Daniel Martin-Williams Coursework 1, penalised logistic regressions and predicting telephone marketing success

2023-10-21

1. Show that the estimated probability $\Pr(y=1)$ for the null model is $\hat{p} = \bar{y}$. Show your derivations. Comment on your result, but be brief. (5 marks)

The null model obtains by restricting $\beta_1 = \beta_2 = \ldots = \beta_p = 0$. Let $p = \Pr(y=1) = \frac{\exp(\beta_0)}{(1+\exp(\beta_0))}$. The likelihood function for the null model therefore reads

$$L(\beta_0) = \prod_{i=1}^{n} \left(\frac{\exp(\beta_0)}{1+\exp(\beta_0)} \right)^{y_i} \left(\frac{1}{1+\exp(\beta_0)} \right)^{1-y_i}.$$

The log-likelihood is

$$\ln L(\beta_0) = \sum_{i=1}^{n} \left\{ y_i \beta_0 - \ln[1+\exp(\beta_0)] \right\},$$

so the maximum likelihood estimator (or minimum deviance estimator) of β_0 solves

$$\sum_{i=1}^{n} \left\{ y_i - \frac{\exp(\hat{\beta}_0)}{1+\exp(\hat{\beta}_0)} \right\} = 0 \iff \frac{\exp(\hat{\beta}_0)}{1+\exp(\hat{\beta}_0)} = \frac{1}{n} \sum_{i=1}^{n} y_i;$$

that is, $\hat{p} = \bar{y}$. The SOC for the maximum likelihood problem is $-\hat{p}(1-\hat{p}) < 0$, which holds.

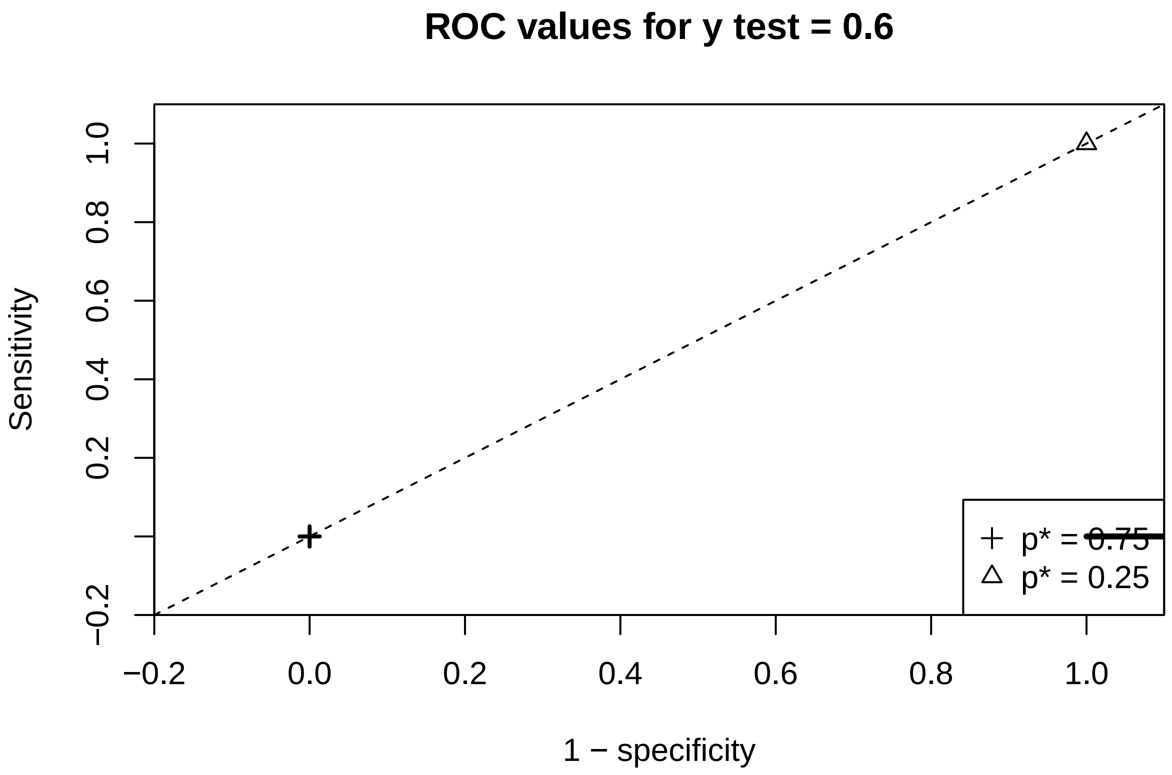
The null model uses the proportion with $y = 1$ in the sample to estimate $p = \Pr(y=1)$. As $\mathbb{E}(y) = p$, the minimum deviance estimator of the null model replaces the unknown population expectation $\mathbb{E}(y)$ with its sample analogue, the known sample average $\bar{y}$. The estimator is unbiased: $\mathbb{E}(\hat{p}) = \mathbb{E}(y) = p$.

