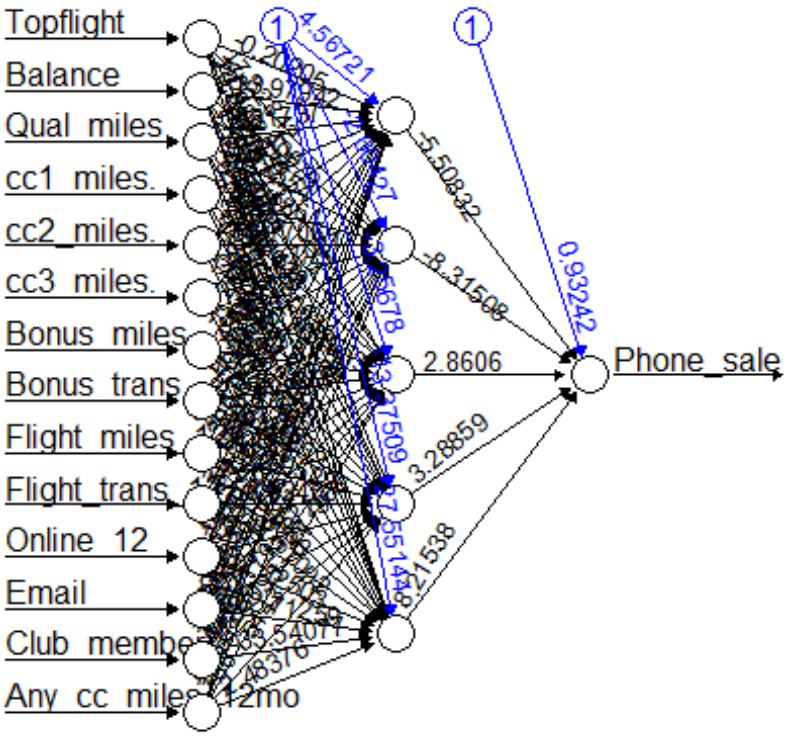
Individual Assignment 10

Charlie Ling

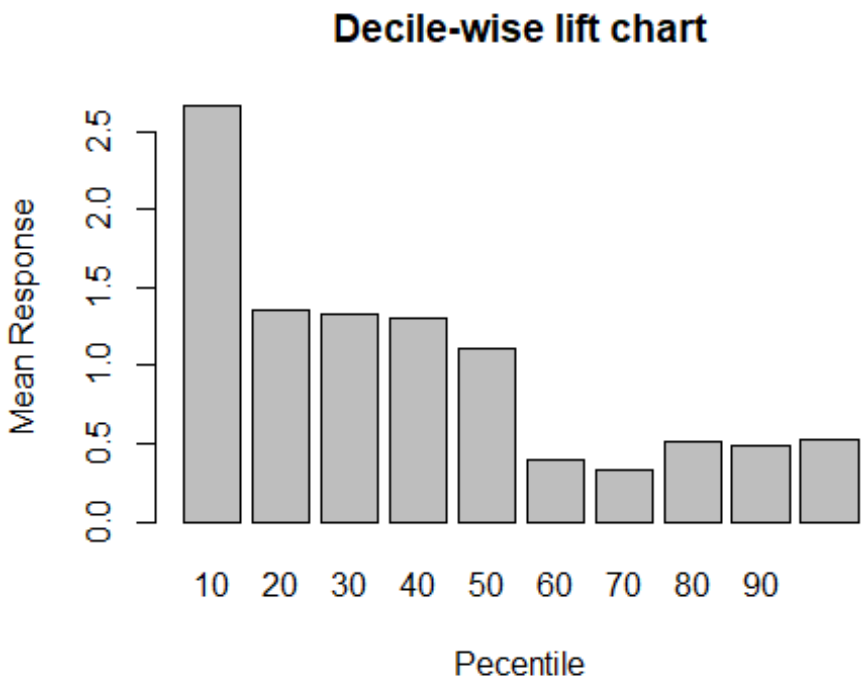
11/21/2021

1.Run a neural net model on these data, using a single hidden layer with 5 nodes. Remember to first convert categorical variables into dummies and scale numerical predictor variables to a 0-1 (use function preprocess() with method="range" - see Chapter 7). Generate a deciles-wise lift chart for the training and validation sets. Interpret the meaning (in business terms) of the leftmost bar of the validation decile- wise lift chart.

