STA 141C Project

Ryan Cosgrove, Eric Kye, Ravinit Chand, Revanth Rao

2024-05-11

Logistic Regression Model

```
weather = read.csv("weatherAUS.csv")
```

```
weather = weather %>% mutate_at(c('WindGustDir', 'WindDir9am', 'WindDir3pm',
                                  'RainToday', 'RainTomorrow'), as.factor)
weather$Year = as.integer(sapply(strsplit(weather[,1], "-"), getElement, 1))
summary(weather)
```

```
##       Date              Location            MinTemp          MaxTemp
##  Length:145460      Length:145460      Min.   :-8.50   Min.   :-4.80
##  Class :character   Class :character   1st Qu.: 7.60   1st Qu.:17.90
##  Mode  :character   Mode  :character   Median :12.00   Median :22.60
##                                        Mean   :12.19   Mean   :23.22
##                                        3rd Qu.:16.90   3rd Qu.:28.20
##                                        Max.   :33.90   Max.   :48.10
##                                        NA's   :1485    NA's   :1261
##     Rainfall        Evaporation        Sunshine      WindGustDir
##  Min.   :  0.000   Min.   :  0.00   Min.   : 0.00   W      : 9915
##  1st Qu.:  0.000   1st Qu.:  2.60   1st Qu.: 4.80   SE     : 9418
##  Median :  0.000   Median :  4.80   Median : 8.40   N      : 9313
##  Mean   :  2.361   Mean   :  5.47   Mean   : 7.61   SSE    : 9216
##  3rd Qu.:  0.800   3rd Qu.:  7.40   3rd Qu.:10.60   E      : 9181
##  Max.   :371.000   Max.   :145.00   Max.   :14.50   (Other):88091
##  NA's   :3261      NA's   :62790    NA's   :69835   NA's   :10326
##  WindGustSpeed      WindDir9am      WindDir3pm     WindSpeed9am
##  Min.   :  6.00   N      :11758   SE     :10838   Min.   :  0.00
##  1st Qu.: 31.00   SE     : 9287   W      :10110   1st Qu.:  7.00
##  Median : 39.00   E      : 9176   S      : 9926   Median : 13.00
##  Mean   : 40.03   SSE    : 9112   WSW    : 9518   Mean   : 14.04
##  3rd Qu.: 48.00   NW     : 8749   SSE    : 9399   3rd Qu.: 19.00
##  Max.   :135.00   (Other):86812   (Other):91441   Max.   :130.00
##  NA's   :10263    NA's   :10566   NA's   : 4228   NA's   :1767
##   WindSpeed3pm    Humidity9am      Humidity3pm      Pressure9am
##  Min.   : 0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 980.5
##  1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00   1st Qu.:1012.9
##  Median :19.00   Median : 70.00   Median : 52.00   Median :1017.6
##  Mean   :18.66   Mean   : 68.88   Mean   : 51.54   Mean   :1017.6
##  3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 66.00   3rd Qu.:1022.4
```

```
##  Max.   :87.00   Max.   :100.00   Max.   :100.00   Max.   :1041.0
##  NA's   :3062    NA's   :2654     NA's   :4507     NA's   :15065
##   Pressure3pm        Cloud9am        Cloud3pm        Temp9am
##  Min.   : 977.1   Min.   :0.00    Min.   :0.00    Min.   :-7.20
##  1st Qu.:1010.4   1st Qu.:1.00    1st Qu.:2.00    1st Qu.:12.30
##  Median :1015.2   Median :5.00    Median :5.00    Median :16.70
##  Mean   :1015.3   Mean   :4.45    Mean   :4.51    Mean   :16.99
##  3rd Qu.:1020.0   3rd Qu.:7.00    3rd Qu.:7.00    3rd Qu.:21.60
##  Max.   :1039.6   Max.   :9.00    Max.   :9.00    Max.   :40.20
##  NA's   :15028    NA's   :55888   NA's   :59358   NA's   :1767
##     Temp3pm       RainToday     RainTomorrow       Year
##  Min.   :-5.40   No  :110319   No  :110316   Min.   :2007
##  1st Qu.:16.60   Yes : 31880   Yes : 31877   1st Qu.:2011
##  Median :21.10   NA's:  3261   NA's:  3267   Median :2013
##  Mean   :21.68                               Mean   :2013
##  3rd Qu.:26.40                               3rd Qu.:2015
##  Max.   :46.70                               Max.   :2017
##  NA's   :3609
```