Question 2

```r
library(glmnet)
source("naref.R") # input code from naref.R
bank <- read.csv("bank-additional-full.csv", sep = ";", stringsAsFactors = T)
regs <- bank[,1:20] # Select regressors
regs <- subset(regs, select = -c(duration,pdays,nr.employed))
regs$age <- factor(cut(regs$age, breaks = c(25,30,35,40,45,50,55,60,99)))
x <- model.matrix(~ (age + job + marital + education + default + housing + loan)^2 +
                    contact + month + day_of_week + campaign + previous + poutcome +
                    emp.var.rate + cons.price.idx + cons.conf.idx + euribor3m,
                    data = naref(regs))
y <- bank[,"y"] == "yes"

obs <- nrow(x)
regs <- ncol(x)
```

OOS sensitivity for {y}_{train} = 0.62. for the classification rule p* = 0.25, sensitivity and false positive rate (1-sensitivity) will be 1, as the average is above the cutoff, and all points will be given a value of 1. For the p* = 0.75, the opposite will be true and sensitivity and FPR will be 0 , as the value is above the cutoff and so will be given a value of 0.

```r
plot(c(0,0),
     type = "s", lty = 1, lwd = 3,
     xlim = c(-0.2,1.1), ylim = c(-0.2,1.1), xaxs = "i", yaxs = "i",
     xlab = "1 - specificity", ylab =  "Sensitivity", main = "ROC values for y test = 0.6")

points(0,0, pch = 3, lwd = 2)
points(1,1, pch = 2)
abline(0, 1, lty = 2)
legend("bottomright",
       legend=c("p* = 0.75","p* = 0.25"),
       pch = c(3,2))
```



Question 3

How many observations are there in x? How many input variables are there in x? (2 marks)

There are 41188 observations and 751 regressors.

Question 4

A member of your team has heard about a classification method called K-Nearest Neighbour. They find it compelling and suggests to use it. Explain why K-Nearest Neighbour is not a suitable tool forthe telemarketing classification problem? (10 marks)

In General, the K nearest number classification method has its flaws, it tends to be unstable, changing its optimal K when cross-validating as well as giving crude estimations of classification probabilities. For one, this is because it often gives probabilities of 0 or 1. When K is small, if the nearest k points are all from group x, there will be a probability of 1 assigned to x, and 0 to all other possibilities. This is simplistic and less useful than other estimators which give us more accurate probabilities. It is also very sensitive to different values of K, and can change its estimations drastically for different values. Furthermore, by only looking at data immediately clustering around data points, K-NN can be influenced heavily by outliers or outlying clusters of data. For this dataset specifically,K-NN will be impractical due to how large it is, and how computationally taxing the method is. K-NN is computationally expensive as it requires the calculation of its nearest neighbours.

Question 5

Split the data 50/50 in training and test data. Use a table with number of observations and proportions of positives (i.e. y == TRUE) to verify that training and test data are similar (3 marks)

```r
library(knitr)
set.seed(1)

split <- sample((nrow(x)), ceiling(nrow(x)/2))
test <- x[split,]
sample <- x[-split,]
rows <- data.frame(
  Description = c('sample rows', 'test rows', 'sample proportion y == True',
                  'test proportion y == True'),
  Value = c(nrow(sample), nrow(test), sum(y[-split] == T) / nrow(sample),
            sum(y[split] == T) / nrow(test))
)
rows$Value <- format(rows$Value, scientific = FALSE)
kable(rows, caption = "Number of rows and respective proportion of y = true",
      align = c('l', 'r'))
```

Table 1: Number of rows and respective proportion of y = true

Description	Value
sample rows	20594.0000000
test rows	20594.0000000
sample proportion y == True	0.1123628
test proportion y == True	0.1129455

The values are very similar, the data has been split fairly

Question 6

Fit a logistic lasso regression to the training data with cross validation of the penalty weight lambda. Explainwhy a penalized regression approach is appropriate. (10 marks)

