STA141C Final Project

Ryan Cosgrove, Ravinit Chand, Eric Kye, Revanth Rao

2024-06-07

Libraries

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3

## Warning: package 'tidyr' was built under R version 4.2.3

## Warning: package 'readr' was built under R version 4.2.3

## Warning: package 'dplyr' was built under R version 4.2.3

## Warning: package 'stringr' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(glmnet)
```

```
## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.2.3

##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-8
```

```r
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.2.3
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(ROCR)
library(vtable)
```

```
## Loading required package: kableExtra
```

```
## Warning: package 'kableExtra' was built under R version 4.2.3
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.2.3
```

```
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

Loading Data and Preprocessing

```r
weather = read.csv("/Users/eric/Documents/Classroom/23-24/3 Spring/STA141C/Project/weatherAUS.csv")

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

Splitting Data and Removing Variables

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

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

Remove columns

```r
train = train[, c(-1, -2, -8, -10, -11, -24)]
test = test[, c(-1, -2, -8, -10, -11, -24)]
weather = weather[, c(-1, -2, -8, -10, -11, -24)]
weather_plotting = weather[, c(-17, -18)]
```

Table 1: Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
MinTemp	58090	13	6.5	-6.7	8.4	18	31
MaxTemp	58090	24	7	4.1	19	30	48
Rainfall	58090	2.1	7	0	0	0.6	206
Evaporation	58090	5.4	3.7	0	2.8	7.4	81
Sunshine	58090	7.7	3.8	0	5	11	14
WindGustSpeed	58090	41	13	9	31	48	124
WindSpeed9am	58090	15	8.6	0	9	20	67
WindSpeed3pm	58090	20	8.6	0	13	24	76
Humidity9am	58090	66	19	0	55	80	100
Humidity3pm	58090	50	20	0	36	63	100
Pressure9am	58090	1017	6.9	980	1013	1022	1040
Pressure3pm	58090	1015	6.9	977	1010	1020	1039
Cloud9am	58090	4.2	2.8	0	1	7	8
Cloud3pm	58090	4.3	2.7	0	2	7	9
Temp9am	58090	18	6.6	-0.9	13	23	39
Temp3pm	58090	23	6.8	3.7	17	28	46
RainToday	58090						
... No	45323	78%					
... Yes	12767	22%					
RainTomorrow	58090						
... No	45361	78%					
... Yes	12729	22%					

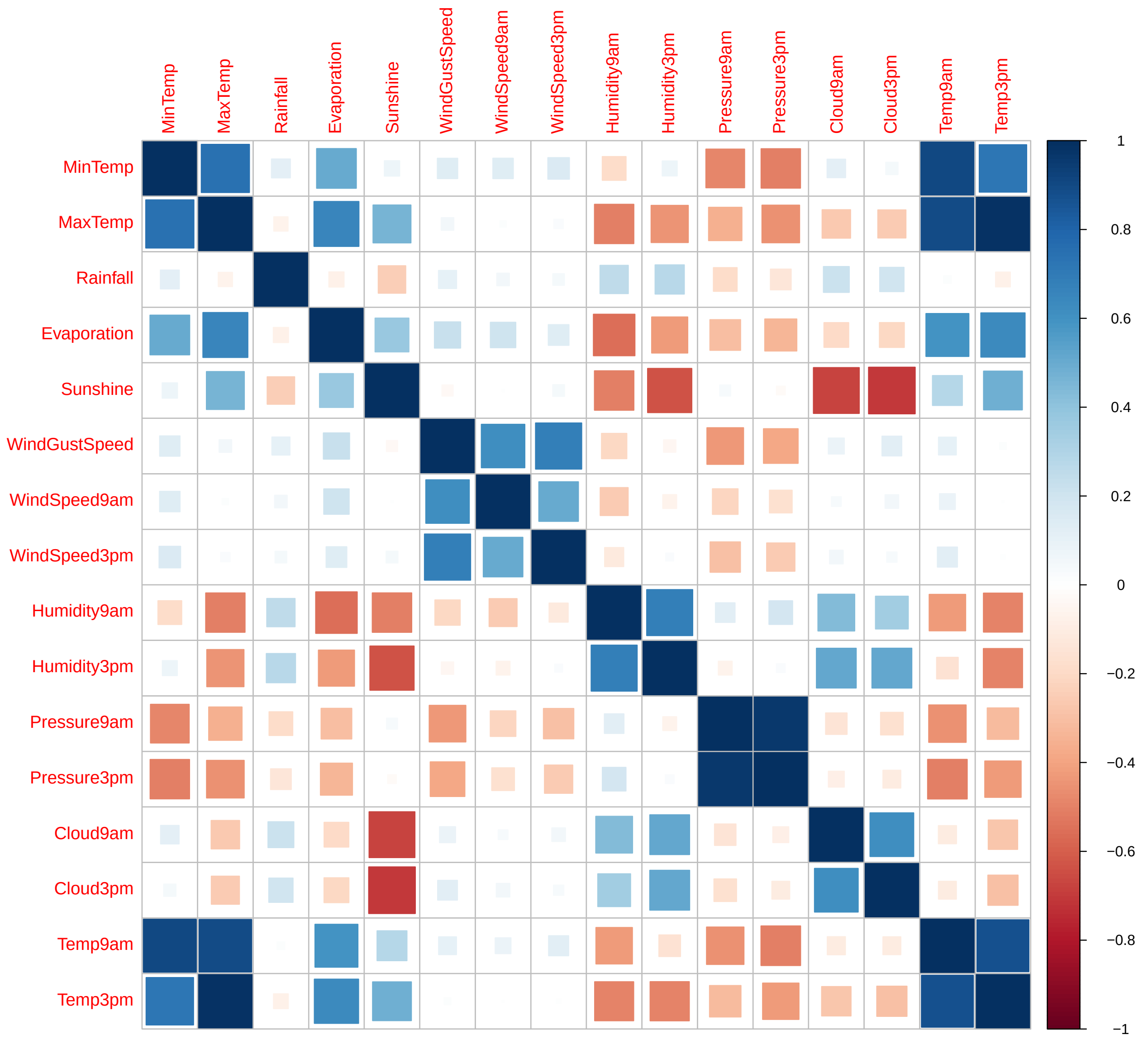
Remove NAs