Use a training set that has data from before 2013 and a test set with data after 2013. We remove the variables `WindGustDir`, `WindDir9am`, `WindDir3pm` that just tells us the wind direction at certain times and then generate a GLM Regression model on the training set with the remaining variables and use it to predict if it is going to Rain tomorrow on the Test Set.

```
train_index = (weather$Year < 2013)
test_index = !train_index

train = weather[train_index, ]
test = weather[test_index, ]

# Remove columns
train = train[, c(-1, -2, -8, -10, -11, -24)]
test = test[, c(-1, -2, -8, -10, -11, -24)]

# Remove NAs
train = na.omit(train)
test = na.omit(test)

RainTom.test <- test$RainTomorrow

# GLM Model
glm.fits <- glm(RainTomorrow ~ ., data = train, family = binomial)
glm.fits
```

```
##
## Call:  glm(formula = RainTomorrow ~ ., family = binomial, data = train)
##
## Coefficients:
##   (Intercept)        MinTemp        MaxTemp       Rainfall    Evaporation
##    56.2998671     -0.0478350     -0.0001738      0.0126430     -0.0017503
##      Sunshine  WindGustSpeed   WindSpeed9am   WindSpeed3pm    Humidity9am
##    -0.1410623      0.0608414     -0.0099919     -0.0282713      0.0020836
##   Humidity3pm    Pressure9am    Pressure3pm       Cloud9am       Cloud3pm
```

```
##     0.0573718      0.1513636     -0.2137042     -0.0158576      0.1260501
##       Temp9am        Temp3pm   RainTodayYes
##     0.0492442      0.0046234      0.4284623
##
## Degrees of Freedom: 31668 Total (i.e. Null);  31651 Residual
## Null Deviance:       33700
## Residual Deviance: 20990     AIC: 21030
```

```
glm.probs <- predict(glm.fits, test, type = "response")
```

```
glm.pred <- rep("No", length(glm.probs))
glm.pred[glm.probs > .5] <- "Yes"
table(glm.pred, RainTom.test)
```

```
##         RainTom.test
## glm.pred    No   Yes
##      No  19686  2728
##      Yes  1105  2902
```

```
mean(glm.pred == RainTom.test)
```

```
## [1] 0.854926
```

```
mean(glm.pred != RainTom.test)
```

```
## [1] 0.145074
```

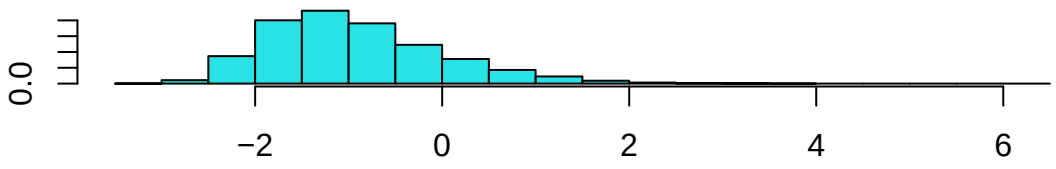
Linear Discriminant Analysis

```
lda.fit <- lda(RainTomorrow ~ ., data = train)
lda.fit
```