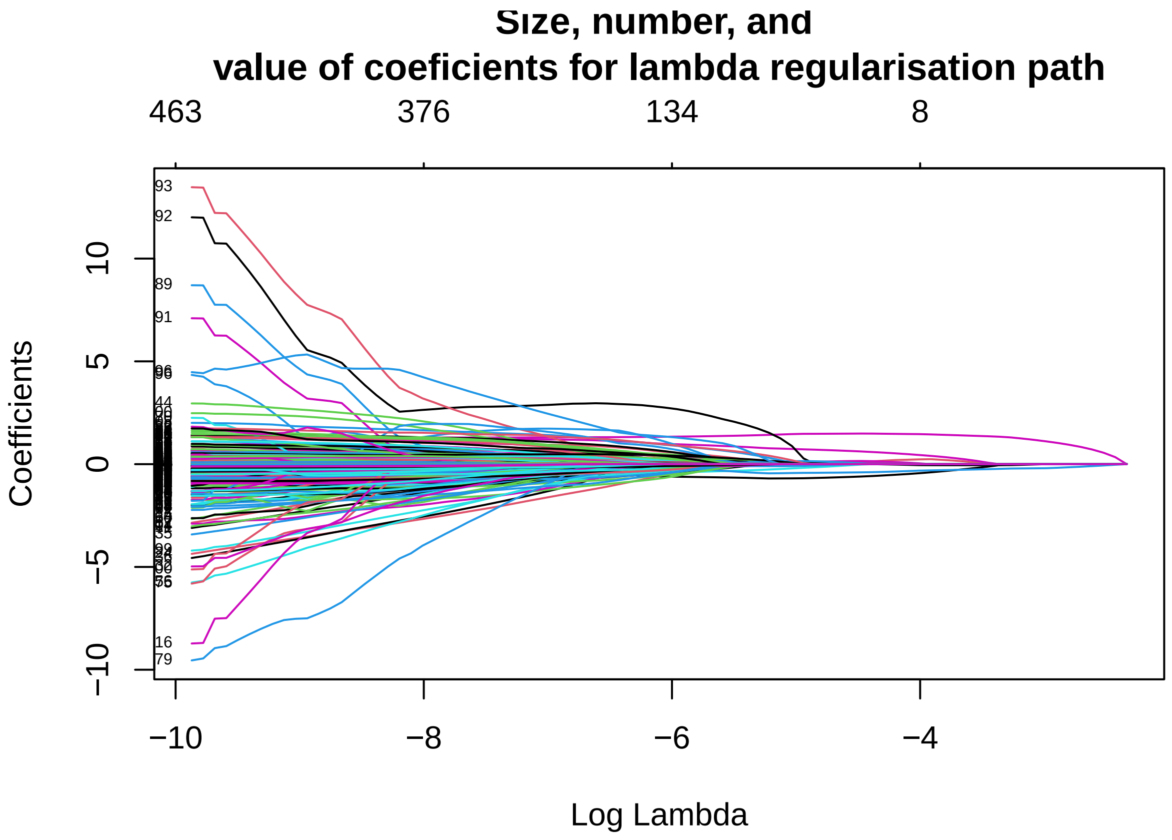
```r
lasso <- cv.glmnet(sample, y[-split], family = "binomial", alpha = 1,
                   )
```

A penalized regression here is appropriate as it is less computationally intensive than a K-NN regression, and given what a large dataset we have, thats significant. The penalized logistic approach as the logistic regression gives us the probability that each term is a 0 or 1, which is what we want to find here as y is boolean. The use of a penalty function is also significant here, as it stops us from overfitting, however it should be noted that there is a bias - vairance tradeoff involved in this. Furthermore, using lasso rather than ridge lets us simplify the model, as it sets many of the regressors to 0, compared to ridge which sets the regressors to be very small, leading to a less reduced model.

Question 7

Plot the coefficients along the lasso regularization path, i.e. against log lambda. What do you observe? Doesthe pattern make sense?Briefly substantiate your answer. (5 marks)

```r
plot(lasso$glmnet.fit, xvar = "lambda", label = T, main = "Size, number, and
value of coeficients for lambda regularisation path

  ")
```