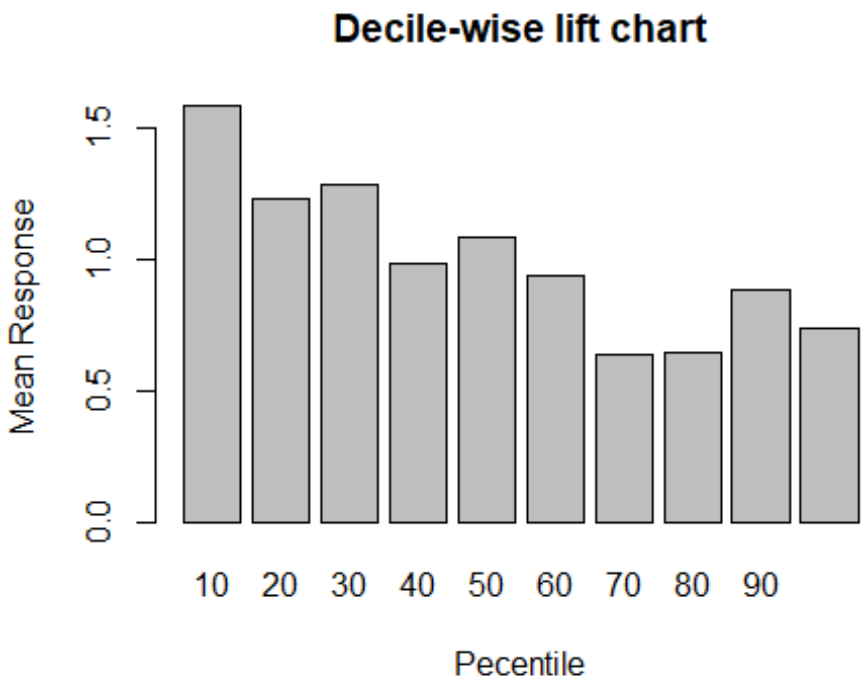
```
setwd("C:/LearnAndWorkdDocuments/500s")
df=read.csv("EastWestAirlinesNN.csv")
df = na.omit(df)
df = subset(df, select = -c(1))

phone_sale = df[,"Phone_sale"]
max_sale = range(df["Phone_sale"])[2]
min_sale = range(df["Phone_sale"])[1]

library(caret)

## 载入需要的程辑包：ggplot2

## 载入需要的程辑包：lattice

numerical = c("Balance", "Bonus_miles", "Bonus_trans",
"Flight_miles_12mo","Flight_trans_12")
norm.values = preProcess(df[, numerical], method = "range")
df[, numerical] = predict(norm.values, df[, numerical])

set.seed(100)
train = sample(1:nrow(df), nrow(df)*0.7)
df.train = df[train,]
df.test = df[-train,]

library(neuralnet)
nn = neuralnet(Phone_sale ~., data = df.train, linear.output = F, hidden = 5)
plot(nn, rep = "best")
```




```
nn.pred.train = predict(nn, df.train, type = "class")
nn.pred.test = predict(nn, df.test, type = "class")


library(gains)

gain.train = gains(df.train$Phone_sale, nn.pred.train)
barplot(gain.train$mean.resp/mean(df.train$Phone_sale), names.arg =
gain.train$depth, xlab = "Pecentile", ylab = "Mean Response", main = "Decile-
wise lift chart")
```




```
gain.test = gains(df.test$Phone_sale, nn.pred.test)
barplot(gain.test$mean.resp/mean(df.test$Phone_sale), names.arg =
gain.test$depth, xlab = "Pecentile", ylab = "Mean Response", main = "Decile-
wise lift chart")
```