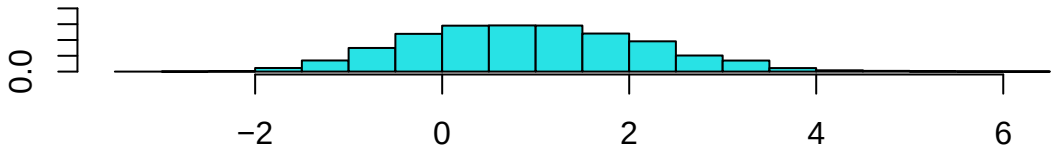
```
## Call:
## lda(RainTomorrow ~ ., data = train)
##
## Prior probabilities of groups:
##        No       Yes
## 0.7758376 0.2241624
##
## Group means:
##      MinTemp  MaxTemp Rainfall Evaporation Sunshine WindGustSpeed WindSpeed9am
## No  12.52694 24.25628 1.197285    5.434355 8.557794      38.88225     14.84953
## Yes 14.30008 22.07656 5.815833    4.435526 4.414861      46.34780     16.76884
##     WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am
## No      19.23386    64.22145    44.89251    1018.214    1015.675 3.775173
## Yes     21.13706    76.08635    67.62290    1013.794    1011.646 6.022961
##     Cloud3pm  Temp9am  Temp3pm RainTodayYes
## No  3.824786 17.69324 22.92026    0.1531136
```

```
## Yes 6.334272 17.77133 20.16353    0.4699253
##
## Coefficients of linear discriminants:
##                        LD1
## MinTemp       -0.043234050
## MaxTemp        0.048836234
## Rainfall       0.013509768
## Evaporation    0.014764696
## Sunshine      -0.132892322
## WindGustSpeed  0.040882713
## WindSpeed9am  -0.002372743
## WindSpeed3pm  -0.027360965
## Humidity9am   -0.004433778
## Humidity3pm    0.042286151
## Pressure9am    0.095342751
## Pressure3pm   -0.137223770
## Cloud9am      -0.030950057
## Cloud3pm       0.028058901
## Temp9am       -0.001911742
## Temp3pm       -0.007092874
## RainTodayYes   0.427766488
```

```
plot(lda.fit)
```



group No



group Yes

```
lda.pred <- predict(lda.fit, test)

lda.class <- lda.pred$class
table(lda.class, RainTom.test)
```

```
##          RainTom.test
## lda.class    No   Yes
```

```
##       No  19592  2657
##       Yes  1199  2973
```

```
mean(lda.class == RainTom.test)
```

```
## [1] 0.8540555
```

```
###
sum(lda.pred$posterior[, 1] >= .5)
```

```
## [1] 22249
```

```
sum(lda.pred$posterior[, 1] < .5)
```

```
## [1] 4172
```

```
###
lda.pred$posterior[1:20, 1]
```

```
##      10464      10465      10466      10467      10472      10473      10474
## 0.96799235 0.97727137 0.62769121 0.31341585 0.07218133 0.31421158 0.93979174
##      10478      10479      10480      10481      10488      10490      10492
## 0.08184123 0.31678701 0.73803694 0.78757038 0.89154952 0.19334619 0.21560016
##      10493      10494      10495      10500      10501      10502
## 0.17698203 0.91268072 0.90905031 0.98303784 0.63239428 0.96169231
```

```
lda.class[1:20]
```

```
##  [1] No  No  No  Yes Yes Yes No  Yes Yes No  No  No  Yes Yes Yes No  No  No  No
## [20] No
## Levels: No Yes
```

```
###
sum(lda.pred$posterior[, 1] > .9)
```

```
## [1] 15875
```

