 single :11512   
## Mean :40.03 services : 3951 unknown : 80   
## 3rd Qu.:47.00 management : 2912   
## Max. :98.00 retired : 1717   
## (Other) : 6132   
## education default housing   
## university.degree :12102 no :32430 no :18527   
## high.school : 9466 unknown: 8555 unknown: 988   
## basic.9y : 6023 yes : 3 yes :21473   
## professional.course: 5219   
## basic.4y : 4158   
## basic.6y : 2278   
## (Other) : 1742   
## loan contact month day\_of\_week  
## no :33785 cellular :26020 may :13708 fri:7791   
## unknown: 988 telephone:14968 jul : 7138 mon:8476   
## yes : 6215 aug : 6142 thu:8572   
## jun : 5288 tue:8055   
## nov : 4090 wed:8094   
## apr : 2616   
## (Other): 2006   
## duration campaign pdays previous   
## Min. : 0.0 Min. : 1.000 Min. : 0.0 Min. :0.0000   
## 1st Qu.: 102.0 1st Qu.: 1.000 1st Qu.:999.0 1st Qu.:0.0000   
## Median : 180.0 Median : 2.000 Median :999.0 Median :0.0000   
## Mean : 258.2 Mean : 2.568 Mean :962.4 Mean :0.1728   
## 3rd Qu.: 319.0 3rd Qu.: 3.000 3rd Qu.:999.0 3rd Qu.:0.0000   
## Max. :4918.0 Max. :56.000 Max. :999.0 Max. :7.0000   
##   
## poutcome emp.var.rate cons.price.idx cons.conf.idx   
## failure : 4222 Min. :-3.40000 Min. :92.20 Min. :-50.8   
## nonexistent:35396 1st Qu.:-1.80000 1st Qu.:93.08 1st Qu.:-42.7   
## success : 1370 Median : 1.10000 Median :93.75 Median :-41.8   
## Mean : 0.08144 Mean :93.58 Mean :-40.5   
## 3rd Qu.: 1.40000 3rd Qu.:93.99 3rd Qu.:-36.4   
## Max. : 1.40000 Max. :94.77 Max. :-26.9   
##   
## tw3m nr.employed y   
## Min. :0.634 Min. :4964 no :36373   
## 1st Qu.:1.344 1st Qu.:5099 yes: 4615   
## Median :4.857 Median :5191   
## Mean :3.621 Mean :5167   
## 3rd Qu.:4.961 3rd Qu.:5228   
## Max. :5.045 Max. :5228   
##

data preprocessing

d$y2 = as.integer(d$y)   
# categorical:  
d[, c("job", "marital", "education", "default", "housing", "loan", "contact", "month", "day\_of\_week", "poutcome")] =  
 lapply(d[, c("job", "marital", "education", "default", "housing", "loan", "contact", "month", "day\_of\_week", "poutcome")], factor)  
  
#standardize numreical variables: ("age", "duration", "campaign", "pdays", "previous", "emp.var.rate", "cons.price.idx", "cons.conf.idx", "taiwan3m", "nr.employed")  
d = d %>%  
 mutate(age\_std = as.vector(scale(age))) %>%  
 mutate(duration\_std = as.vector(scale(duration))) %>%  
 mutate(pdays\_std = as.vector(scale(pdays))) %>%  
 mutate(previous\_std = as.vector(scale(previous))) %>%  
 mutate(emp.var.raten\_std = as.vector(scale(emp.var.rate))) %>%  
 mutate(cons.price.idx\_std = as.vector(scale(cons.price.idx))) %>%  
 mutate(cons.conf.idx\_std = as.vector(scale(cons.conf.idx))) %>%  
 mutate(tw3m\_std = as.vector(scale(tw3m))) %>%  
 mutate(nr.employed\_std = as.vector(scale(nr.employed)))

create dummy

d2 = fastDummies::dummy\_cols(d) # with dummy  
head(d2)

