A) First and foremost, we need to divide our data to test and train datasets

1.

> d<- read.csv("D:/data sets/bank.csv")

d$day=factor(d$day)

2.

s=sample(nrow(d),floor(0.2\*nrow(d)),replace = F)

> dtest=d[s,]

> nrow(dtest)

[1] 2125

> dtrain = d[-s,]

> nrow(dtrain)

[1] 8500

3. **#Remove duration from dtrain:**

dtrain=dtrain[c('age','job','marital','education','default','balance','housing','loan','contact','day','month','campaign','pdays','previous','poutcome','deposit')]

> names(dtrain)

[1] "age" "job" "marital" "education"

[5] "default" "balance" "housing" "loan"

[9] "contact" "day" "month" "campaign"

[13] "pdays" "previous" "poutcome" "deposit"

> str(dtrain)

'data.frame': 8500 obs. of 16 variables:

$ age : int 59 56 41 55 42 60 37 28 38 29 ...

$ job : Factor w/ 12 levels "admin.","blue-collar",..: 1 1 10 8 5 6 10

$ marital : Factor w/ 3 levels "divorced","married",..: 2 2 2 2 3 1 2 3 3 2

$ education: Factor w/ 4 levels "primary","secondary",..: 2 2 2 2 3 2 2 2 2

$ default : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...

$ balance : int 2343 45 1270 2476 0 545 1 5090 100 199 ...

$ housing : Factor w/ 2 levels "no","yes": 2 1 2 2 2 2 2 2 2 2 ...

$ loan : Factor w/ 2 levels "no","yes": 1 1 1 1 2 1 1 1 1 2 ...

$ contact : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3

$ day : Factor w/ 1 level "c(5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 19, 20, 21, 23, 26, 27, 28, 29, 30, 2, 3, 4, 11, 17, 18, 24, 1, 10, 22, 25, 31)": NA NA NA NA NA NA NA NA NA NA ...

$ month : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9

$ campaign : int 1 1 1 1 2 1 1 3 1 4 ...

$ pdays : int -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...

$ previous : int 0 0 0 0 0 0 0 0 0 0 ...

$ poutcome : Factor w/ 3 levels "failure","success",..: 3 3 3 3 3 3 3 3 …

$ deposit : Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...

> c= names (dtrain) [1:length(dtrain)-1]

> k=paste("deposit~", paste(c , collapse="+"))

mf=glm(deposit~., family=binomial, data=dtrain)

stb=step(mf, direction=“backward”)

**The last step is:**

Step: AIC=9580.77

deposit ~ job + marital + education + default + balance + housing + loan + contact + day + month + campaign + poutcome

Df Deviance AIC

<none> 9446.8 9580.8

- default 1 9449.4 9581.4

- education 3 9458.7 9586.7

- loan 1 9460.9 9592.9

- balance 1 9466.3 9598.3

- job 11 9490.4 9602.4

- marital 2 9481.1 9611.1

- housing 1 9484.0 9616.0

- campaign 1 9491.9 9623.9

- day 30 9560.0 9634.0

- contact 2 9622.2 9752.2

- month 11 9824.2 9936.2

- poutcome 2 9837.3 9967.3

Now testing the Regression and stepwise by dtest dataset:

dtest=dtest[c('age','job','marital','education','default','balance','housing','loan','contact','day','month','campaign','pdays','previous','poutcome','deposit')]

You should not fit a model using your test data. The model must be fitted using the training data. We just evaluate the model based on the test data.

mftest=glm(deposit~., family=binomial, data=dtest)

stb=step(mftest, direction=“backward”)

the last step is :

Step: AIC=2501.99

deposit ~ marital + default + housing + loan + contact + day +

month + campaign + poutcome

Df Deviance AIC

<none> 2398.0 2502.0

- campaign 1 2401.1 2503.1

- default 1 2402.1 2504.1

- day 30 2460.3 2504.3

- marital 2 2409.8 2509.8

- housing 1 2413.1 2515.1

- loan 1 2416.6 2518.6

- month 11 2471.5 2553.5

- contact 2 2468.5 2568.5

- poutcome 2 2484.4 2584.4

The results shows, the AIC for the model of dtest is less than the relevant AIC for dtrain ( AIC=2501.99< AIC=9580.77) So the dtest model is more accurate than the dtrain model.

**B)** Interpret your coefficient estimates using odds ratio concepts (it is enough to interpret two of the coefficients).