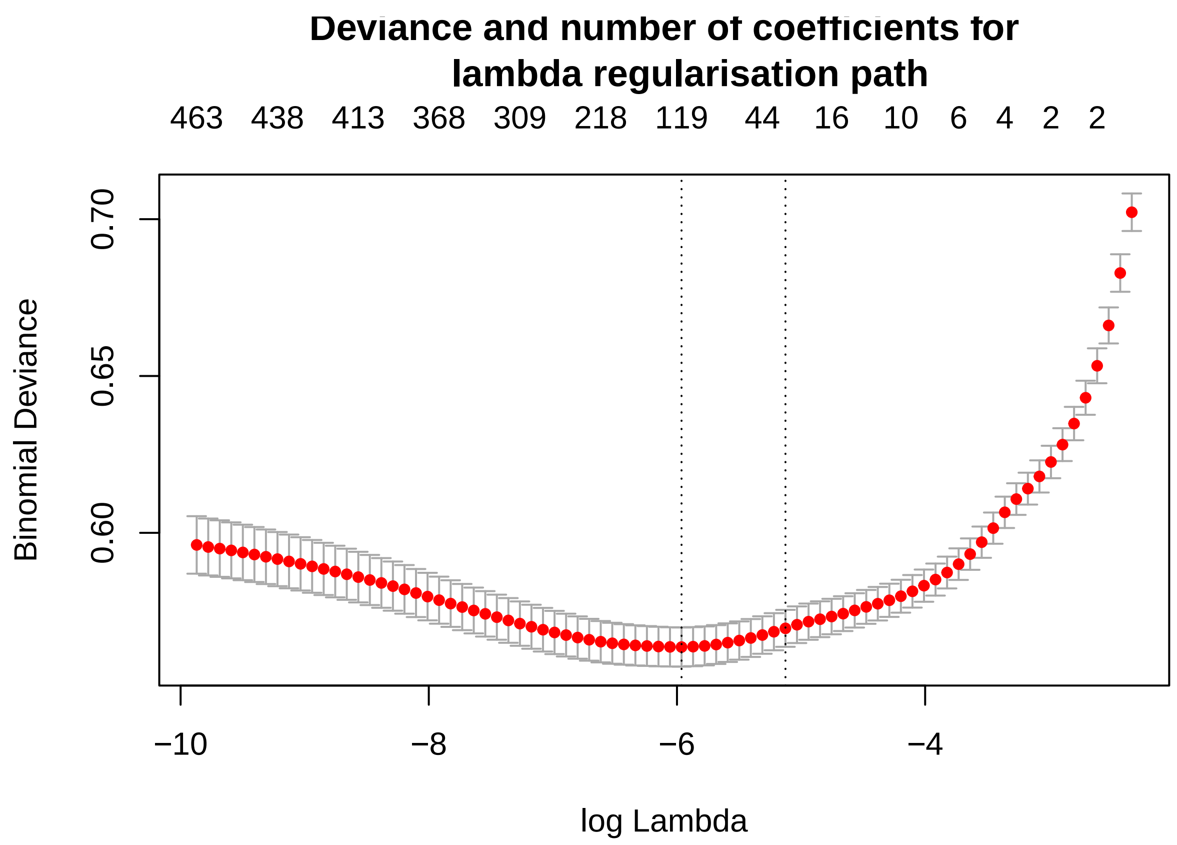


The pattern that is observed, is that as the lambda increases, coefficient values tend towards 0. This makes sense, as the lambda is the penalty weight on the size of the regressors, so the higher the penalty weight the higher the incentive to set a regressor to 0. The number of regressors is also decreasing in lambda for the same reason, less coefficient 'budget' leads to fewer regressors. Some regressors increased in penalty weight, this is suprising at first, but just reflects their significance. The higher the lambda, and the lower the 'budget' for regressors, the more 'budget' wants to be spent on that specific regressor. All regressors end up at 0. This also makes sense, as cv.glmnet calucluates optimal lambda by setting the highest value of lambda to the one where all regressors are set to 0, and then working backwards from there to a specified minimum lambda.

Question 8

Plot the deviance along the lasso regularization path, i.e. against log lambda. What do you observe? Doesthe pattern make sense? Briefly substantiate your answer. (5 marks)

```r
plot(lasso, xlab = "log Lambda", main = "Deviance and number of coefficients for
     lambda regularisation path

     ")
```