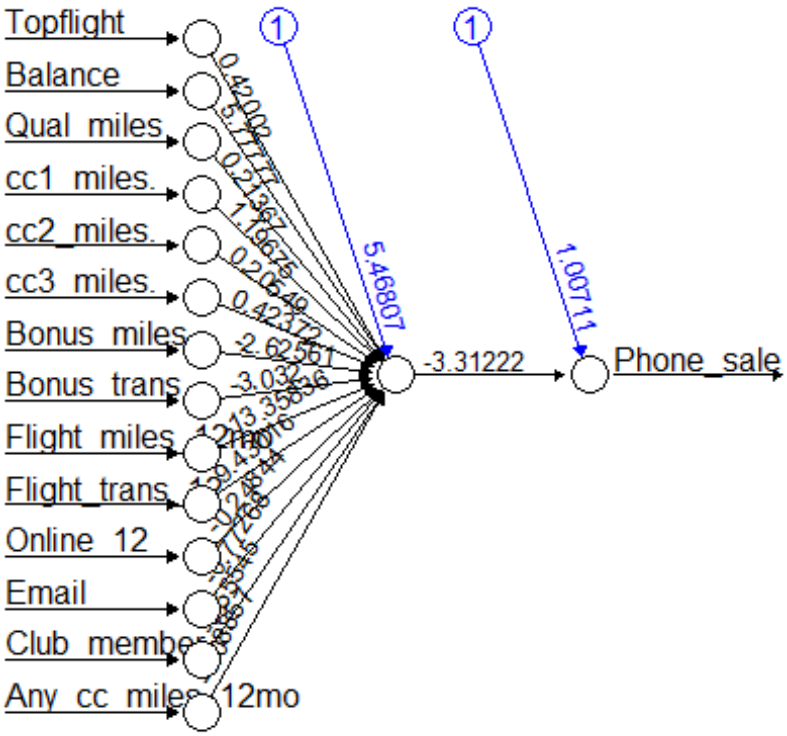

```
# If we choose 10% of the customers, we would gain about 1.3 times the
probability of successful phone sale.
```

2.Comment on the difference between the training and validation lift charts.

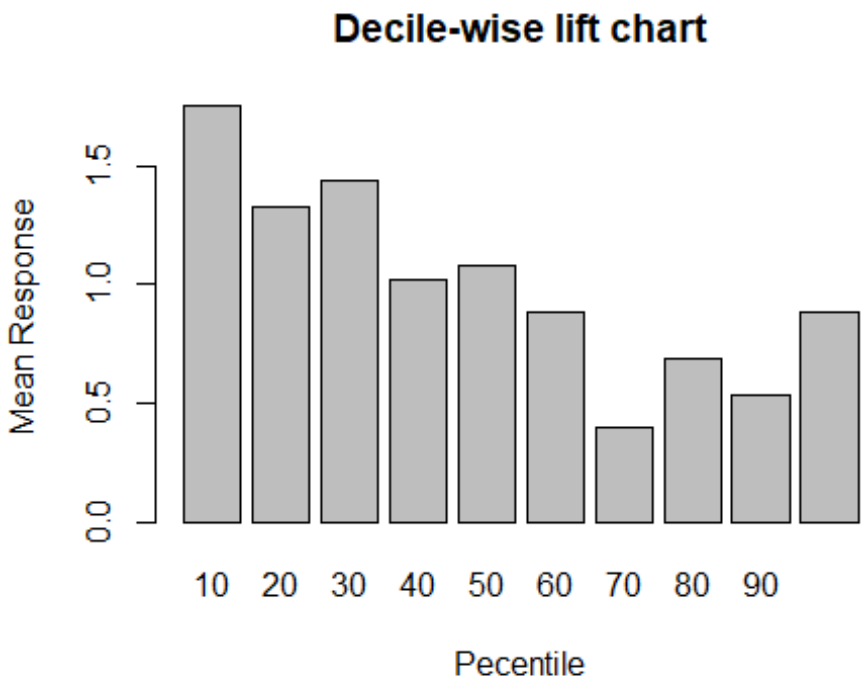
```
# The result in training set is better than validation set, so there is over-
fitting problem
```

