

# Modeling

*Shukry Zablah*

*06 December, 2018*

## Contents

Imports . . . . .	1
Load Data . . . . .	1
Baseline . . . . .	1
Logistic Regression Classifier (Full) . . . . .	2
Training . . . . .	2
Model Evaluation on Train Set . . . . .	2
Logistic Regression Classifier (Optimized) . . . . .	4
Training . . . . .	4
Model Evaluation on Train Set . . . . .	4
Random Forest Classifier . . . . .	6
Training . . . . .	6
Model Evaluation on the Train Set . . . . .	6

## Imports

```
library(MASS)
library(dplyr)
library(mosaic)
library(ROCR)
library(randomForest)
library(caret)
library(reprtree) #devtools::install_github('araastat/reprtree')
```

## Load Data

```
train <- readRDS(file = "../data/PIMA_train.Rds")
test <- readRDS(file = "../data/PIMA_test.Rds")
```

## Baseline

```
tally(~ hasDiabetes, data = train)
```

```
## hasDiabetes
##    0    1
## 196  98
```

Any model we choose has to have an accuracy higher than  $98/(98+196) = 33\%$ . This is the baseline accuracy score.

## Logistic Regression Classifier (Full)

### Training

```
set.seed(1)
Fit_LR_Full <- train(hasDiabetes ~ ., data = train,
                     method = "glm",
                     trControl = trainControl(method = "none"))
clf_LR_Full <- with(Fit_LR_Full, finalModel)
#this is the same model as glm(hasDiabetes ~ ., family = binomial(link = "logit"), data = train)

summary(clf_LR_Full)

##
## Call:
## NULL
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.8732  -0.6344  -0.3526   0.5918   2.2488
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.090e+01  1.516e+00  -7.190 6.48e-13 ***
## pregnancies      1.310e-01  6.738e-02   1.944  0.05192 .
## glucoseConcentration  4.080e-02  6.740e-03   6.054 1.41e-09 ***
## bloodPressure      2.284e-03  1.367e-02   0.167  0.86732
## skinThickness     -3.725e-06  2.118e-02   0.000  0.99986
## insulin          -1.764e-03  1.544e-03  -1.142  0.25325
## bmi               9.483e-02  3.472e-02   2.731  0.00631 **
## diabetesPedigreeFunction  1.472e+00  4.967e-01   2.963  0.00305 **
## age               2.227e-02  2.099e-02   1.061  0.28881
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 374.27  on 293  degrees of freedom
## Residual deviance: 251.84  on 285  degrees of freedom
## AIC: 269.84
##
## Number of Fisher Scoring iterations: 5
saveRDS(clf_LR_Full, file = "../models/LogisticRegressionClassifier_Full.Rds")
```

### Model Evaluation on Train Set

```
predict_LR_Full <- predict(clf_LR_Full, type = 'response')

with(train,
      table(hasDiabetes, predict_LR_Full > 0.3))

##
```

