Classification

Bishal Neupane

9/17/2022

Source:

https://www.kaggle.com/code/abhpasha/logistic-regression-predicting-rain-in-australia

Importing data

```
df <- read.csv("weatherAUS.csv", header = TRUE)
```

```
head(df)
```

```
##        Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
## 1 12/1/2008   Albury    13.4    22.9      0.6          NA       NA           W
## 2 12/2/2008   Albury     7.4    25.1      0.0          NA       NA         WNW
## 3 12/3/2008   Albury    12.9    25.7      0.0          NA       NA         WSW
## 4 12/4/2008   Albury     9.2    28.0      0.0          NA       NA          NE
## 5 12/5/2008   Albury    17.5    32.3      1.0          NA       NA           W
## 6 12/6/2008   Albury    14.6    29.7      0.2          NA       NA         WNW
##   WindGustSpeed WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am
## 1            44          W        WNW           20           24          71
## 2            44        NNW        WSW            4           22          44
## 3            46          W        WSW           19           26          38
## 4            24         SE          E           11            9          45
## 5            41        ENE         NW            7           20          82
## 6            56          W          W           19           24          55
##   Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm
## 1          22      1007.7      1007.1        8       NA    16.9    21.8
## 2          25      1010.6      1007.8       NA       NA    17.2    24.3
## 3          30      1007.6      1008.7       NA        2    21.0    23.2
## 4          16      1017.6      1012.8       NA       NA    18.1    26.5
## 5          33      1010.8      1006.0        7        8    17.8    29.7
## 6          23      1009.2      1005.4       NA       NA    20.6    28.9
##   RainToday RainTomorrow
## 1        No           No
## 2        No           No
## 3        No           No
## 4        No           No
## 5        No           No
## 6        No           No
```

#There are alot of column so removing columns with non numeric values.

```
df$Date<- NULL
df$WindGustDir<-NULL
df$WindGustDir <-NULL
df$WindDir3pm <- NULL
df$WindDir3pm <-NULL
df$Location <-NULL
df$Sunshine <-NULL
df$RainToday <- NULL
df$WindDir9am <-NULL
df$Evaporation <-NULL
```

Structure of Data Frame

```
str(df)
```

```
## 'data.frame':    145460 obs. of  15 variables:
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ WindGustSpeed: int  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindSpeed9am : int  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : int  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : int  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : int  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : int  8 NA NA NA 7 NA 1 NA NA NA ...
##  $ Cloud3pm     : int  NA NA 2 NA 8 NA NA NA NA NA ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ...
##  $ RainTomorrow : chr  "No" "No" "No" "No" ...
```

Data Exploration

Names of Column

```
names(df)
```

```
##  [1] "MinTemp"       "MaxTemp"       "Rainfall"      "WindGustSpeed"
##  [5] "WindSpeed9am"  "WindSpeed3pm"  "Humidity9am"   "Humidity3pm"
##  [9] "Pressure9am"   "Pressure3pm"   "Cloud9am"      "Cloud3pm"
## [13] "Temp9am"       "Temp3pm"       "RainTomorrow"
```

Importing Package and using it to Change to factor

```
#install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
df <- mutate_if(df, is.character, as.factor)
```

Dimensions of df

```
dim(df)
```

```
## [1] 145460     15
```

```
str(df)
```

```
## 'data.frame':    145460 obs. of  15 variables:
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ WindGustSpeed: int  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindSpeed9am : int  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : int  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : int  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : int  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : int  8 NA NA NA 7 NA 1 NA NA NA ...
##  $ Cloud3pm     : int  NA NA 2 NA 8 NA NA NA NA NA ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
```

Statistics Summary of Each column

```
summary(df)
```

```
##     MinTemp          MaxTemp          Rainfall       WindGustSpeed
##  Min.   :-8.50   Min.   :-4.80   Min.   :  0.000   Min.   :  6.00
##  1st Qu.: 7.60   1st Qu.:17.90   1st Qu.:  0.000   1st Qu.: 31.00
##  Median :12.00   Median :22.60   Median :  0.000   Median : 39.00
##  Mean   :12.19   Mean   :23.22   Mean   :  2.361   Mean   : 40.03
##  3rd Qu.:16.90   3rd Qu.:28.20   3rd Qu.:  0.800   3rd Qu.: 48.00
##  Max.   :33.90   Max.   :48.10   Max.   :371.000   Max.   :135.00
##  NA's   :1485    NA's   :1261    NA's   :3261      NA's   :10263
##   WindSpeed9am     WindSpeed3pm    Humidity9am      Humidity3pm
##  Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  7.00   1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00
##  Median : 13.00   Median :19.00   Median : 70.00   Median : 52.00
##  Mean   : 14.04   Mean   :18.66   Mean   : 68.88   Mean   : 51.54
##  3rd Qu.: 19.00   3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 66.00
##  Max.   :130.00   Max.   :87.00   Max.   :100.00   Max.   :100.00
##  NA's   :1767     NA's   :3062    NA's   :2654     NA's   :4507
##   Pressure9am      Pressure3pm        Cloud9am        Cloud3pm
##  Min.   : 980.5   Min.   : 977.1   Min.   :0.00    Min.   :0.00
##  1st Qu.:1012.9   1st Qu.:1010.4   1st Qu.:1.00    1st Qu.:2.00
##  Median :1017.6   Median :1015.2   Median :5.00    Median :5.00
##  Mean   :1017.6   Mean   :1015.3   Mean   :4.45    Mean   :4.51
##  3rd Qu.:1022.4   3rd Qu.:1020.0   3rd Qu.:7.00    3rd Qu.:7.00
##  Max.   :1041.0   Max.   :1039.6   Max.   :9.00    Max.   :9.00
##  NA's   :15065    NA's   :15028    NA's   :55888   NA's   :59358
##     Temp9am          Temp3pm      RainTomorrow
##  Min.   :-7.20   Min.   :-5.40   No  :110316
##  1st Qu.:12.30   1st Qu.:16.60   Yes : 31877
##  Median :16.70   Median :21.10   NA's:  3267
##  Mean   :16.99   Mean   :21.68
##  3rd Qu.:21.60   3rd Qu.:26.40
##  Max.   :40.20   Max.   :46.70
##  NA's   :1767    NA's   :3609
```