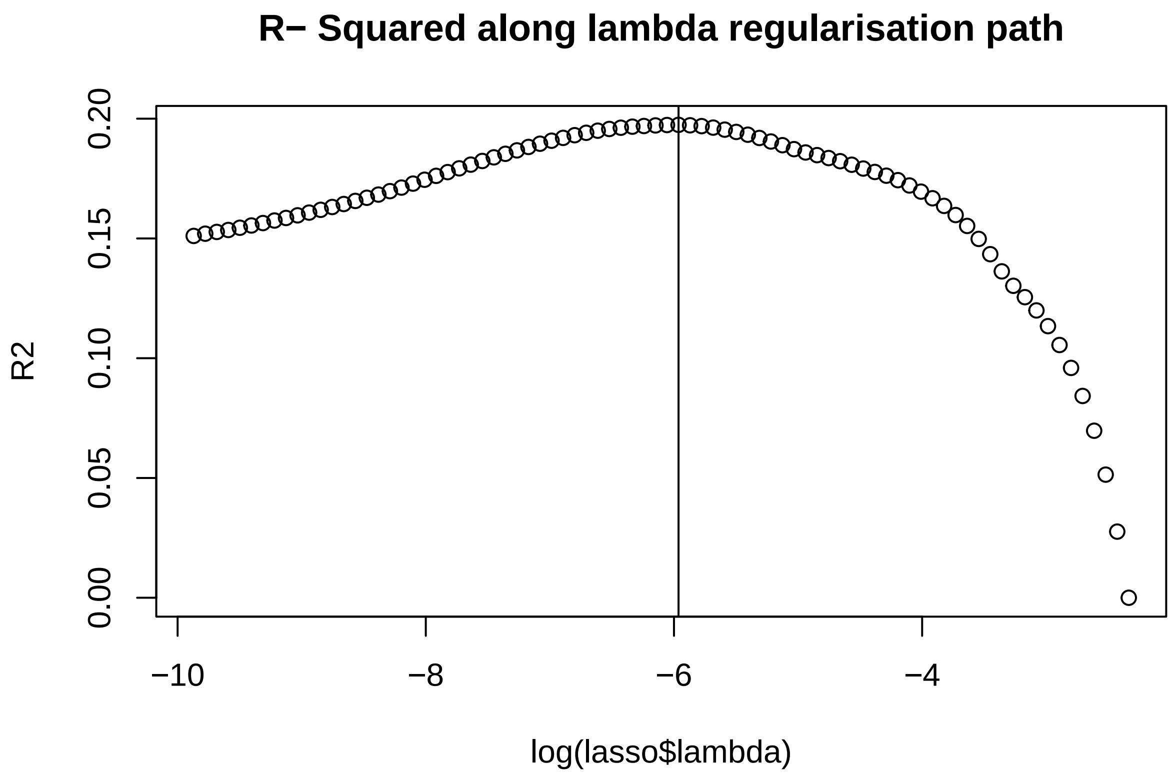


When plotting deviance, we find that it follows an initially decreasing, and then increasing patern with the minimum deviance at 0.002096444, and within 1 standard deviation up to a lambda of 0.00556834. This pattern makes sense, we expect deviance to increase pas a certain point as lambda decreases, as a lower lambda = a lower pentalty weight, and the closer the model becomes to the full unpenalised model. As a result we expect this model to be overfitted, and with a high OOS deviance as a result. We expect the deviance to be decreasing initially, as the penalty is relaxed, allowing for more explanatory power from our model, as our coeficient budget is increased. The same logic applies as in question 7 for why the number of coefficients decreases.

Question 9

Produce a plot of the R2-values along the lasso regularization path, i.e. against log lambda and indicate thelog lambda value that gives the highest R2. What do you observe? Is this a surprising observation? (10marks)

```
R2 <- 1 - lasso$cvm/lasso$cvm[1]
plot(log(lasso$lambda),R2, main = "R- Squared along lambda regularisation path")
abline(v = log(lasso$lambda.min))
```



This graph of the R-Squared looks like the reflection of the deviance graph. This makes sense because is is calculated as follows: $R^2 = 1 - \frac{currentmodeldeviance}{nulldeviance}$, and as a result we would expect R2 to be decreasing as deviance increases, and at its highest value when deviance is at its lowest, as is shown by the vertical line representing the optimal value of lambda