## age job marital education default housing loan contact month  
## 1 56 housemaid married basic.4y no no no telephone may  
## 2 57 services married high.school unknown no no telephone may  
## 3 37 services married high.school no yes no telephone may  
## 4 40 admin. married basic.6y no no no telephone may  
## 5 56 services married high.school no no yes telephone may  
## 6 45 services married basic.9y unknown no no telephone may  
## day\_of\_week duration campaign pdays previous poutcome emp.var.rate  
## 1 mon 261 1 999 0 nonexistent 1.1  
## 2 mon 149 1 999 0 nonexistent 1.1  
## 3 mon 226 1 999 0 nonexistent 1.1  
## 4 mon 151 1 999 0 nonexistent 1.1  
## 5 mon 307 1 999 0 nonexistent 1.1  
## 6 mon 198 1 999 0 nonexistent 1.1  
## cons.price.idx cons.conf.idx tw3m nr.employed y y2 age\_std  
## 1 93.994 -36.4 4.857 5191 no 1 1.532148271  
## 2 93.994 -36.4 4.857 5191 no 1 1.628076045  
## 3 93.994 -36.4 4.857 5191 no 1 -0.290479450  
## 4 93.994 -36.4 4.857 5191 no 1 -0.002696126  
## 5 93.994 -36.4 4.857 5191 no 1 1.532148271  
## 6 93.994 -36.4 4.857 5191 no 1 0.476942748  
## duration\_std pdays\_std previous\_std emp.var.raten\_std cons.price.idx\_std  
## 1 0.01070184 0.19557 -0.3493102 0.6484087 0.7228814  
## 2 -0.42160726 0.19557 -0.3493102 0.6484087 0.7228814  
## 3 -0.12439475 0.19557 -0.3493102 0.6484087 0.7228814  
## 4 -0.41388745 0.19557 -0.3493102 0.6484087 0.7228814  
## 5 0.18825736 0.19557 -0.3493102 0.6484087 0.7228814  
## 6 -0.23247203 0.19557 -0.3493102 0.6484087 0.7228814  
## cons.conf.idx\_std tw3m\_std nr.employed\_std job\_housemaid job\_services  
## 1 0.8865058 0.7126401 0.3319672 1 0  
## 2 0.8865058 0.7126401 0.3319672 0 1  
## 3 0.8865058 0.7126401 0.3319672 0 1  
## 4 0.8865058 0.7126401 0.3319672 0 0  
## 5 0.8865058 0.7126401 0.3319672 0 1  
## 6 0.8865058 0.7126401 0.3319672 0 1  
## job\_admin. job\_blue-collar job\_technician job\_retired job\_management  
## 1 0 0 0 0 0  
## 2 0 0 0 0 0  
## 3 0 0 0 0 0  
## 4 1 0 0 0 0  
## 5 0 0 0 0 0  
## 6 0 0 0 0 0  
## job\_unemployed job\_self-employed job\_unknown job\_entrepreneur  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## job\_student marital\_married marital\_single marital\_divorced  
## 1 0 1 0 0  
## 2 0 1 0 0  
## 3 0 1 0 0  
## 4 0 1 0 0  
## 5 0 1 0 0  
## 6 0 1 0 0  
## marital\_unknown education\_basic.4y education\_high.school  
## 1 0 1 0  
## 2 0 0 1  
## 3 0 0 1  
## 4 0 0 0  
## 5 0 0 1  
## 6 0 0 0  
## education\_basic.6y education\_basic.9y education\_professional.course  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 1 0 0  
## 5 0 0 0  
## 6 0 1 0  
## education\_unknown education\_university.degree education\_illiterate  
## 1 0 0 0  
## 2 0 0 0  
## 3 0 0 0  
## 4 0 0 0  
## 5 0 0 0  
## 6 0 0 0  
## default\_no default\_unknown default\_yes housing\_no housing\_yes  
## 1 1 0 0 1 0  
## 2 0 1 0 1 0  
## 3 1 0 0 0 1  
## 4 1 0 0 1 0  
## 5 1 0 0 1 0  
## 6 0 1 0 1 0  
## housing\_unknown loan\_no loan\_yes loan\_unknown contact\_telephone  
## 1 0 1 0 0 1  
## 2 0 1 0 0 1  
## 3 0 1 0 0 1  
## 4 0 1 0 0 1  
## 5 0 0 1 0 1  
## 6 0 1 0 0 1  
## contact\_cellular month\_may month\_jun month\_jul month\_aug month\_oct  
## 1 0 1 0 0 0 0  
## 2 0 1 0 0 0 0  
## 3 0 1 0 0 0 0  
## 4 0 1 0 0 0 0  
## 5 0 1 0 0 0 0  
## 6 0 1 0 0 0 0  
## month\_nov month\_dec month\_mar month\_apr month\_sep day\_of\_week\_mon  
## 1 0 0 0 0 0 1  
## 2 0 0 0 0 0 1  
## 3 0 0 0 0 0 1  
## 4 0 0 0 0 0 1  
## 5 0 0 0 0 0 1  
## 6 0 0 0 0 0 1  
## day\_of\_week\_tue day\_of\_week\_wed day\_of\_week\_thu day\_of\_week\_fri  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## poutcome\_nonexistent poutcome\_failure poutcome\_success y\_no y\_yes  
## 1 1 0 0 1 0  
## 2 1 0 0 1 0  
## 3 1 0 0 1 0  
## 4 1 0 0 1 0  
## 5 1 0 0 1 0  
## 6 1 0 0 1 0