```
train = na.omit(train)
test = na.omit(test)
weather = na.omit(weather)
RainTom.test <- test$RainTomorrow
```

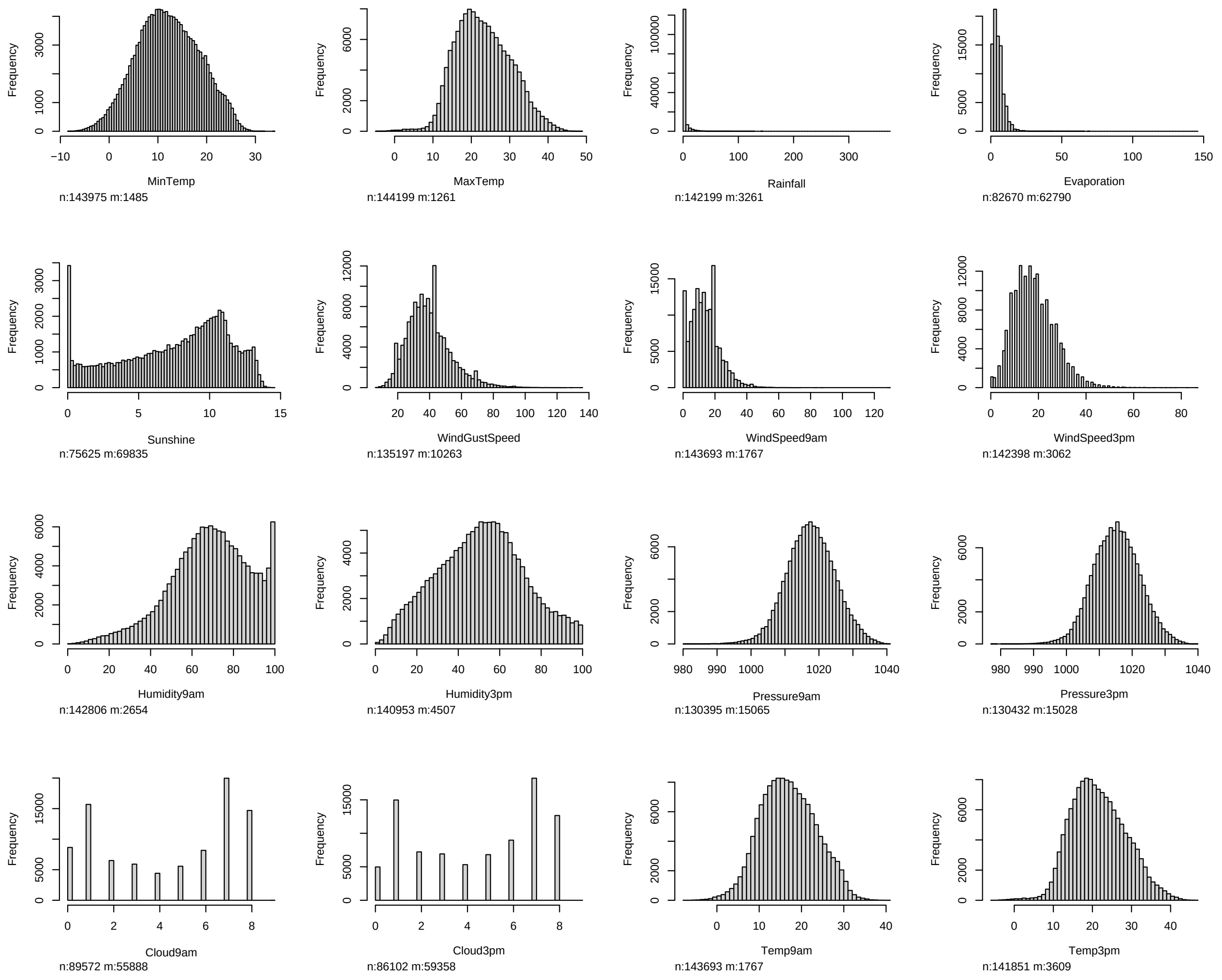
Exploratory Data Analysis