Question 10

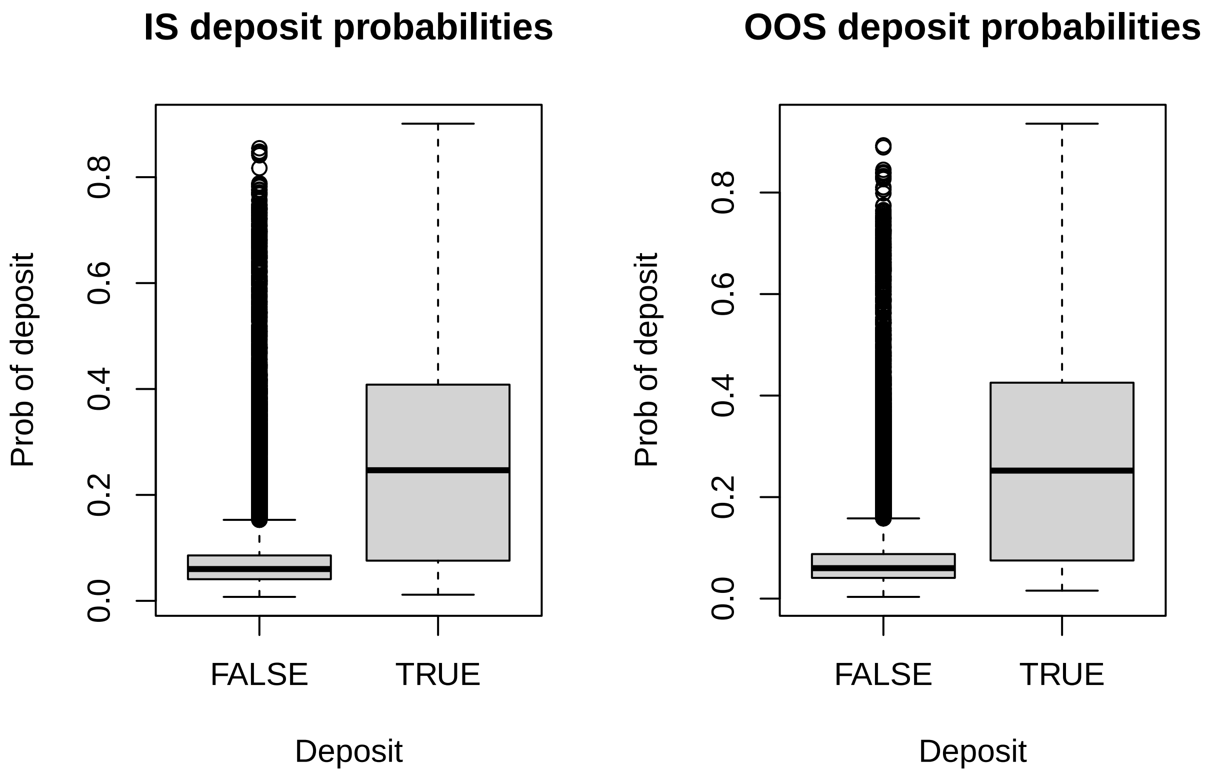
Compute in- and out-of-sample predicted probabilities of y == TRUE using the fitted logit lasso withthe deviance minimizing penalty weight. Produce a table that shows the average predicted probabilityof y == TRUE in training and test data. (5 marks)

```
ISpred <- predict(lasso, newx = sample,type = "response", s = lasso$lambda.min)
OOSpred <- predict(lasso, newx = test, type = "response", s = lasso$lambda.min)
avgIS <- mean(ISpred)
avgOOS  <- mean(OOSpred)
table <- data.frame(Values = c(avgIS, avgOOS))
rownames(table) <- c("Avg In-Sample", "Avg Out-of-Sample")
colnames(table) <- "Prob Y == True"
kable(table, caption = "Probablity Y is true for IS and OOS model predictions")
```

Table 2: Probablity Y is true for IS and OOS model predictions

	Prob Y == True
Avg In-Sample	0.1123628
Avg Out-of-Sample	0.1145118

```
par(mfrow=c(1,2)) # Prepare fig: Arrange fig in 1 row, 2 cols
boxplot(ISpred~y[-split],
        xlab = "Deposit",
        ylab = "Prob of deposit",
        main = "IS deposit probabilities") # Plot dist pDef by actual default
# Default probabilities (out-of-sample)
boxplot(OOSpred~y[split],
        xlab = "Deposit",
        ylab = "Prob of deposit",
        main = "OOS deposit probabilities") # Plot dist pDef by actual default
```



These box plots represent the models in and out of sample fits. They demonstrate that there is a large amount of overlap between probabilities for true deposits and non deposits in both models. This means that there will likely be a number of false positives and negatives from our model.

