# Intuit Online

## Intuit Online Case

This is a group assignment for MKTG482. Group member: Javier Urbina, Arsan Kamran, Toshi Murakami

### Load data and library

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(mktg482)
library(nnet)

rm(list=ls())
set.seed(13579)
load("intuit_online.Rdata")
```

### Split Data into test and training

```
intuit_training <- intuit %>% filter(training==1)
intuit_test<-intuit %>% filter(training==0)
```

### Neural Network

```
intuit_nn5 <- nnet(res~speeddown+speedup+last+numords+dollars+sincepurch+version2013+upgraded+payroll+bi
```

```
## # weights:  81
## initial  value 44603.969807
## iter  10 value 5538.980205
## iter  20 value 5529.955773
## iter  30 value 5483.844558
## iter  40 value 5452.140290
## iter  50 value 5382.124027
## iter  60 value 5167.464103
## iter  70 value 5073.989475
## iter  80 value 5029.526283
## iter  90 value 4961.548060
## iter 100 value 4943.058843
## iter 110 value 4890.244429
## iter 120 value 4847.632904
## iter 130 value 4830.568462
## iter 140 value 4816.643627
## iter 150 value 4808.486414
## iter 160 value 4796.099811
## iter 170 value 4766.905112
## iter 180 value 4744.312183
## iter 190 value 4725.013492
## iter 200 value 4719.309384
## iter 210 value 4715.938412
## iter 220 value 4712.760522
## iter 230 value 4709.183992
## iter 240 value 4707.124128
## iter 250 value 4706.396575
## iter 260 value 4706.055268
## iter 270 value 4705.842572
## iter 280 value 4705.762572
## iter 290 value 4705.751774
## final  value 4705.751253
## converged
```

```
intuit_nn10 <- nnet(res~speeddown+speedup+last+numords+dollars+sincepurch+version2013+upgraded+payroll+
```

```
## # weights:  161
## initial  value 55859.328260
## iter  10 value 5560.668759
## iter  20 value 5471.323714
## iter  30 value 5413.854631
## iter  40 value 5380.432117
## iter  50 value 5342.916233
## iter  60 value 5306.519359
## iter  70 value 5033.807150
## iter  80 value 4965.182602
## iter  90 value 4937.494165
## iter 100 value 4922.657541
## iter 110 value 4910.172058
## iter 120 value 4900.197580
## iter 130 value 4895.886934
## iter 140 value 4890.393869
## iter 150 value 4869.069318
## iter 160 value 4840.955204
## iter 170 value 4819.725977
## iter 180 value 4789.643816
## iter 190 value 4769.731716
## iter 200 value 4752.547269
## iter 210 value 4738.219298
## iter 220 value 4721.109711
## iter 230 value 4707.592327
## iter 240 value 4688.700666
## iter 250 value 4667.673893
## iter 260 value 4646.663766
## iter 270 value 4621.480457
## iter 280 value 4591.088762
## iter 290 value 4575.687602
## iter 300 value 4572.192427
## iter 310 value 4571.288013
## iter 320 value 4570.563776
## iter 330 value 4567.522820
## iter 340 value 4567.029928
## iter 350 value 4566.919904
## iter 360 value 4566.811612
## iter 370 value 4566.130100
## iter 380 value 4566.086239
## iter 390 value 4566.004884
## iter 400 value 4565.903935
## iter 410 value 4565.588660
## iter 420 value 4565.256582
## iter 430 value 4564.422361
## iter 440 value 4563.625734
## iter 450 value 4562.900591
## iter 460 value 4562.285523
## iter 470 value 4561.910688
## iter 480 value 4561.868884
## final  value 4561.868503
## converged
```

```
pred_nn5_training<- predict(intuit_nn5,type="raw")
pred_nn5_test<- predict(intuit_nn5,newdata = intuit_test,type="raw")