```
st(weather) # Summary Statistics
```

```
corrplot(cor(weather[, c(-17, -18)]), method = "square") # Correlation plot
```



```r
# Adjust margins and create histograms of predictor variables
par(mar = c(5, 4, 4, 2) + 0.1)
hist.data.frame(weather_plotting) # Histogram of the Predictor Variables
```



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")
preds= prediction(glm.probs, RainTom.test)
prf = performance(preds, measure = "tpr", x.measure = "fpr")

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)
```

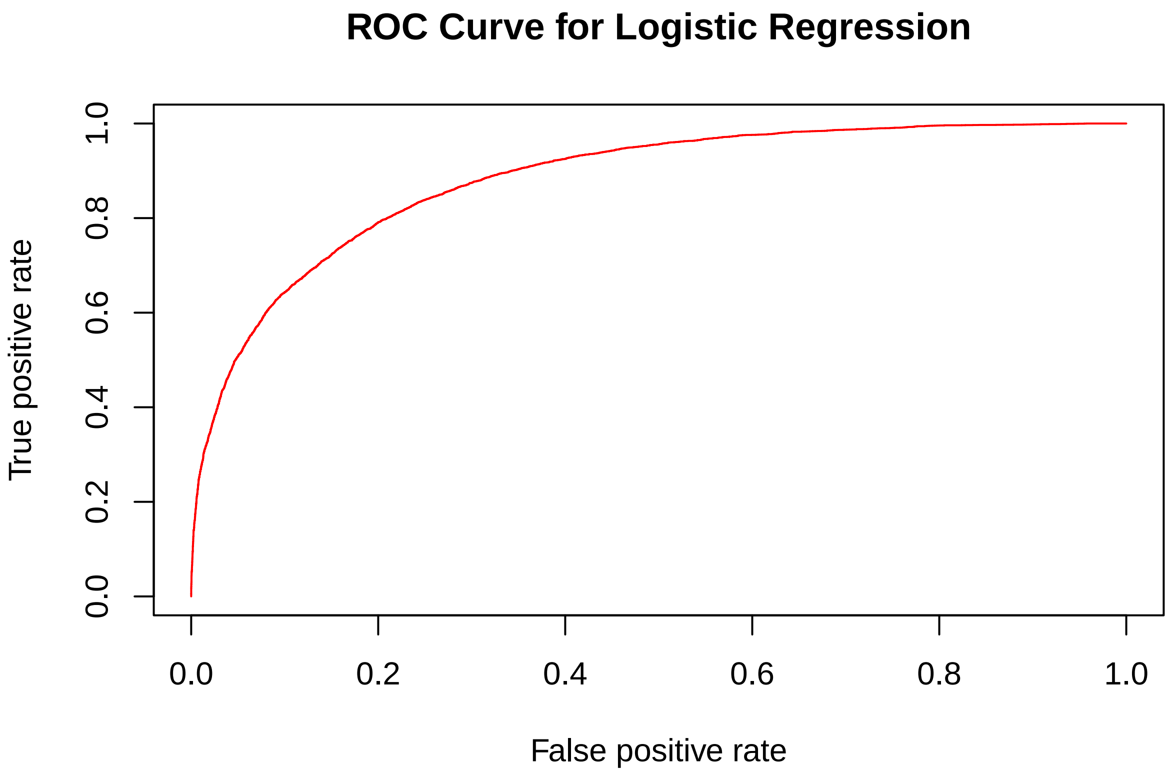
```
## [1] 0.854926
```

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

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

GLM plot

```
plot(prf, col = 'red', main = 'ROC Curve for Logistic Regression')
```

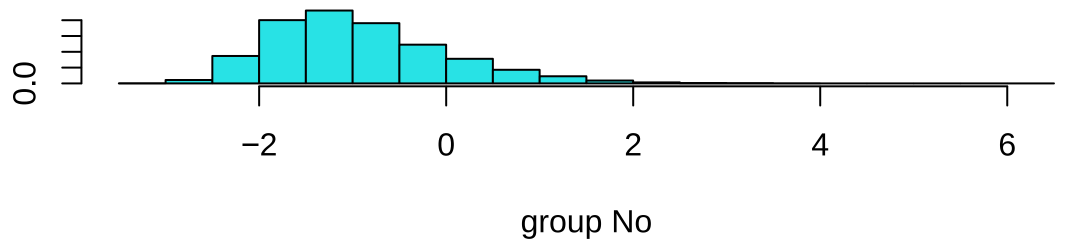


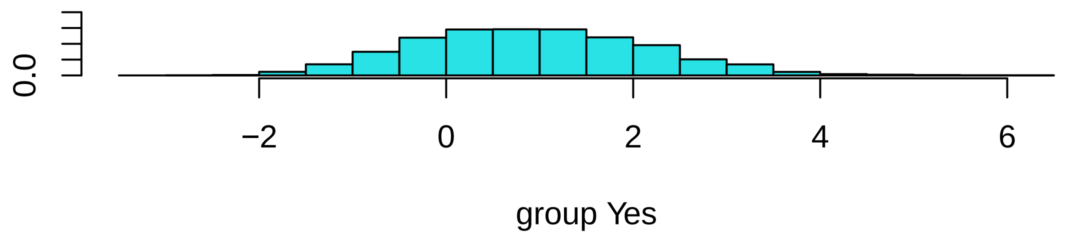
LDA Model

```
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, ylab = "Frequency")
```





```r
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
```