Question 11

Compute and report out-of-sample specificity and sensitivity for the classification rule pstar = 0.5. Interpretthe two statistics. What does 1 - sensitivity measure? (10 marks)

```
senspe = matrix(0,2,2) # Initialize matrix of 0s
rownames(senspe) <- c("IS","OOS") # Label rows
colnames(senspe) <- c("Sensitivity","Specificity") # Label cols
# Sensitivity
prule <- 0.5
senspe[1,1] <- mean((ISpred > prule)[y[-split] == 1])
senspe[2,1] <- mean((OOSpred > prule)[y[split] == 1])
# Specificity
senspe[1,2] <- mean((ISpred < prule)[y[-split] == 0])
senspe[2,2] <- mean((OOSpred < prule)[y[split] == 0])
kable(senspe, caption = "sensitivity and specificity for p* = 0.5")
```

Table 3: sensitivity and specificity for p* = 0.5

	Sensitivity	Specificity
IS	0.1957649	0.9878556
OOS	0.2098022	0.9878476

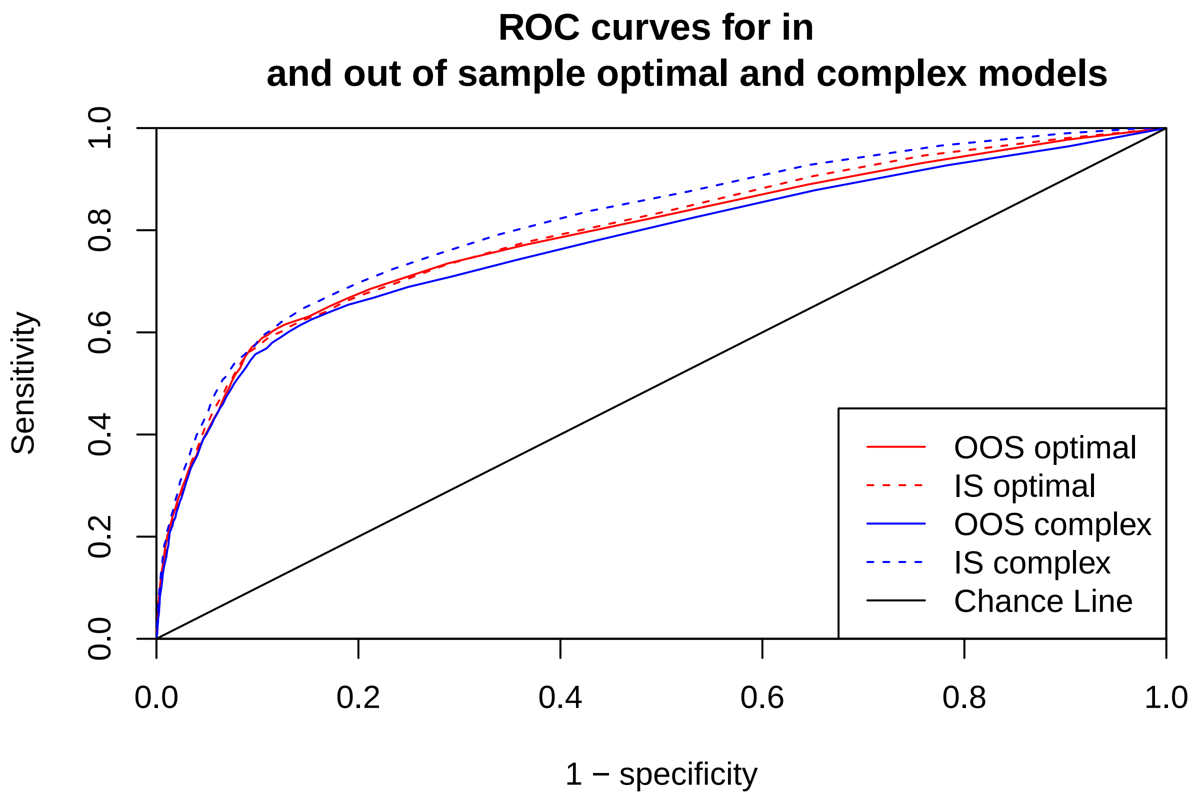
Sensitivity reflects the proportion of true y = 1 that are classified as such, and specificity reflects the proportion of the true y = 0 that are classified as such. So for this data, the model successfully classifies 21.1% of in sample, and 19.6% of out of sample data points where y = 1, or true, and successfully classifies 98.8% of both in and out of sample data where why = 0 or false. It should also be noted that this is the case at the classification rule of p = 0.5, or when the probability is > 50%, y is classified as true, and when the probability < 50%, false. sensitivity and specificity rates will change for different values of p*. So for this data, our model is very specific, but not very sensitive. 1 - sensitivity would represent the false negative rate, or the rate at which values that are actually true are classified as false. By plotting sensitivity againt 1-specificity, we are modelling the tradeoff of sensitivity improving against the false positive rate.