drop outcome and duration according to the notation

x\_train = dplyr::select(d2,-c(y, y2, duration, y\_no, y\_yes))   
y\_train = dplyr::select(d2, y2)

(Note that since this dataset would be high-dimensionl after dummy and there are 40988 observations which is a large dataset for my laptop’s capicty, so I would like to just run numerical and few categorical variables in this case.. though it may lose some information… )

x\_train2 = sample\_n(x\_train, 1000)  
y\_train2 = sample\_n(y\_train, 1000)

# 1) Part 1 (20%).

create a single-level model using logistic regression select only numerical data

x\_train0 = dplyr::select(x\_train2, c( "age", "campaign"))   
y\_train0 = y\_train2

m.single <- "  
 data {  
 int K;   
 int N;   
 int N2;  
 int D;   
 int y[N];   
 vector[D] x[N];   
 vector[D] x\_test[N2];   
 }  
   
 parameters {  
 matrix[K, D] beta;   
 }  
   
 model {  
  
 // prior for beta   
 for (c in 1:K)  
 beta[c] ~ normal(0,5);  
   
 // likelihood of outcome  
 for (i in 1:N)  
 y[i] ~ categorical\_logit(beta \* x[i]); //softmax  
   
 }  
 generated quantities{  
 vector[N] log\_lik;  
 vector[N2] output;  
   
 for(i in 1:N2){  
 log\_lik[i] = categorical\_logit\_lpmf(y[i] | beta \* x[i]);  
 output[i] = categorical\_logit\_rng(beta \* x\_test[i]);  
 }  
 }  
 "  
  
mod0 <- stan\_model(model\_code = m.single)

dat <- list(  
 N = nrow(x\_train0),  
 K = 2,  
 D = ncol(x\_train0),  
 y = y\_train0$y2,  
 x = x\_train0,  
 N2 = nrow(x\_train0),  
 x\_test = x\_train0 # for backtesting  
)  
  
fit0 = vb(mod0, data = dat, iter = 20000)