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

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

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

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

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

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

```r
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
```

```r
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
```

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

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

QDA Model

```r
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
```

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

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

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

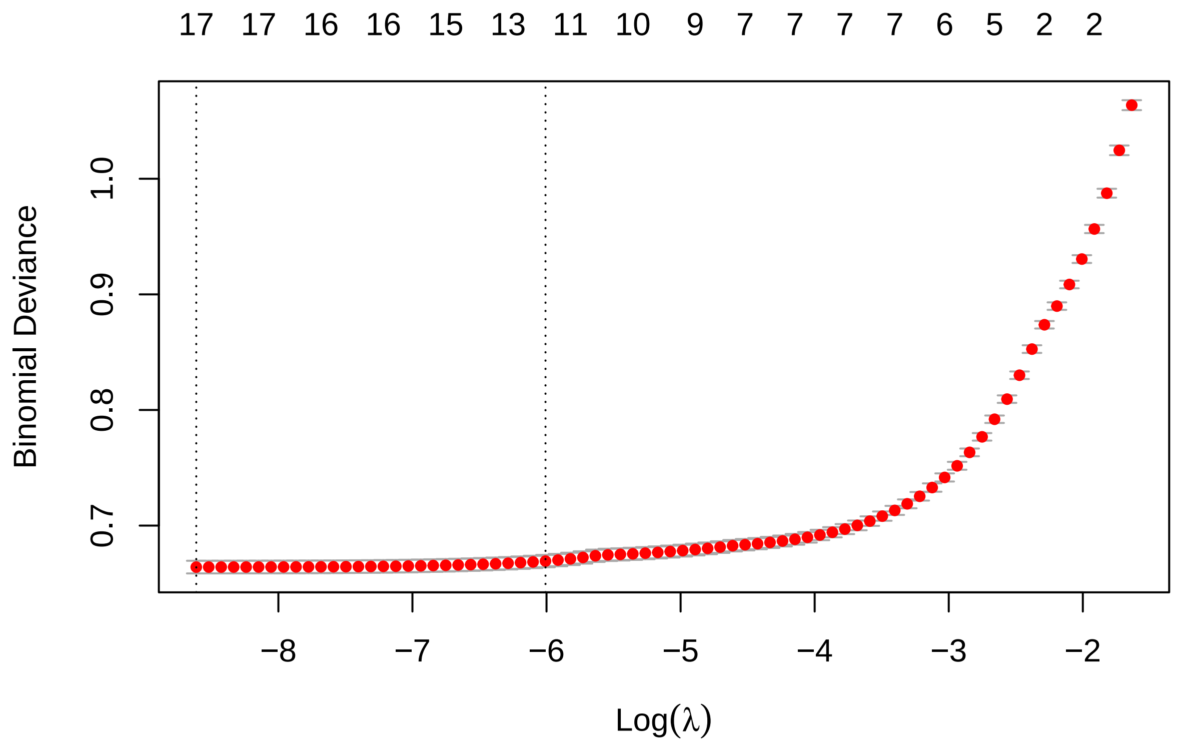
```
## [1] 0.8372507
```

```r
# 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 Model

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



```r
lasso.pred <- predict(lasso.fit, s = "lambda.min", newx = x[test_rows, ], type = "class")
lasso.pred <- ifelse(lasso.pred == "1", "Yes", "No")
table(lasso.pred, RainTom.test)
```

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

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