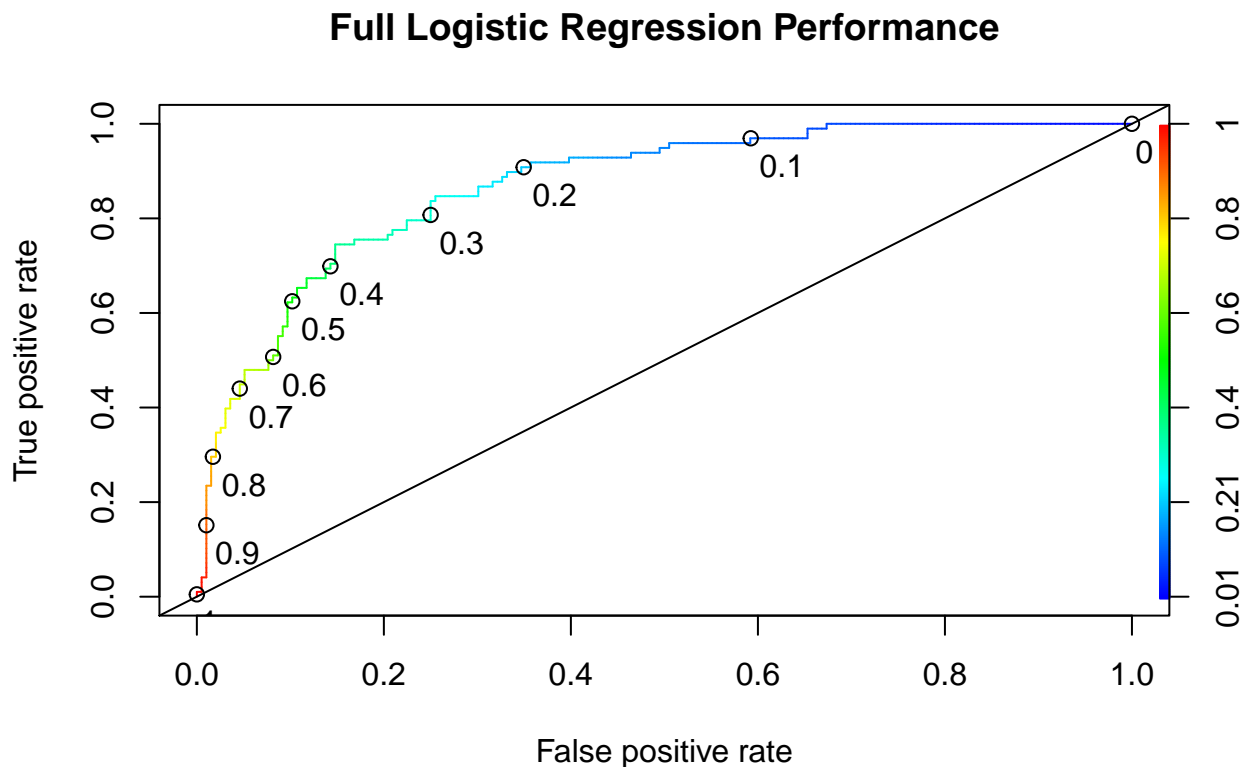
```
## hasDiabetes FALSE TRUE
##           0    147    49
##           1     19    79
```

We can see in the confusion matrix that our accuracy is  $(79 + 147)/(147 + 49 + 19 + 79) = 0.7687$ . This is with a cutoff of 0.3.

```
ROCpred_LR_Full <- with(train,
                        prediction(predict_LR_Full, hasDiabetes))

ROCperf_LR_Full <- performance(ROCpred_LR_Full, 'tpr', 'fpr')

plot(ROCperf_LR_Full,
     colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),
     text.adj = c(-0.2,1.7),
     main = "Full Logistic Regression Performance")
abline(0,1)
```



In the ROC curve we can see that our model is good (the curve is away from the diagonal). Since we care about not predicting a negative result for someone that is actually positive for diabetes (false negative rate), we want to have a larger true positive rate ( $1 - \text{TPR} = \text{FNR}$ ). This means that we choose a cutoff near the blue part of the curve, the lower the cutoff the more cautious our model and the less accurate.

## Logistic Regression Classifier (Optimized)

### Training

```
clf_LR_Op <- MASS::stepAIC(glm(hasDiabetes ~ 1, family = binomial(link='logit'),
                               data=train),
                           list(upper = ~ pregnancies +
                                glucoseConcentration +
                                bloodPressure +
                                skinThickness +
                                insulin +
                                bmi +
                                diabetesPedigreeFunction +
                                age),
                               direction="both",
                               trace = FALSE)
summary(clf_LR_Op)

##
## Call:
## glm(formula = hasDiabetes ~ glucoseConcentration + bmi + diabetesPedigreeFunction +
##      pregnancies, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.1901  -0.6546  -0.3663   0.6179   2.3511
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -10.069964    1.258507  -8.002 1.23e-15 ***
## glucoseConcentration    0.038765    0.005836   6.642 3.09e-11 ***
## bmi                0.090308    0.024662   3.662 0.000250 ***
## diabetesPedigreeFunction  1.495673    0.486638   3.073 0.002116 **
## pregnancies        0.179393    0.049746   3.606 0.000311 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 374.27  on 293  degrees of freedom
## Residual deviance: 254.61  on 289  degrees of freedom
## AIC: 264.61
##
## Number of Fisher Scoring iterations: 5
saveRDS(clf_LR_Op, file = "../models/LogisticRegressionClassifier_Optimized.Rds")
```

