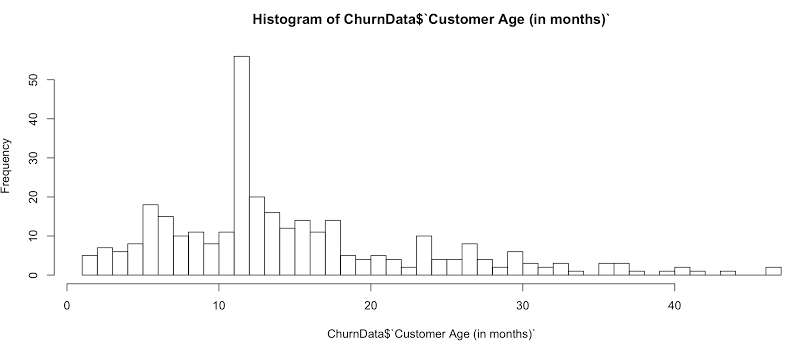
Problem 1.

1. Is Wall’s belief about the dependence of churn rates on customer age supported by the data? To get some intuition, try visualizing this dependence (Hint: You can answer this question based on graphs. No need to run any statistical tests).

Based on the graph below, we saw that the churn rates are highest when the customer age is at 12. Going back to Wall’s belief about dependence of churn rates is supported by the data, because again at age 12 churn frequency is the highest.



1. Construct the best model to predict customers churn. You can try different classification models such as Logistic Regression, Neural Network and Decision Trees. You need to find the best settings for these models. Explain how you evaluate your models i.e. what is your evaluation measure? Accuracy? Precision? Recall? etc.

C:\Users\Ravi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\PartB.JPGAccuracy: (1777 + 4)/ (1777 + 94 + 4) = 94.99%

Recall: 1777/1777 + 94 = 94.98%

Precision 1777/ (1777+4) = 99.77%

Overall the logistic regression is better because the accuracy of the model is better comparatively to the neural network dataset.

Code for question above:

#logistic regression

View(Churn\_Data)

Churn\_Data = MyData

set.seed(1234)

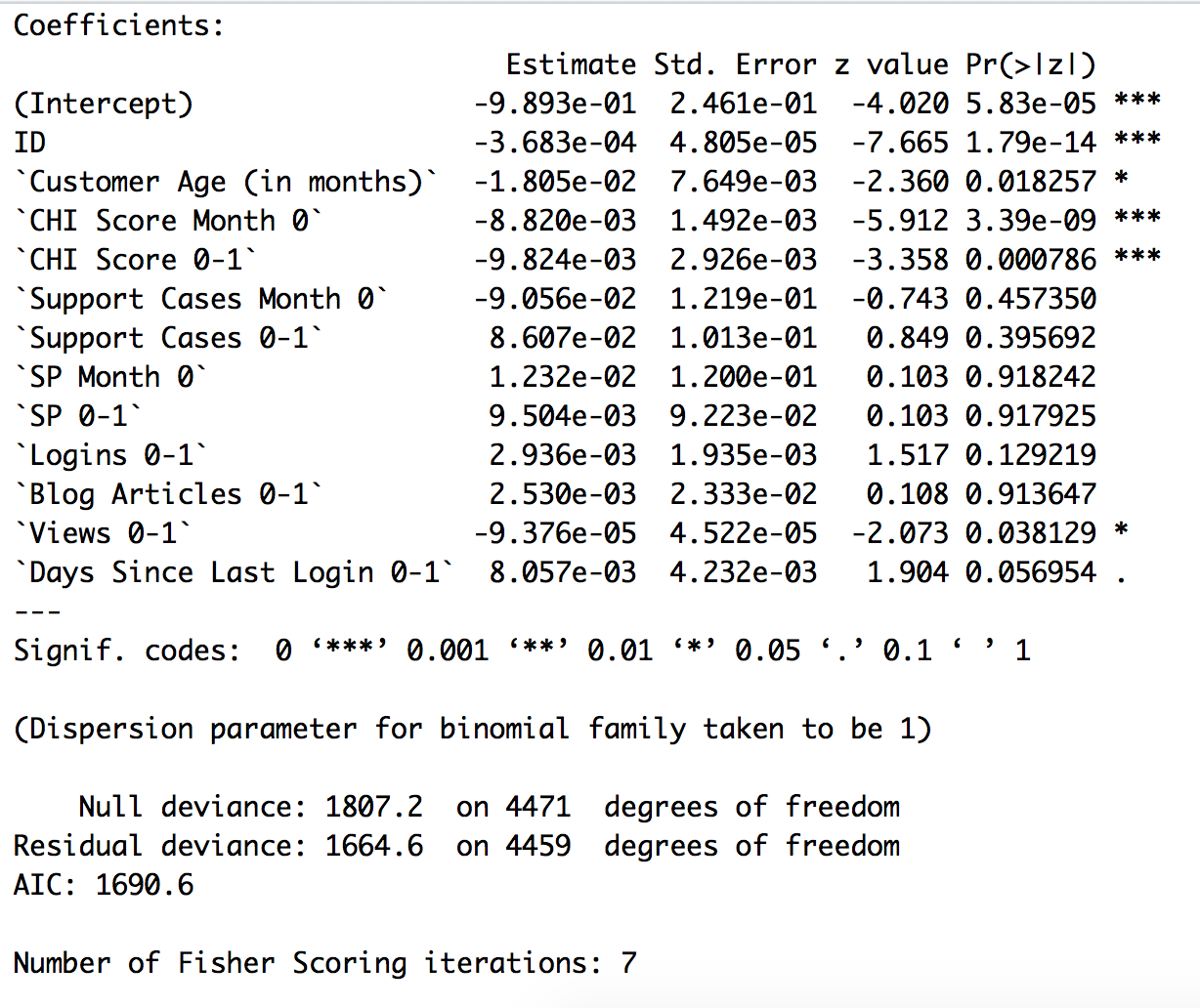
logis = sample(2, nrow(MyData), replace = T, prob = c(0.7,0.3))