```
## [1] 0.8550774
```

```
lasso_coefficients <- predict(lasso.fit, type = "coefficients", s = "lambda.min")
lasso_coefficients[lasso_coefficients != 0]
```

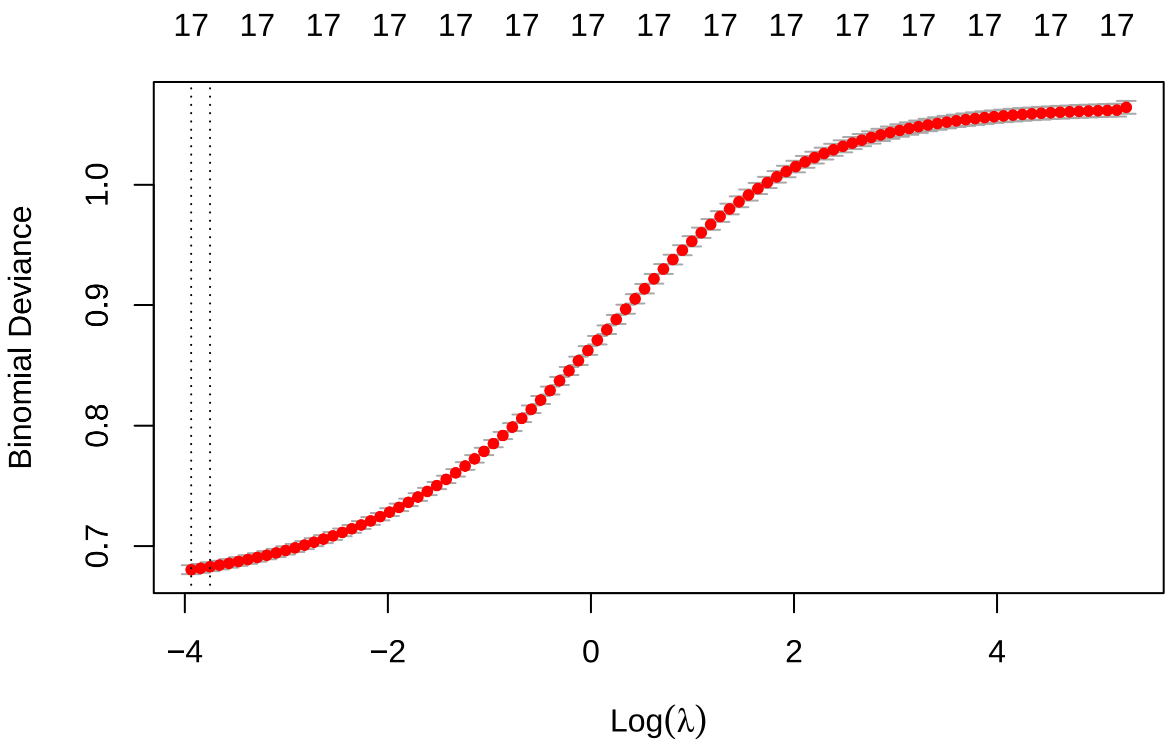
```
##  [1] 56.672900914 -0.039524262  0.002861825  0.012524012 -0.002833204
##  [6] -0.139631993  0.059845053 -0.010288875 -0.027363712  0.001074917
## [11]  0.057336990  0.141957640 -0.204532156 -0.014934135  0.125236085
## [16]  0.037511549  0.005217262  0.418399522
```

```
length(lasso_coefficients[lasso_coefficients != 0])
```

```
## [1] 18
```

Ridge Model

```
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
```

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

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