### Model Evaluation on Train Set

```
predict_LR_Op <- predict(clf_LR_Op, type = 'response')

with(train,
      table(hasDiabetes, predict_LR_Op > 0.3))
```

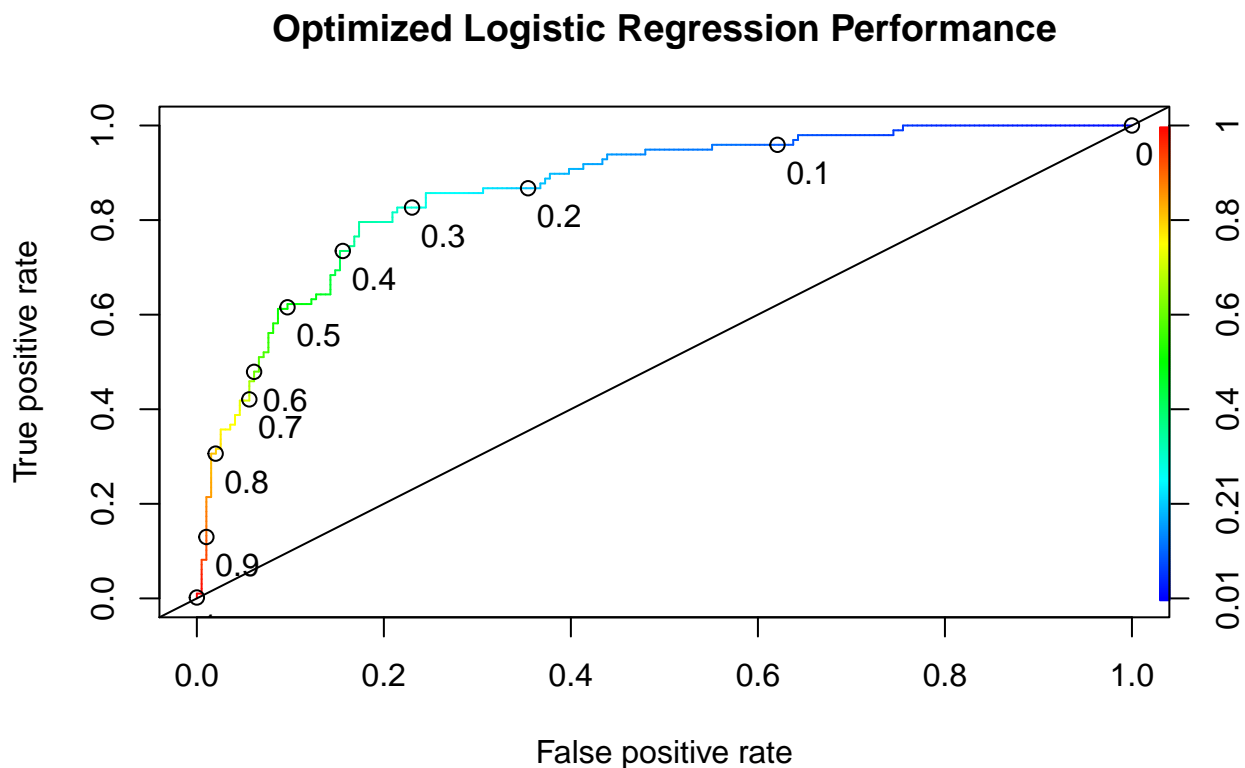
```
##
## hasDiabetes FALSE TRUE
##           0    151   45
##           1     17   81
```

We can see in the confusion matrix that our accuracy is  $(151 + 81)/(151 + 45 + 17 + 81) = 0.7891$ . This is with a cutoff of 0.3.

```
ROCRpred_LR_Op <- with(train,
                        prediction(predict_LR_Op, hasDiabetes))

ROCRperf_LR_Op <- performance(ROCRpred_LR_Op, 'tpr', 'fpr')

plot(ROCRperf_LR_Op,
     colorize = TRUE,
     print.cutoffs.at = seq(0,1,0.1),
     text.adj = c(-0.2,1.7),
     main = "Optimized Logistic Regression Performance")
abline(0,1)
```



In the ROC curve we can see that our model is good (the curve is away from the diagonal). Comparing this to our previous full logistic regression model we can see that the models are almost the same, and the optimized version has the advantage of being simpler.