## Chain 1: ------------------------------------------------------------  
## Chain 1: EXPERIMENTAL ALGORITHM:  
## Chain 1: This procedure has not been thoroughly tested and may be unstable  
## Chain 1: or buggy. The interface is subject to change.  
## Chain 1: ------------------------------------------------------------  
## Chain 1:   
## Chain 1:   
## Chain 1:   
## Chain 1: Gradient evaluation took 0.001052 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10.52 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Begin eta adaptation.  
## Chain 1: Iteration: 1 / 250 [ 0%] (Adaptation)  
## Chain 1: Iteration: 50 / 250 [ 20%] (Adaptation)  
## Chain 1: Iteration: 100 / 250 [ 40%] (Adaptation)  
## Chain 1: Iteration: 150 / 250 [ 60%] (Adaptation)  
## Chain 1: Iteration: 200 / 250 [ 80%] (Adaptation)  
## Chain 1: Iteration: 250 / 250 [100%] (Adaptation)  
## Chain 1: Success! Found best value [eta = 0.1].  
## Chain 1:   
## Chain 1: Begin stochastic gradient ascent.  
## Chain 1: iter ELBO delta\_ELBO\_mean delta\_ELBO\_med notes   
## Chain 1: 100 -6827.766 1.000 1.000  
## Chain 1: 200 -4349.841 0.785 1.000  
## Chain 1: 300 -3701.175 0.582 0.570  
## Chain 1: 400 -3195.454 0.476 0.570  
## Chain 1: 500 -2455.520 0.441 0.301  
## Chain 1: 600 -2217.677 0.385 0.301  
## Chain 1: 700 -2165.384 0.334 0.175  
## Chain 1: 800 -2055.189 0.299 0.175  
## Chain 1: 900 -1756.160 0.284 0.170  
## Chain 1: 1000 -1660.963 0.262 0.170  
## Chain 1: 1100 -1626.900 0.240 0.158  
## Chain 1: 1200 -1663.917 0.222 0.158  
## Chain 1: 1300 -1207.646 0.234 0.158  
## Chain 1: 1400 -1079.866 0.225 0.158  
## Chain 1: 1500 -1069.881 0.211 0.118  
## Chain 1: 1600 -929.941 0.207 0.150  
## Chain 1: 1700 -936.696 0.195 0.118  
## Chain 1: 1800 -892.777 0.187 0.118  
## Chain 1: 1900 -944.786 0.180 0.107  
## Chain 1: 2000 -1046.692 0.176 0.107  
## Chain 1: 2100 -804.825 0.141 0.107  
## Chain 1: 2200 -759.057 0.116 0.097  
## Chain 1: 2300 -724.945 0.109 0.060  
## Chain 1: 2400 -719.279 0.102 0.057  
## Chain 1: 2500 -626.991 0.094 0.057  
## Chain 1: 2600 -545.963 0.096 0.057  
## Chain 1: 2700 -558.526 0.096 0.057  
## Chain 1: 2800 -557.154 0.094 0.057  
## Chain 1: 2900 -522.472 0.088 0.057  
## Chain 1: 3000 -559.908 0.089 0.060  
## Chain 1: 3100 -501.977 0.094 0.066  
## Chain 1: 3200 -519.502 0.094 0.066  
## Chain 1: 3300 -517.771 0.075 0.060  
## Chain 1: 3400 -452.331 0.077 0.060  
## Chain 1: 3500 -473.728 0.079 0.060  
## Chain 1: 3600 -434.421 0.076 0.060  
## Chain 1: 3700 -444.932 0.076 0.060  
## Chain 1: 3800 -445.192 0.074 0.060  
## Chain 1: 3900 -431.361 0.073 0.060  
## Chain 1: 4000 -425.266 0.069 0.047  
## Chain 1: 4100 -423.420 0.054 0.045  
## Chain 1: 4200 -415.316 0.052 0.034  
## Chain 1: 4300 -417.823 0.050 0.032  
## Chain 1: 4400 -409.219 0.050 0.032  
## Chain 1: 4500 -413.874 0.044 0.024  
## Chain 1: 4600 -406.371 0.037 0.022  
## Chain 1: 4700 -419.754 0.038 0.024  
## Chain 1: 4800 -401.195 0.040 0.032  
## Chain 1: 4900 -400.655 0.037 0.024  
## Chain 1: 5000 -401.026 0.033 0.021  
## Chain 1: 5100 -397.862 0.028 0.020  
## Chain 1: 5200 -394.853 0.027 0.018  
## Chain 1: 5300 -394.091 0.026 0.018  
## Chain 1: 5400 -397.213 0.020 0.014  
## Chain 1: 5500 -392.616 0.018 0.012  
## Chain 1: 5600 -392.147 0.013 0.011  
## Chain 1: 5700 -392.040 0.012 0.008 MEDIAN ELBO CONVERGED  
## Chain 1:   
## Chain 1: Drawing a sample of size 1000 from the approximate posterior...   
## Chain 1: COMPLETED.

post0 <- as.data.frame(fit0)

# 2) Part 2 (20%).

Build up an advanced statistical model (option: monster, mixture, ormultilevel), with the following data set: final 1.csv

put all the variables including numerical and all dummy variables

m.logit <- "  
 data {  
 int K;   
 int N;   
 int N2;  
 int D;   
 int y[N];   
 vector[D] x[N];   
 vector[D] x\_test[N2];   
 }  
   
 parameters {  
 matrix[K, D] beta;   
 }  
   
 model {  
  
 // prior for beta   
 for (c in 1:K)  
 beta[c] ~ normal(0,5);  
   
 // likelihood of outcome  
 for (i in 1:N)  
 y[i] ~ categorical\_logit(beta \* x[i]); //softmax  
   
 }  
 generated quantities{  
 vector[N] log\_lik;  
 vector[N2] output;  
   
 for(i in 1:N2){  
 log\_lik[i] = categorical\_logit\_lpmf(y[i] | beta \* x[i]);  
 output[i] = categorical\_logit\_rng(beta \* x\_test[i]);  
 }  
 }  
 "  
  
mod <- stan\_model(model\_code = m.logit)

dat <- list(  
 N = nrow(x\_train2),  
 K = 2,  
 D = ncol(x\_train2),  
 y = y\_train2$y2,  
 x = x\_train2,  
 N2 = nrow(x\_train2),  
 x\_test = x\_train2 # for backtesting  
)  
  
fit1 = vb(mod, data = dat, iter = 10000)