```r
ridge_coefficients <- predict(ridge.fit, type = "coefficients", s = "lambda.min")
ridge_coefficients[ridge_coefficients != 0]
```

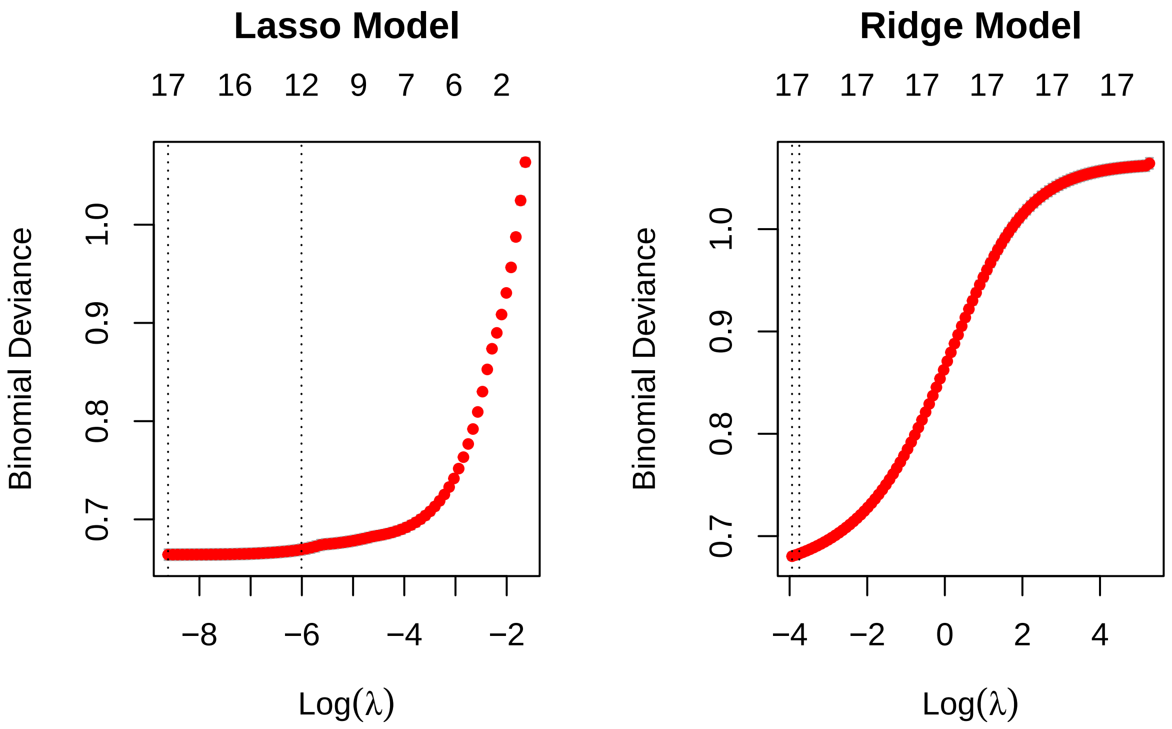
```
##  [1] 57.1883694152 -0.0004860718  0.0049484487  0.0129704448 -0.0199356224
##  [6] -0.1231279365  0.0383400223 -0.0060877086 -0.0125301733  0.0050073728
## [11]  0.0363669856 -0.0047604579 -0.0565809903  0.0033221501  0.1264372069
## [16]  0.0174424096 -0.0075681767  0.2988933497
```

```r
length(ridge_coefficients[ridge_coefficients != 0])
```

```
## [1] 18
```

Plots for Lasso and Ridge Model

```r
par(mfrow = c(1, 2), mar = c(5, 4, 6, 2) + 0.1)
plot(lasso.fit, main = "Lasso Model")
plot(ridge.fit, main = "Ridge Model")
```



Random Forest

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