Quadratic Discriminant Analysis

```
qda.fit <- qda(RainTomorrow ~ ., data = train)
qda.fit
```

```
## Call:
## qda(RainTomorrow ~ ., data = train)
##
## Prior probabilities of groups:
##        No       Yes
```

```
## 0.7758376 0.2241624
##
## Group means:
##      MinTemp  MaxTemp Rainfall Evaporation Sunshine WindGustSpeed WindSpeed9am
## No  12.52694 24.25628 1.197285    5.434355 8.557794      38.88225     14.84953
## Yes 14.30008 22.07656 5.815833    4.435526 4.414861      46.34780     16.76884
##     WindSpeed3pm Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am
## No      19.23386    64.22145    44.89251    1018.214    1015.675 3.775173
## Yes     21.13706    76.08635    67.62290    1013.794    1011.646 6.022961
##     Cloud3pm  Temp9am  Temp3pm RainTodayYes
## No  3.824786 17.69324 22.92026    0.1531136
## Yes 6.334272 17.77133 20.16353    0.4699253
```

```
qda.class <- predict(qda.fit, test)$class
table(qda.class, RainTom.test)
```

```
##          RainTom.test
## qda.class    No   Yes
##       No  18961  2470
##       Yes  1830  3160
```

```
mean(qda.class == RainTom.test)
```

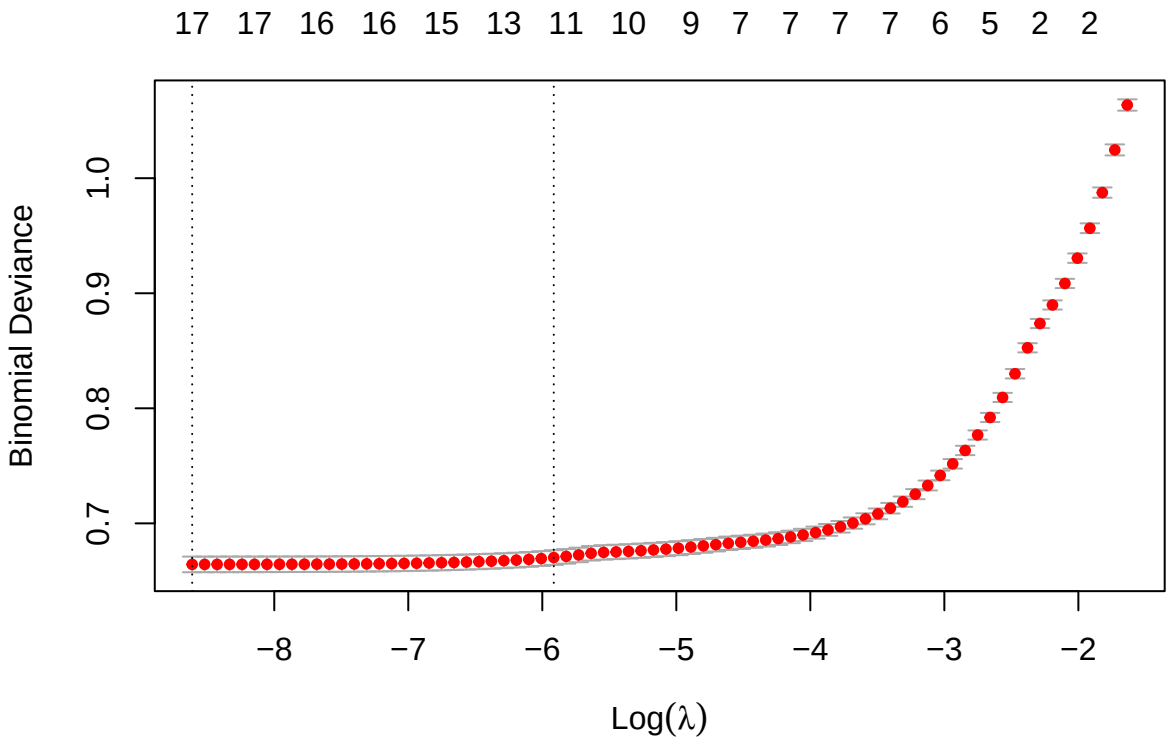
```
## [1] 0.8372507
```

Lasso Regression

```
# Recreate x and y after removing NA rows from train and test
x <- model.matrix(RainTomorrow ~ ., rbind(train, test))[,-1]
y <- as.numeric(rbind(train, test)$RainTomorrow) - 1

train_rows <- 1:nrow(train)
test_rows <- (nrow(train) + 1):nrow(x)

lasso.fit <- cv.glmnet(x[train_rows, ], y[train_rows], family = "binomial", alpha = 1)
plot(lasso.fit)
```




```
lasso.pred <- predict(lasso.fit, s = "lambda.min", newx = x[test_rows, ], type = "class")
table(lasso.pred, RainTom.test)
```