## Chain 1: ------------------------------------------------------------  
## Chain 1: EXPERIMENTAL ALGORITHM:  
## Chain 1: This procedure has not been thoroughly tested and may be unstable  
## Chain 1: or buggy. The interface is subject to change.  
## Chain 1: ------------------------------------------------------------  
## Chain 1:   
## Chain 1:   
## Chain 1:   
## Chain 1: Gradient evaluation took 0.002413 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 24.13 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Begin eta adaptation.  
## Chain 1: Iteration: 1 / 250 [ 0%] (Adaptation)  
## Chain 1: Iteration: 50 / 250 [ 20%] (Adaptation)  
## Chain 1: Iteration: 100 / 250 [ 40%] (Adaptation)  
## Chain 1: Iteration: 150 / 250 [ 60%] (Adaptation)  
## Chain 1: Iteration: 200 / 250 [ 80%] (Adaptation)  
## Chain 1: Success! Found best value [eta = 1] earlier than expected.  
## Chain 1:   
## Chain 1: Begin stochastic gradient ascent.  
## Chain 1: iter ELBO delta\_ELBO\_mean delta\_ELBO\_med notes   
## Chain 1: 100 -257367.660 1.000 1.000  
## Chain 1: 200 -154808.636 0.831 1.000  
## Chain 1: 300 -76634.390 0.894 1.000  
## Chain 1: 400 -126483.962 0.769 1.000  
## Chain 1: 500 -78061.030 0.739 0.662  
## Chain 1: 600 -144440.844 0.693 0.662  
## Chain 1: 700 -57425.449 0.810 0.662  
## Chain 1: 800 -26466.126 0.855 1.000  
## Chain 1: 900 -118224.965 0.846 0.776  
## Chain 1: 1000 -76102.284 0.817 0.776  
## Chain 1: 1100 -43724.316 0.791 0.741 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1200 -88278.997 0.775 0.741 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1300 -100635.336 0.686 0.620 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1400 -23078.556 0.982 0.741 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1500 -18408.184 0.946 0.741 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1600 -24813.024 0.926 0.741 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1700 -70940.185 0.839 0.650 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1800 -42888.456 0.787 0.650 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 1900 -50664.950 0.725 0.554 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2000 -23352.864 0.787 0.650 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2100 -24402.691 0.717 0.505 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2200 -15727.649 0.722 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2300 -18555.620 0.725 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2400 -45339.932 0.448 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2500 -59085.824 0.446 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2600 -47126.016 0.445 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2700 -13319.752 0.634 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2800 -24452.570 0.614 0.455 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 2900 -59551.960 0.658 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3000 -52029.867 0.555 0.455 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3100 -19742.100 0.714 0.552 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3200 -22304.812 0.671 0.455 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3300 -13511.150 0.721 0.589 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3400 -23426.376 0.704 0.455 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3500 -39479.070 0.721 0.455 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3600 -22113.359 0.774 0.589 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3700 -57017.328 0.582 0.589 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3800 -11045.345 0.952 0.612 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 3900 -39768.421 0.966 0.651 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4000 -44622.286 0.962 0.651 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4100 -19998.957 0.922 0.651 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4200 -18432.612 0.919 0.651 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4300 -39727.475 0.907 0.612 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4400 -11450.157 1.112 0.722 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4500 -29231.339 1.132 0.722 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4600 -16742.818 1.128 0.722 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4700 -16155.893 1.071 0.722 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4800 -34208.146 0.707 0.608 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 4900 -26957.331 0.662 0.536 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5000 -18700.061 0.695 0.536 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5100 -15059.144 0.596 0.528 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5200 -30765.592 0.639 0.528 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5300 -16480.690 0.672 0.528 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5400 -11410.375 0.469 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5500 -8474.607 0.443 0.444  
## Chain 1: 5600 -30274.467 0.440 0.444  
## Chain 1: 5700 -20036.149 0.488 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5800 -11549.582 0.509 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 5900 -14045.270 0.499 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6000 -42955.171 0.523 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6100 -8059.520 0.931 0.673 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6200 -8489.152 0.885 0.673 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6300 -9164.721 0.806 0.511 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6400 -44814.113 0.841 0.673 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6500 -9510.287 1.178 0.720 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6600 -7900.191 1.126 0.673 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6700 -11029.380 1.103 0.673 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6800 -14750.648 1.055 0.284 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 6900 -40842.729 1.101 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7000 -15506.682 1.197 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7100 -9098.077 0.835 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7200 -8111.673 0.842 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7300 -15147.408 0.881 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7400 -7902.343 0.893 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7500 -6415.016 0.545 0.464 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7600 -19384.116 0.592 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7700 -27457.889 0.593 0.639 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7800 -6616.518 0.882 0.669 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 7900 -11695.711 0.862 0.669 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8000 -11399.039 0.701 0.464 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8100 -33032.853 0.696 0.464 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8200 -21429.818 0.738 0.541 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8300 -14833.147 0.736 0.541 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8400 -19274.606 0.668 0.445 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8500 -13788.734 0.684 0.445 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8600 -27255.182 0.667 0.445 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8700 -5077.866 1.074 0.494 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8800 -23939.821 0.838 0.494 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 8900 -11258.695 0.907 0.541 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9000 -35921.164 0.973 0.655 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9100 -4645.839 1.581 0.687 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9200 -12373.386 1.589 0.687 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9300 -8601.324 1.589 0.687 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9400 -34491.243 1.641 0.751 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9500 -27630.067 1.626 0.751 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9600 -9665.786 1.762 0.788 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9700 -6513.702 1.374 0.751 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9800 -9954.898 1.329 0.687 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 9900 -13253.342 1.242 0.625 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: 10000 -31162.469 1.231 0.575 MAY BE DIVERGING... INSPECT ELBO  
## Chain 1: Informational Message: The maximum number of iterations is reached! The algorithm may not have converged.  
## Chain 1: This variational approximation is not guaranteed to be meaningful.  
## Chain 1:   
## Chain 1: Drawing a sample of size 1000 from the approximate posterior...   
## Chain 1: COMPLETED.