```
##         RainTom.test
## rf.pred    No   Yes
##     No  19808  2766
##     Yes   983  2864
```

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

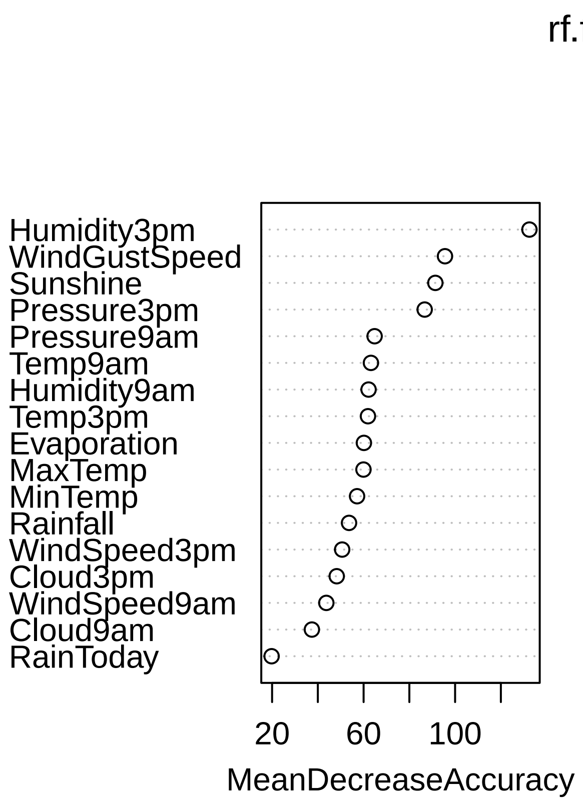
```
## [1] 0.8581053
```

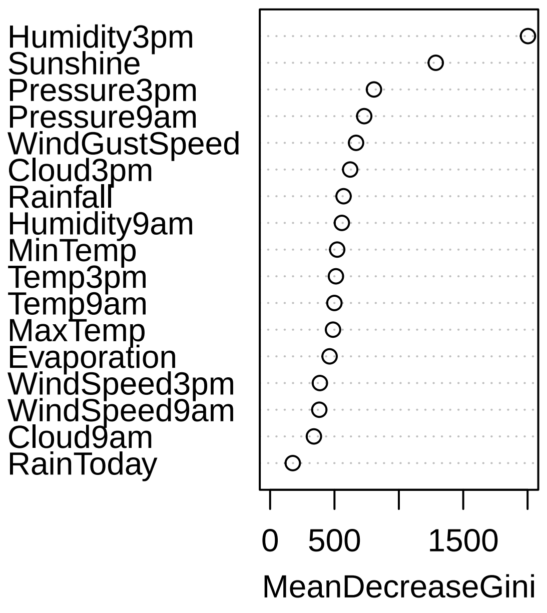
Random Forest Importance

```r
importance(rf.fit)
```

```
##                     No         Yes MeanDecreaseAccuracy MeanDecreaseGini
## MinTemp       52.30700   9.2441145             57.15189         521.0837
## MaxTemp       57.32062   0.4328667             59.95045         488.4701
## Rainfall      33.41010  43.0130046             53.63717         570.6358
## Evaporation   57.92732   1.5813540             60.13198         462.0444
## Sunshine      55.94480  78.3050253             91.36357        1286.5645
## WindGustSpeed 76.74449  49.9837049             95.57111         667.5121
## WindSpeed9am  48.44583  -2.3211877             43.71003         382.0238
## WindSpeed3pm  52.58007   2.6907981             50.62741         386.6336
## Humidity9am   55.95183  14.1262398             62.17725         557.1742
## Humidity3pm   79.70070 134.6170325            132.48281        2002.7659
## Pressure9am   58.39126  13.9229813             64.80674         730.8086
## Pressure3pm   74.84276  31.1631977             86.69644         806.5455
## Cloud9am      30.13842  20.2480638             37.39862         340.0611
## Cloud3pm      25.85419  40.8545023             48.25398         621.8783
## Temp9am       60.31213   0.2214884             63.21186         499.1424
## Temp3pm       56.94497  10.4248667             61.91070         511.1755
## RainToday     14.11826  17.4034119             19.78700         175.8386
```

```r
varImpPlot(rf.fit)
```


rf.fit
Humidity3pm
WindGustSpeed
Sunshine
Pressure3pm
Pressure9am
Temp9am
Humidity9am
Temp3pm
Evaporation
MaxTemp
MinTemp
Rainfall
WindSpeed3pm
Cloud3pm
WindSpeed9am
Cloud9am
RainToday
20
60
100
MeanDecreaseAccuracy


Humidity3pm
Sunshine
Pressure3pm
Pressure9am
WindGustSpeed
Cloud3pm
Rainfall
Humidity9am
MinTemp
Temp3pm
Temp9am
MaxTemp
Evaporation
WindSpeed3pm
WindSpeed9am
Cloud9am
RainToday
0
500
1500
MeanDecreaseGini