pred_nn10_training<- predict(intuit_nn10,type="raw")
pred_nn10_test<- predict(intuit_nn10,newdata = intuit_test,type="raw")
```

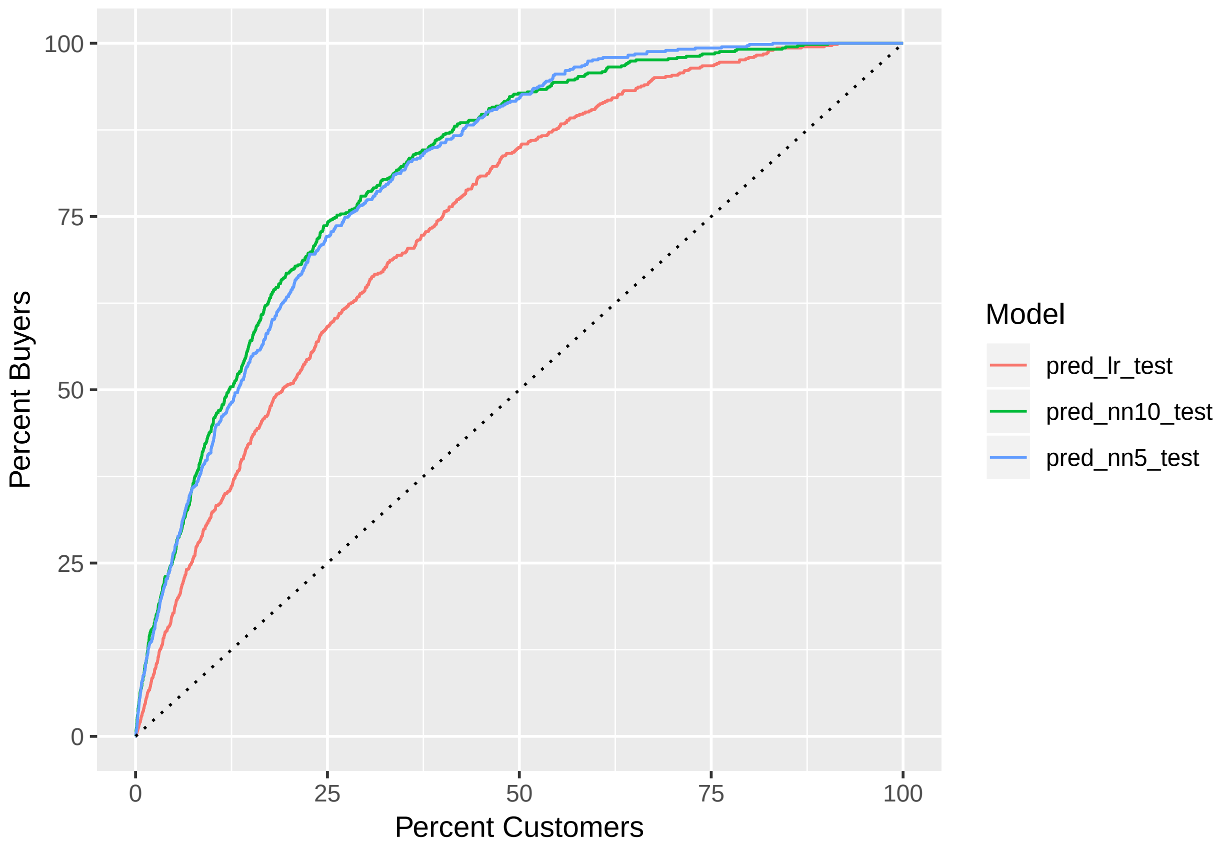
## Logistic Regression

```
intuit_lr<-glm(res~speeddown+speedup+last+numords+dollars+sincepurch+version2013+upgraded+payroll+bizfla

pred_lr_training<-predict(intuit_lr, type="response")
pred_lr_test<-predict(intuit_lr,newdata=intuit_test ,type="response")
```

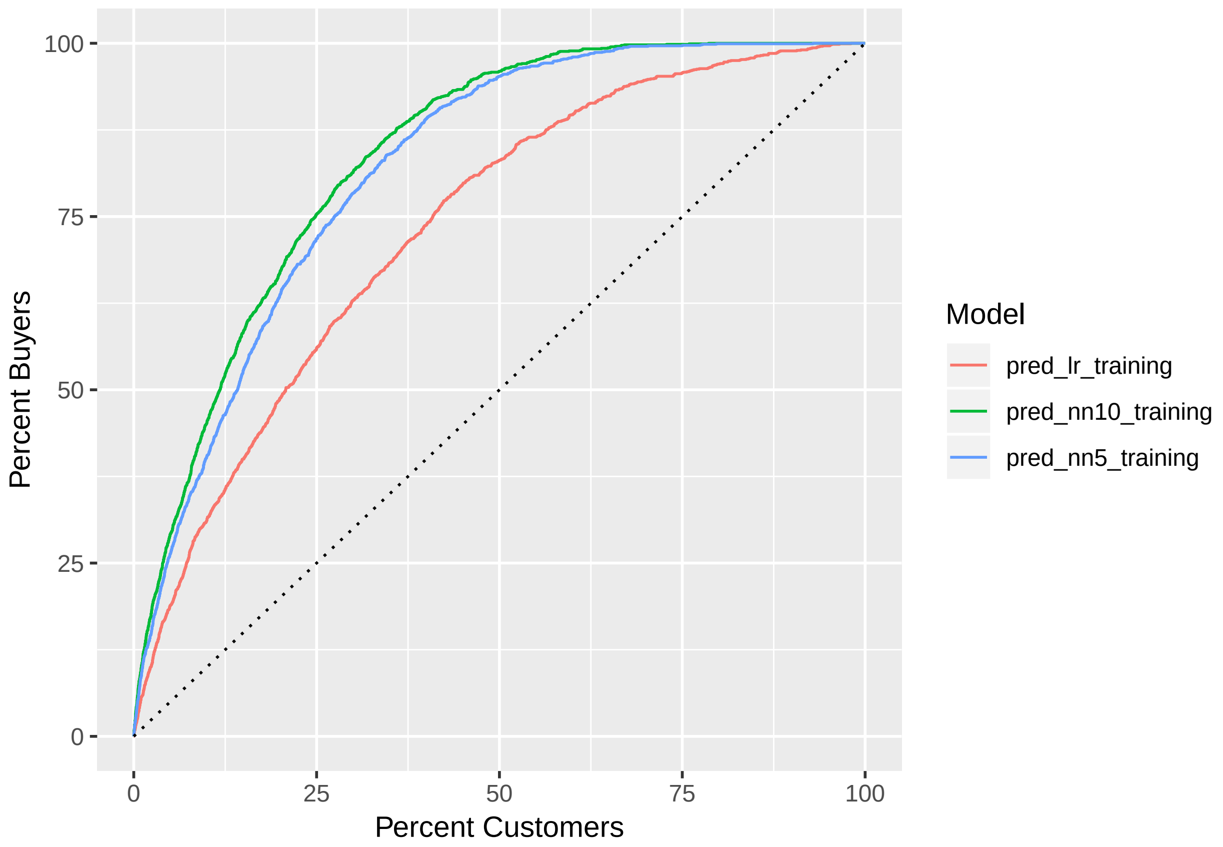
## Model Performance comparison