post1 <- as.data.frame(fit1)

# 3) Part 3 (20%).

Model comparison: calculate WAIC of the two models developed.

log\_lik\_1.1 = extract\_log\_lik(fit0, merge\_chains = FALSE)  
log\_lik\_1.2 = extract\_log\_lik(fit1, merge\_chains = FALSE)  
(waic\_1.1 = waic(log\_lik\_1.1))

##   
## Computed from 1000 by 1000 log-likelihood matrix  
##   
## Estimate SE  
## elpd\_waic -384.9 25.4  
## p\_waic 24.8 3.2  
## waic 769.7 50.8

(waic\_1.2 = waic(log\_lik\_1.2))

##   
## Computed from 1000 by 1000 log-likelihood matrix  
##   
## Estimate SE  
## elpd\_waic -40687.9 3751.7  
## p\_waic 13314.3 1480.4  
## waic 81375.7 7503.4

mod\_comp <- loo::compare(waic\_1.1, waic\_1.2)  
mod\_comp

## elpd\_diff se   
## -40303.0 3728.1

# 4) Part 4 (20%).

Make an ensemble model by combining the two models developed.

# 5) Part 5 (20%).

After answering the questions above, check the estimation accuracy using the following data set: final 2.csv. Specifically, use the three models developed to examine whether they make acceptable prediction performance via calculating the correct rate of these three models, respectively.

d\_test = read.csv("final 2.csv", header = TRUE, sep = ",")   
d\_test$y2 = as.integer(d\_test$y)

x\_test = dplyr::select(d\_test, c( "age", "campaign"))   
y\_test = d\_test$y2

dat <- list(  
 N = nrow(x\_train0),  
 K = 2,  
 D = ncol(x\_train0),  
 y = y\_train0$y2,  
 x = x\_train0,  
 N2 = nrow(x\_test),  
 x\_test = x\_test # for backtesting  
)  
  
fit3 = vb(mod0, data = dat, iter = 20000)