3.Run a second neural net model on the data, this time setting the number of hidden nodes to 1. Comment now on the difference between this model and the model you ran earlier, and how overfitting might have affected results.

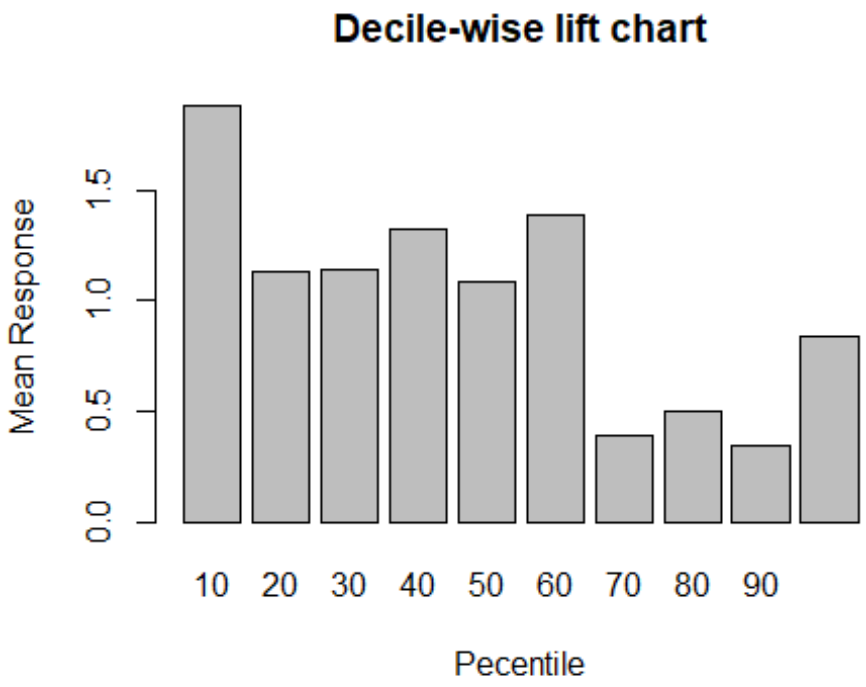
```
nn.new = neuralnet(Phone_sale ~., data = df.train, linear.output = F, hidden
= 1)
plot(nn.new, rep = "best")
```




```
nn.pred.train.new = predict(nn.new, df.train, type = "class")
nn.pred.test.new = predict(nn.new, df.test, type = "class")


gain.train.new = gains(df.train$Phone_sale, nn.pred.train.new)
barplot(gain.train.new$mean.resp/mean(df.train$Phone_sale), names.arg =
gain.train.new$depth, xlab = "Pecentile", ylab = "Mean Response", main =
"Decile-wise lift chart")
```




```
gain.test.new = gains(df.test$Phone_sale, nn.pred.test.new)
barplot(gain.test.new$mean.resp/mean(df.test$Phone_sale), names.arg =
gain.test.new$depth, xlab = "Pecentile", ylab = "Mean Response", main =
"Decile-wise lift chart")
```




```
# The model with more hidden nodes would have a lower error rate for both
training set and testing set. Overfitting makes testing error is greater than
training error
```

4.What sort of information, if any, is provided about the effects of the various variables?

```
# By observing the weights, we can see balance and flight miles 12 mo are
most effective variables
```