```
gainsplot(pred_nn5_test,pred_nn10_test,pred_lr_test, label.var =intuit_test$res )
```



```
## # A tibble: 3 x 2
##   model             auc
##   <chr>           <dbl>
## 1 pred_nn5_test  0.827
## 2 pred_nn10_test 0.83
## 3 pred_lr_test   0.755
```

```
gainsplot(pred_nn5_training,pred_nn10_training,pred_lr_training, label.var =intuit_training$res )
```



```
## # A tibble: 3 x 2
##   model                auc
##   <chr>              <dbl>
## 1 pred_nn5_training  0.833
## 2 pred_nn10_training 0.85
## 3 pred_lr_training   0.744
```

In each NN prediction works better than logistic regression, thus we decided to use NN model. In NN models, 10 layer has more overfitting problem than 5 layer model. Thus we decided to use 5 layer model.

Assign NN result to Wave2 data.

```
pred_nn5_wave2<- intuit.wave2 %>%
  mutate(pred_wave2=predict(intuit_nn5,newdata = intuit.wave2,type="raw"))
```

### Calculate break even rate

Cost/target : \$1.60 Target Profit : \$5.6 Break Even Response rate is described as 7.2/180 = 4%

```
BER=7.2/180
BER
```

```
## [1] 0.04
```

## Set target customers

For the customers in the dataset, create a new variable (call it ?gtarget?h) with a value of 1 if the customer?fs predicted probability is greater than or equal to the breakeven response rate(4%) and 0 otherwise.

```
options(encoding = "utf-8")

intuit.wave2<-pred_nn5_wave2 %>%
  mutate(target=1*(pred_wave2>= BER))
```

```
intuit.wave2 %>%
  group_by(target) %>%
  summarise(num_target=n(), perc_target=num_target/25000, response_rate=mean(pred_wave2))
```

```
## # A tibble: 2 x 4
##   target num_target perc_target response_rate
##    <dbl>      <int>       <dbl>         <dbl>
## 1      0      16496       0.660        0.0104
## 2      1       8504       0.340        0.0931
```

Of all the customer base, 32.7% of the customer are the target and response rate is 9.6%

## Profitability analysis for Logistic Regression

```
intuit_wave2_res<- intuit.wave2%>%
  filter(target==1) %>%
  summarise(frac_mailed=n()/25000, resp_rate=mean(pred_wave2)) %>%
  mutate(num_mailed=25000*frac_mailed,
         num_responded=resp_rate*num_mailed,
         profit=180*num_responded-1.6*num_mailed,
         ave_profit=profit/num_mailed)
intuit_wave2_res
```

```
## # A tibble: 1 x 6
##   frac_mailed resp_rate num_mailed num_responded  profit ave_profit
##         <dbl>     <dbl>      <dbl>         <dbl>   <dbl>      <dbl>
## 1       0.340    0.0931       8504          792. 128913.       15.2
```

So basically if we decide to send a email to all the customers who exceeds the break even rate of 4%, we get 9% response rate, and profit of $129,330(average profit of $15.7). We can lower the threashold(now it is 4%) and send broader customer base(our objective is to maximize profit not ROI).