## Chain 1: ------------------------------------------------------------  
## Chain 1: EXPERIMENTAL ALGORITHM:  
## Chain 1: This procedure has not been thoroughly tested and may be unstable  
## Chain 1: or buggy. The interface is subject to change.  
## Chain 1: ------------------------------------------------------------  
## Chain 1:   
## Chain 1:   
## Chain 1:   
## Chain 1: Gradient evaluation took 0.000651 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 6.51 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Begin eta adaptation.  
## Chain 1: Iteration: 1 / 250 [ 0%] (Adaptation)  
## Chain 1: Iteration: 50 / 250 [ 20%] (Adaptation)  
## Chain 1: Iteration: 100 / 250 [ 40%] (Adaptation)  
## Chain 1: Iteration: 150 / 250 [ 60%] (Adaptation)  
## Chain 1: Iteration: 200 / 250 [ 80%] (Adaptation)  
## Chain 1: Success! Found best value [eta = 1] earlier than expected.  
## Chain 1:   
## Chain 1: Begin stochastic gradient ascent.  
## Chain 1: iter ELBO delta\_ELBO\_mean delta\_ELBO\_med notes   
## Chain 1: 100 -2718.264 1.000 1.000  
## Chain 1: 200 -1081.267 1.257 1.514  
## Chain 1: 300 -743.037 0.990 1.000  
## Chain 1: 400 -722.080 0.750 1.000  
## Chain 1: 500 -527.452 0.673 0.455  
## Chain 1: 600 -389.445 0.620 0.455  
## Chain 1: 700 -441.553 0.549 0.369  
## Chain 1: 800 -412.849 0.489 0.369  
## Chain 1: 900 -387.716 0.442 0.354  
## Chain 1: 1000 -494.645 0.419 0.354  
## Chain 1: 1100 -430.428 0.394 0.216  
## Chain 1: 1200 -385.431 0.371 0.216  
## Chain 1: 1300 -559.215 0.367 0.216  
## Chain 1: 1400 -489.237 0.351 0.216  
## Chain 1: 1500 -384.009 0.346 0.216  
## Chain 1: 1600 -403.395 0.327 0.216  
## Chain 1: 1700 -398.915 0.308 0.149  
## Chain 1: 1800 -384.081 0.293 0.149  
## Chain 1: 1900 -385.016 0.278 0.143  
## Chain 1: 2000 -448.460 0.271 0.143  
## Chain 1: 2100 -390.766 0.229 0.143  
## Chain 1: 2200 -383.743 0.154 0.141  
## Chain 1: 2300 -434.214 0.137 0.118  
## Chain 1: 2400 -408.931 0.139 0.118  
## Chain 1: 2500 -395.830 0.122 0.117  
## Chain 1: 2600 -433.201 0.108 0.116  
## Chain 1: 2700 -383.680 0.109 0.116  
## Chain 1: 2800 -391.295 0.106 0.116  
## Chain 1: 2900 -385.363 0.104 0.116  
## Chain 1: 3000 -492.139 0.104 0.116  
## Chain 1: 3100 -384.585 0.111 0.116  
## Chain 1: 3200 -421.701 0.109 0.088  
## Chain 1: 3300 -384.692 0.098 0.088  
## Chain 1: 3400 -574.295 0.108 0.088  
## Chain 1: 3500 -510.573 0.100 0.088  
## Chain 1: 3600 -446.752 0.105 0.096  
## Chain 1: 3700 -541.283 0.113 0.116  
## Chain 1: 3800 -400.570 0.129 0.125  
## Chain 1: 3900 -390.798 0.130 0.125  
## Chain 1: 4000 -386.314 0.123 0.116  
## Chain 1: 4100 -390.378 0.117 0.096  
## Chain 1: 4200 -389.178 0.116 0.096  
## Chain 1: 4300 -489.274 0.120 0.096  
## Chain 1: 4400 -459.213 0.120 0.096  
## Chain 1: 4500 -385.607 0.128 0.125  
## Chain 1: 4600 -405.517 0.126 0.125  
## Chain 1: 4700 -474.872 0.127 0.125  
## Chain 1: 4800 -464.100 0.127 0.125  
## Chain 1: 4900 -392.976 0.136 0.143  
## Chain 1: 5000 -422.499 0.128 0.125  
## Chain 1: 5100 -386.786 0.119 0.096  
## Chain 1: 5200 -385.547 0.115 0.096  