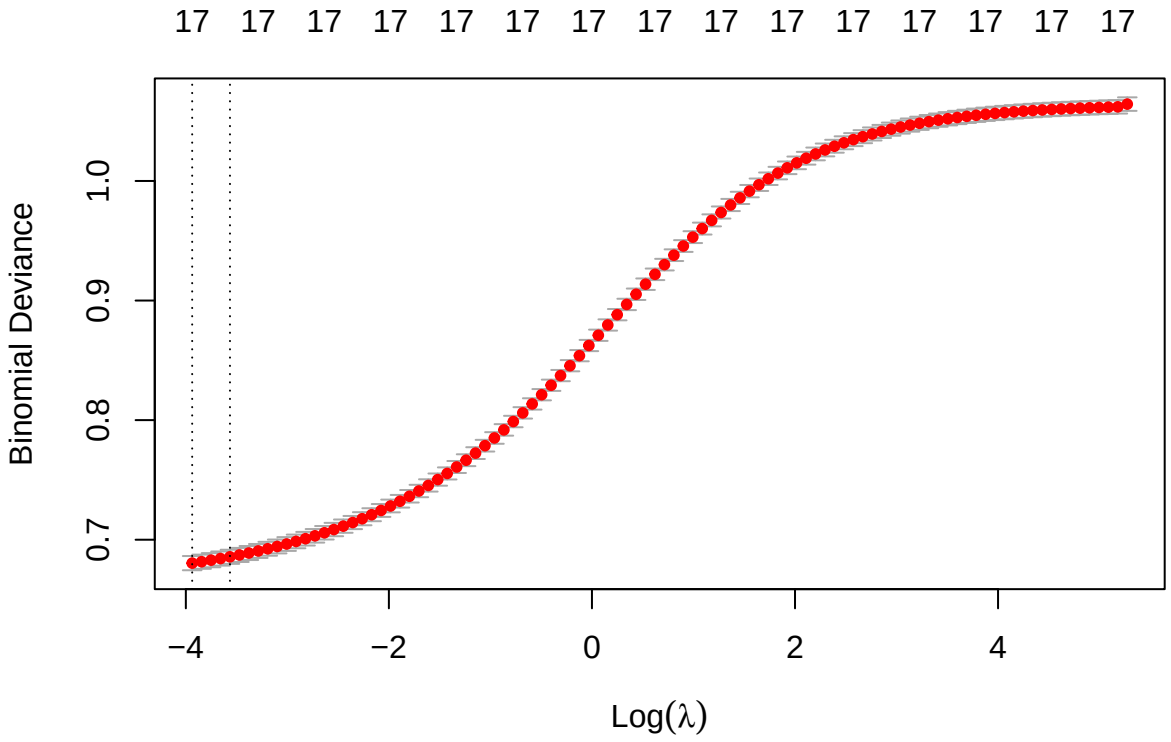
```
##           RainTom.test
## lasso.pred    No   Yes
##          0 19687  2725
##          1  1104  2905
```

```
mean(lasso.pred == RainTom.test)
```

```
## [1] 0
```

Ridge Regression

```
ridge.fit <- cv.glmnet(x[train_rows, ], y[train_rows], family = "binomial", alpha = 0)
plot(ridge.fit)
```




```
ridge.pred <- predict(ridge.fit, s = "lambda.min", newx = x[test_rows, ], type = "class")
table(ridge.pred, RainTom.test)
```

```
##           RainTom.test
## ridge.pred    No   Yes
##          0 19778  2933
##          1  1013  2697
```

```
mean(ridge.pred == RainTom.test)
```

```
## [1] 0
```