## Random Forest Classifier

### Training

```
clf_RF <- randomForest(hasDiabetes ~ ., data = train)
saveRDS(clf_RF, file = "../models/RandomForestClassifier_Full.Rds")
```

### Model Evaluation on the Train Set

```
predict_RF <- predict(clf_RF, type = 'prob')

with(train,
  table(hasDiabetes, (predict_RF %>% as_tibble() %>% dplyr::select(`1`)) > 0.3))
```

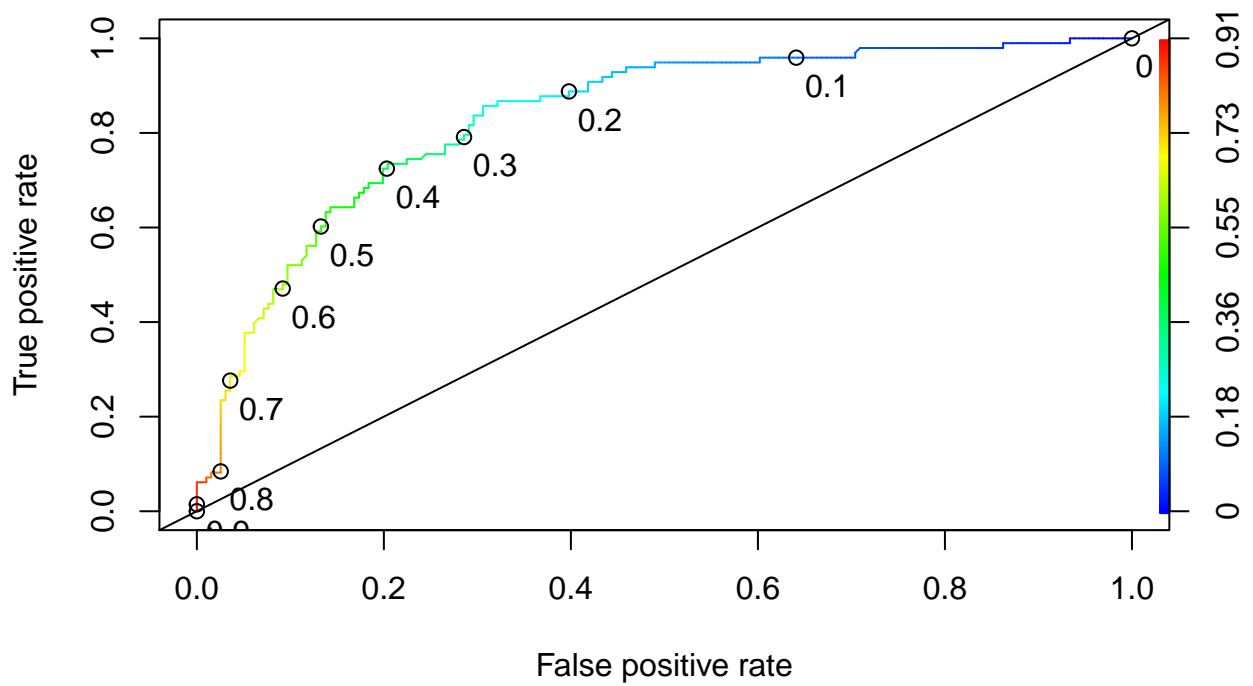
```
##
## hasDiabetes FALSE TRUE
##           0   140   56
##           1    21   77
```

We can see in the confusion matrix that our accuracy is  $(140 + 83)/(140 + 56 + 15 + 83) = 0.7585$ . The accuracy is slightly lower than the logistic regression model.

```
ROCRpred_RF <- with(train,
  prediction(predict_RF %>% as_tibble() %>% dplyr::select(`1`), hasDiabetes))
```

```
ROCRperf_RF <- performance(ROCRpred_RF, 'tpr', 'fpr')
```

```
plot(ROCRperf_RF, colorize = TRUE, print.cutoffs.at = seq(0,1,0.1), text.adj = c(-0.2,1.7)); abline(0,1)
```



```
# plot only part of the representative tree
#reprtree::plot.getTree(clf_RF, depth = 5)
```

The random forest ROC curve is less steep than the logistic regression roc curves. We still want to choose a low cutoff like 0.3. They all perform similarly on the train set.