> stb$coefficients

(Intercept) jobblue-collar jobentrepreneur

9.851422e-01 -7.673708e-03 -2.436367e-01

jobhousemaid jobmanagement jobretired

-3.560020e-01 -1.149598e-01 4.937195e-01

jobself-employed jobservices jobstudent

-3.667316e-02 -2.928435e-02 4.863077e-01

jobtechnician jobunemployed jobunknown

-1.461871e-02 2.605128e-01 -3.239045e-01

maritalmarried maritalsingle educationsecondary

-2.506156e-01 8.730107e-02 1.340446e-01

educationtertiary educationunknown balance

2.832851e-01 1.837390e-01 3.402871e-05

housingyes loanyes contacttelephone

-3.336951e-01 -2.892311e-01 -1.824295e-01

contactunknown day2 day3

-1.168190e+00 -4.784350e-01 -1.856190e-01

day4 day5 day6

-1.573282e-01 -4.726755e-01 -4.675503e-01

day7 day8 day9

-6.398192e-01 -4.398973e-01 -3.389670e-01

day10 day11 day12

3.979081e-01 -4.187264e-01 -1.656791e-01

day13 day14 day15

-7.462723e-02 -3.516379e-01 -2.127094e-01

day16 day17 day18

-1.828047e-01 -6.695066e-01 -4.731767e-01

day19 day20 day21

-9.229931e-01 -6.697650e-01 -2.822913e-01

day22 day23 day24

-4.186260e-01 -3.184735e-02 -3.297364e-01

day25 day26 day27

9.197736e-02 -2.939126e-01 -1.533568e-01

day28 day29 day30

-6.312007e-01 -9.096714e-01 -7.246451e-02

day31 monthaug monthdec

-7.177625e-01 -7.562913e-01 1.793719e+00

monthfeb monthjan monthjul

-3.537627e-01 -8.877877e-01 -6.608528e-01

monthjun monthmar monthmay

8.763286e-02 1.521312e+00 -5.393309e-01

monthnov monthoct monthsep

-7.348004e-01 8.323284e-01 7.181343e-01

campaign poutcomesuccess poutcomeunknown

-7.995848e-02 2.030027e+00 1.798828e-02

Descriptions:

> levels(d$job)

"admin." "blue-collar" "entrepreneur" "housemaid" "management" "retired" "self-employed" "services" "student" "technician" "unemployed" "unknown"

The admin has been choosen the base for comparison the categories

- jobmanagement Beta^ is -1.149598e-01 so In **comparison** with jobadmin indicate that probability of management to making deposit is less than admins .

The interpretation is not correct.

> exp(-1.541026e-01)

[1] 0.8571841

exp(4.345344e-01)

[1] 1.544244

- the retired category’s Beta^ is 4.937195e-01 and it mean this category probability to deposit is more than admins.

**C)** obtain the estimated probabilities of making a deposit using the “predict” function

phat=predict(stb,dtest,type=’response’)

> head(phat,20)

2383 7324 6458 637 4996 10046

0.3107293 0.3500465 0.1503801 0.4779324 0.4614548 0.5325505

6417 3747 8665 1341 5009 6124

0.5899036 0.8047150 0.2712726 0.8996491 0.4202866 0.2628909

3592 8023 1754 2496 9021 5119

0.3259391 0.3280713 0.3632741 0.8820852 0.9047305 0.5766911

1433 3993

0.7766631 0.8469172

**D) Use phat to obtain the class for each of the observations in dtest data and call the estimated classes yhat:**

> yhat<- ifelse(phat<0.5 , "No" , "Yes")

> which(yhat[]=="No")

2383 7324 6458 637 4996 8665 5009 6124 3592 8023

1 2 3 4 5 9 11 12 13 14

1754 9335 7340 1858 9860 735 6586 8615 9859 4697

15 21 23 24 25 26 27 29 31 32

**…**

[ reached getOption("max.print") -- omitted 373 entries ]

> which(yhat[]=="Yes")

10046 6417 3747 1341 2496 9021 5119 1433 3993 5572

6 7 8 10 16 17 18 19 20 22

1315 8943 7809 1561 1668 6024 3600 7123 4715 6719

28 30 33 34 35 37 41 42 46 48

1699 10542 4442 3019 2552 8577 2609 3822 1498 1736

**E) The final step is to compute the testing prediction accuracy so to find out in how many cases the yhat is equal to dtest$deposit :**

> dtest$deposit

[1] yes no no yes no no no yes no yes no no yes no … 1125 entries.

> summary(yhat)

Length Class Mode

2125 character character

> str(yhat)

Named chr [1:2125] "No" "No" "No" "No" "No" "Yes" "Yes" ...

> str(dtest$deposit)

Factor w/ 2 levels "no","yes": 2 1 1 2 1 1 1 2 1 2 ..

> levels(dtest$deposit)

[1] "no" "yes"

**So at first should change the type of the yhat to factor :**

> yhat=as.factor(yhat)

> str(yhat)

Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 2 2 1 2 ...

- attr(\*, "names")= chr [1:2125] "2383" "7324" "6458" "637" ...

> levels(yhat)

[1] "No" "Yes"

> levels(yhat)=c("no","yes")

> levels(yhat)

[1] "no" "yes"

**Now we can compare the yhat and dtest$deposit :**

> same=yhat[]==dtest$deposit

> same

[1] FALSE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE

[10] TRUE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE

**…**

[ reached getOption("max.print") -- omitted 1125 entries ]

> which(same=="TRUE")

[1] 2 3 5 8 9 10 11 12 14 16 19

[12] 20 21 23 25 27 28 29 31 34 35 38

**…**

[ reached getOption("max.print") -- omitted 470 entries ]

**Best regards.**