Random Forest

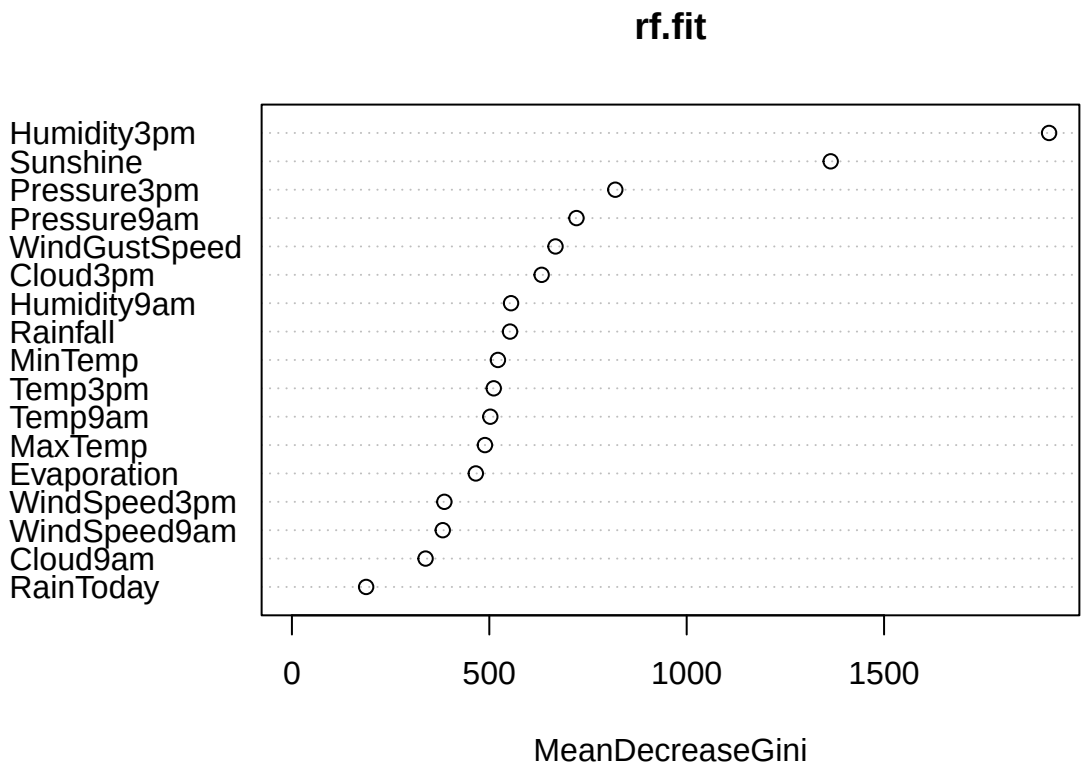
```
rf.fit <- randomForest(RainTomorrow ~ ., data = train)
rf.pred <- predict(rf.fit, newdata = test)
table(rf.pred, RainTom.test)
```

```
##        RainTom.test
## rf.pred    No   Yes
##     No  19792  2767
##     Yes   999  2863
```

```r
mean(rf.pred == RainTom.test)
```

```
## [1] 0.8574619
```