Question 12 & 13

Produce and plot an out-of-sample ROC-curve for the fitted lasso logit (10 marks)

```
lowest <- min(lasso$lambda)
ISpred.complex <- predict(lasso, newx = sample,type = "response", s = lowest)
OOSpred.complex <- predict(lasso, newx = test, type = "response", s = lowest)
my.rocdat <- data.frame(rule = rep(NA,101), senis = rep(NA,101),compsensis
                        = rep(NA,101),
                        speis = rep(NA,101), senoos = rep(NA,101), specoos =
                          rep(NA,101),
                        compspecis = rep(NA,101),compsens = rep(NA, 101))
for (this.rule in 0:100) {
  this.prule <- this.rule/100
  my.rocdat[this.rule+1,"rule"] <- this.prule
  my.rocdat[this.rule+1,"compsens"] <- mean((OOSpred.complex >= this.prule)
                                      [y[split] == 1])
  my.rocdat[this.rule+1,"compspec"] <- mean((OOSpred.complex < this.prule)
                                      [y[split] == 0])
  my.rocdat[this.rule+1,"compsensis"] <- mean((ISpred.complex >= this.prule)
                                        [y[-split] == 1])
  my.rocdat[this.rule+1,"compspecis"] <- mean((ISpred.complex < this.prule)
                                        [y[-split] == 0])
  my.rocdat[this.rule+1, "senis"] <- mean((ISpred >= this.prule)[y[-split] == 1])
  my.rocdat[this.rule+1, "specis"] <- mean((ISpred < this.prule)[y[-split] == 0])
  my.rocdat[this.rule+1,"senoos"] <- mean((OOSpred >= this.prule)[y[split] == 1])
  my.rocdat[this.rule+1,"specoos"] <- mean((OOSpred < this.prule)[y[split] == 0])
  }

plot(1-my.rocdat$specoos,my.rocdat$senoos,
     type = "l", lty = 1, lwd = 1, col = "red",
     xlim = c(0,1), ylim = c(0,1), xaxs = "i", yaxs = "i",
     xlab = "1 - specificity", ylab =  "Sensitivity",main = "ROC curves for in
     and out of sample optimal and complex models")

lines(1-my.rocdat$specis,my.rocdat$senis,
      type = "l", lty = 2, lwd = 1, col = "red")
```

```r
lines(1-my.rocdat$compspecis,my.rocdat$compsensis,
      type = "l", lty = 2, lwd = 1, col = "blue")

lines(1-my.rocdat$compspec,my.rocdat$compsens,
      type = "l", lty = 1, lwd = 1, col = "blue")

abline(0, 1, lty = 1)

legend("bottomright",
       legend=c("OOS optimal","IS optimal","OOS complex","IS complex", "Chance Line" ),
       col=c("red","red", "blue","blue", "black"),
        lty=c(1,2,1,2,1), lwd = c(1,1,1,1,1))
```



This graph shows the ROC curves for in and out of sample complex and optimal models. The chance line represents the line at which the model is no better at predicting than random chance would be, so the further away from it, the better the model is at predicting, the better at obtaining sensitivity, while maintaining specificity we are. If you pick a point on the line, it will show you, for a specific p*, what your sensitivity - specificity tradeoff will be. The gradient of the curve therefore represents how much sensitivity is gained for the loss of specificity. Only looking at OOS predictions, we can see that the optimal model is further from the chance line than the complex model, showing that is has better predictive powers for out of sample predictions. This is likely because by removing complexity, it reduces the impact of overfitting, making it a more accurate general model. When comparing the OOS models to their IS counterparts, it's evident that both IS models exhibit better performance. This underscores the challenge of overfitting, where models might excel on their training data but falter when faced with new data.

Question 14

A member of your team suggests to use the area under the ROC-curve as an alternative way of crossvalidating the oos predictions along the regularization path. Is this a sensible suggestion? Substantiate your answer, but keep it brief. (5 marks)

Measuring the area under the curve is a useful way of measuring a models strength. The area under the ROC curves represents a way to measure the model's strength at discriminating between positive and negative cases, in this example between a positive and negative deposit. The greater the area under the curve, the further the overall distance from the chance line the model provides, and therefore the greater overall predictive power it is likely to have. One advantage of this method over the p threshold method, is that is doesn't rely on only looking at one value of p*, and instead takes an overall look at the models strength. Looking at how the area under the ROC curve changes over the regularization path can help us understand a models changing predictive power for this reason. The tradeoff between complexity and performance can be tracked and optimised in this way.