TrainData = MyData [logis == 1,]

TestData = MyData [logis == 2,]

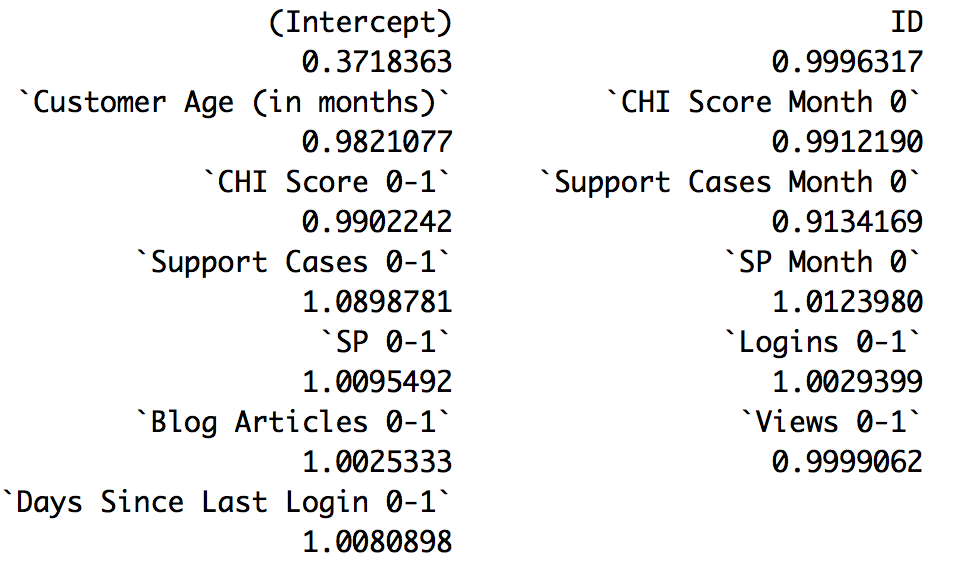
LMData = glm (formula = Churn~., family = “binomial”, data = TrainData)

Summary(LMData)



Based on the code we run using glm, through training data we have A|C = 1690.6, and there are 4 sig variables that are used to test the data.

exp(cbind(Or = coef(LMData), confint(LMData)))



The output of the exponential coefficient is used to get the odd ration and predict like-hood of customer churn rate

With(LMData, null.deviance = deviance)

[1] 142.5723

With(LMData, df.null, df.residual)

[1] 4471

With(LMData, pchisq(null.deviance – deviance, df.null – df.residual, lower.tail = FALSE))

[1] 1.81039e-24

Since our chi-square is 142.57 for the data with a degree of freedom and associated p-value less than .0001 our model fits significantly.

#neural network

library(ISLR)

library(nnet)

library(devtools)

source\_url('<https://gist.githubusercontent.com/fawda123/7471137/raw/466c1474d0a505ff044412703516c34f1a4684a5/nnet_plot_update.r>')

MyData$ID = NULL

MyData$Churn = factor(MyData$Churn)

maxs = apply(q[,3:12], 2, max)

mins = apply(q[,3:12], 2, min)

scaled.data = as.data.frame(scale(MyData[,3:12], center = mins, scale = maxs - mins))

set.seed(1234)

neural = sample(2, nrow(MyData), replace = T, prob = c(0.7,0.3))

trainData = MyData[neural == 1,]

testData = MyData[neural == 2,]

str(MyData)

nn  = nnet(Churn ~ ., data = trainData, linout=F, size=10, decay=0.01, maxit=1000)

plot.nnet(nn)

nn.preds = predict(nn, testData, type = "class")

table(testData$Churn, nn.preds)

lift = performance[pred, “lift”, “rpp”]

plot(perf, main = “lift curve”)

1. Answer Well’s “ultimate question”: provide the list of 100 customers with highest churn probabilities and the top three drivers of churn for each customer (You do not need to include a list of 100 customers. Just mention how you get this list – you can write the code for example).

customerNumber = 100

topCustomer = function(data = q, n = numberConstant)

{

nr = (1:nrow(data))[order(predict(nn), decreasing = T)[1:n]]

return(data[nr,])

}