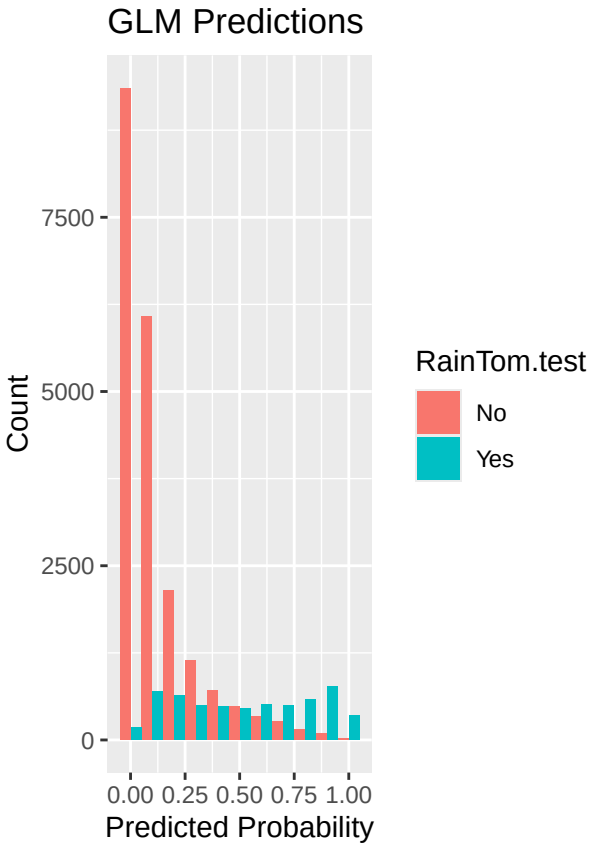
```r
# Variable Importance Plot
varImpPlot(rf.fit)
```

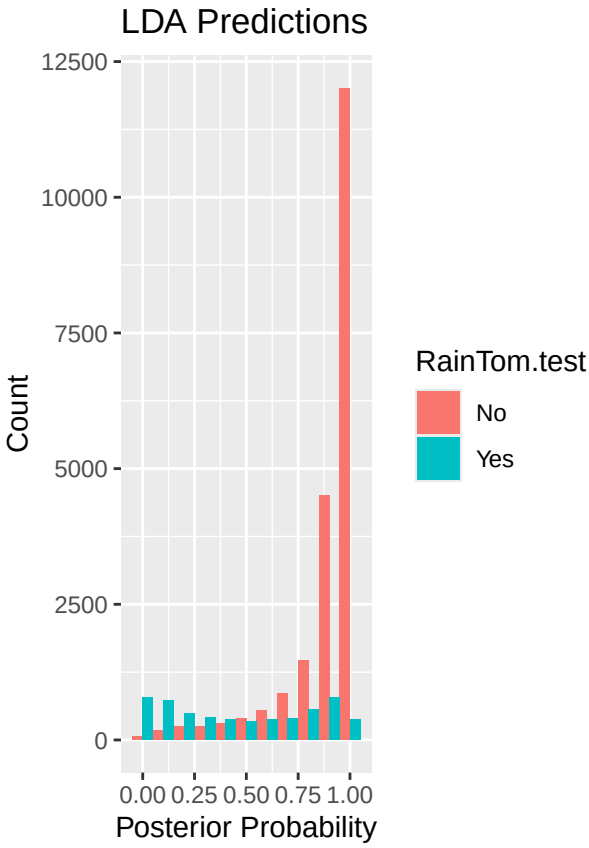



```r
# Plot GLM Predictions
glm_pred_plot <- ggplot(data = test, aes(x = glm.probs, fill = RainTom.test)) +
  geom_histogram(binwidth = 0.1, position = "dodge") +
  labs(title = "GLM Predictions", x = "Predicted Probability", y = "Count")

# Plot LDA Predictions
lda_pred_plot <- ggplot(data = test, aes(x = lda.pred$posterior[,1], fill = RainTom.test)) +
  geom_histogram(binwidth = 0.1, position = "dodge") +
  labs(title = "LDA Predictions", x = "Posterior Probability", y = "Count")

# Arrange plots
grid.arrange(glm_pred_plot, lda_pred_plot, ncol = 2)
```


GLM Predictions
Count
7500
5000
2500
0
0.00 0.25 0.50 0.75 1.00
Predicted Probability
RainTom.test
No
Yes


LDA Predictions
Count
12500
10000
7500
5000
2500
0
0.00 0.25 0.50 0.75 1.00
Posterior Probability
RainTom.test
No
Yes