## Chain 1: 5300 -385.156 0.110 0.092  
## Chain 1: 5400 -395.234 0.095 0.070  
## Chain 1: 5500 -397.073 0.089 0.065  
## Chain 1: 5600 -419.269 0.084 0.053  
## Chain 1: 5700 -390.524 0.079 0.053  
## Chain 1: 5800 -384.754 0.062 0.049  
## Chain 1: 5900 -412.007 0.064 0.053  
## Chain 1: 6000 -450.912 0.068 0.065  
## Chain 1: 6100 -386.424 0.076 0.066  
## Chain 1: 6200 -411.676 0.079 0.066  
## Chain 1: 6300 -389.816 0.072 0.065  
## Chain 1: 6400 -429.196 0.073 0.066  
## Chain 1: 6500 -412.308 0.065 0.061  
## Chain 1: 6600 -386.035 0.066 0.066  
## Chain 1: 6700 -387.002 0.059 0.061  
## Chain 1: 6800 -386.174 0.058 0.061  
## Chain 1: 6900 -384.171 0.049 0.056  
## Chain 1: 7000 -396.090 0.047 0.053  
## Chain 1: 7100 -386.221 0.044 0.041  
## Chain 1: 7200 -390.425 0.044 0.041  
## Chain 1: 7300 -383.743 0.045 0.041  
## Chain 1: 7400 -399.979 0.046 0.041  
## Chain 1: 7500 -407.800 0.047 0.041  
## Chain 1: 7600 -401.477 0.045 0.041  
## Chain 1: 7700 -412.064 0.042 0.030  
## Chain 1: 7800 -448.279 0.046 0.041  
## Chain 1: 7900 -396.022 0.049 0.041  
## Chain 1: 8000 -385.701 0.046 0.030  
## Chain 1: 8100 -386.287 0.038 0.027  
## Chain 1: 8200 -408.743 0.037 0.027  
## Chain 1: 8300 -385.269 0.038 0.027  
## Chain 1: 8400 -390.742 0.034 0.026  
## Chain 1: 8500 -406.763 0.034 0.026  
## Chain 1: 8600 -390.469 0.032 0.026  
## Chain 1: 8700 -387.614 0.033 0.026  
## Chain 1: 8800 -396.326 0.034 0.026  
## Chain 1: 8900 -385.395 0.035 0.027  
## Chain 1: 9000 -402.477 0.035 0.027  
## Chain 1: 9100 -385.145 0.036 0.028  
## Chain 1: 9200 -383.487 0.036 0.028  
## Chain 1: 9300 -386.820 0.036 0.028  
## Chain 1: 9400 -389.180 0.034 0.027  
## Chain 1: 9500 -391.402 0.033 0.027  
## Chain 1: 9600 -386.944 0.033 0.027  
## Chain 1: 9700 -407.125 0.034 0.028  
## Chain 1: 9800 -388.256 0.033 0.028  
## Chain 1: 9900 -385.948 0.026 0.027  
## Chain 1: 10000 -392.737 0.026 0.022  
## Chain 1: 10100 -388.921 0.026 0.022  
## Chain 1: 10200 -395.030 0.024 0.017  
## Chain 1: 10300 -385.888 0.022 0.017  
## Chain 1: 10400 -383.526 0.022 0.017  
## Chain 1: 10500 -385.761 0.020 0.015  
## Chain 1: 10600 -409.672 0.021 0.015  
## Chain 1: 10700 -387.937 0.024 0.017  
## Chain 1: 10800 -385.356 0.023 0.015  
## Chain 1: 10900 -383.993 0.022 0.012  
## Chain 1: 11000 -396.310 0.021 0.012  
## Chain 1: 11100 -402.827 0.020 0.012  
## Chain 1: 11200 -405.168 0.020 0.012  
## Chain 1: 11300 -384.144 0.022 0.015  
## Chain 1: 11400 -389.232 0.022 0.015  
## Chain 1: 11500 -387.886 0.022 0.015  
## Chain 1: 11600 -390.439 0.022 0.015  
## Chain 1: 11700 -385.420 0.020 0.013  
## Chain 1: 11800 -387.076 0.018 0.013  
## Chain 1: 11900 -386.889 0.018 0.013  
## Chain 1: 12000 -385.460 0.017 0.010 MEDIAN ELBO CONVERGED  
## Chain 1:   
## Chain 1: Drawing a sample of size 1000 from the approximate posterior...   
## Chain 1: COMPLETED.

post3 <- as.data.frame(fit3)  
  
ypred <-  
 post3 %>%   
 dplyr::select(contains("output")) %>%  
 apply(., 2, as.integer) %>%  
 apply(., 2, DescTools::Mode)

library(MLmetrics)  
Accuracy(ypred, y\_test)

## [1] 0.875