Exploring Missing values

```
sum(is.na(df))
```

```
## [1] 182242
```

Removing the row with target value NA

```
df <- subset(df,RainTomorrow  != "NA")
```

Dimension after removing rows with NA as Rain Tomorrow

```
dim(df)
```

```
## [1] 142193     15
```

```
str(df)
```

```
## 'data.frame':    142193 obs. of  15 variables:
##  $ MinTemp      : num  13.4 7.4 12.9 9.2 17.5 14.6 14.3 7.7 9.7 13.1 ...
##  $ MaxTemp      : num  22.9 25.1 25.7 28 32.3 29.7 25 26.7 31.9 30.1 ...
##  $ Rainfall     : num  0.6 0 0 0 1 0.2 0 0 0 1.4 ...
##  $ WindGustSpeed: int  44 44 46 24 41 56 50 35 80 28 ...
##  $ WindSpeed9am : int  20 4 19 11 7 19 20 6 7 15 ...
##  $ WindSpeed3pm : int  24 22 26 9 20 24 24 17 28 11 ...
##  $ Humidity9am  : int  71 44 38 45 82 55 49 48 42 58 ...
##  $ Humidity3pm  : int  22 25 30 16 33 23 19 19 9 27 ...
##  $ Pressure9am  : num  1008 1011 1008 1018 1011 ...
##  $ Pressure3pm  : num  1007 1008 1009 1013 1006 ...
##  $ Cloud9am     : int  8 NA NA NA 7 NA 1 NA NA NA ...
##  $ Cloud3pm     : int  NA NA 2 NA 8 NA NA NA NA NA ...
##  $ Temp9am      : num  16.9 17.2 21 18.1 17.8 20.6 18.1 16.3 18.3 20.1 ...
##  $ Temp3pm      : num  21.8 24.3 23.2 26.5 29.7 28.9 24.6 25.5 30.2 28.2 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 2 1 ...
```

Replacing NA's with mean of a column

```
#install.packages('tidyr')
for(i in 1:ncol(df)){
  df[is.na(df[,i]), i] <- mean(df[,i], na.rm = TRUE)
}
```

```
## Warning in mean.default(df[, i], na.rm = TRUE): argument is not numeric or
## logical: returning NA
```

Summary after replacing NA's with mean

```
summary(df)
```

```
##     MinTemp          MaxTemp          Rainfall       WindGustSpeed
##  Min.   :-8.50   Min.   :-4.80   Min.   :  0.00   Min.   :  6.00
##  1st Qu.: 7.60   1st Qu.:17.90   1st Qu.:  0.00   1st Qu.: 31.00
##  Median :12.00   Median :22.70   Median :  0.00   Median : 39.00
##  Mean   :12.19   Mean   :23.23   Mean   :  2.35   Mean   : 39.98
##  3rd Qu.:16.80   3rd Qu.:28.20   3rd Qu.:  0.80   3rd Qu.: 46.00
##  Max.   :33.90   Max.   :48.10   Max.   :371.00   Max.   :135.00
##   WindSpeed9am  WindSpeed3pm    Humidity9am      Humidity3pm
##  Min.   :  0   Min.   : 0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  7   1st Qu.:13.00   1st Qu.: 57.00   1st Qu.: 37.00
##  Median : 13   Median :18.64   Median : 70.00   Median : 51.48
##  Mean   : 14   Mean   :18.64   Mean   : 68.84   Mean   : 51.48
##  3rd Qu.: 19   3rd Qu.:24.00   3rd Qu.: 83.00   3rd Qu.: 65.00
##  Max.   :130   Max.   :87.00   Max.   :100.00   Max.   :100.00
##   Pressure9am      Pressure3pm        Cloud9am        Cloud3pm
##  Min.   : 980.5   Min.   : 977.1   Min.   :0.000   Min.   :0.000
```

```
##  1st Qu.:1013.5   1st Qu.:1011.0   1st Qu.:3.000   1st Qu.:4.000
##  Median :1017.7   Median :1015.3   Median :4.437   Median :4.503
##  Mean   :1017.7   Mean   :1015.3   Mean   :4.437   Mean   :4.503
##  3rd Qu.:1021.8   3rd Qu.:1019.4   3rd Qu.:6.000   3rd Qu.:6.000
##  Max.   :1041.0   Max.   :1039.6   Max.   :9.000   Max.   :9.000
##     Temp9am         Temp3pm      RainTomorrow
##  Min.   :-7.20   Min.   :-5.40   No :110316
##  1st Qu.:12.30   1st Qu.:16.70   Yes: 31877
##  Median :16.80   Median :21.30
##  Mean   :16.99   Mean   :21.69
##  3rd Qu.:21.50   3rd Qu.:26.30
##  Max.   :40.20   Max.   :46.70
```

Data Visualization

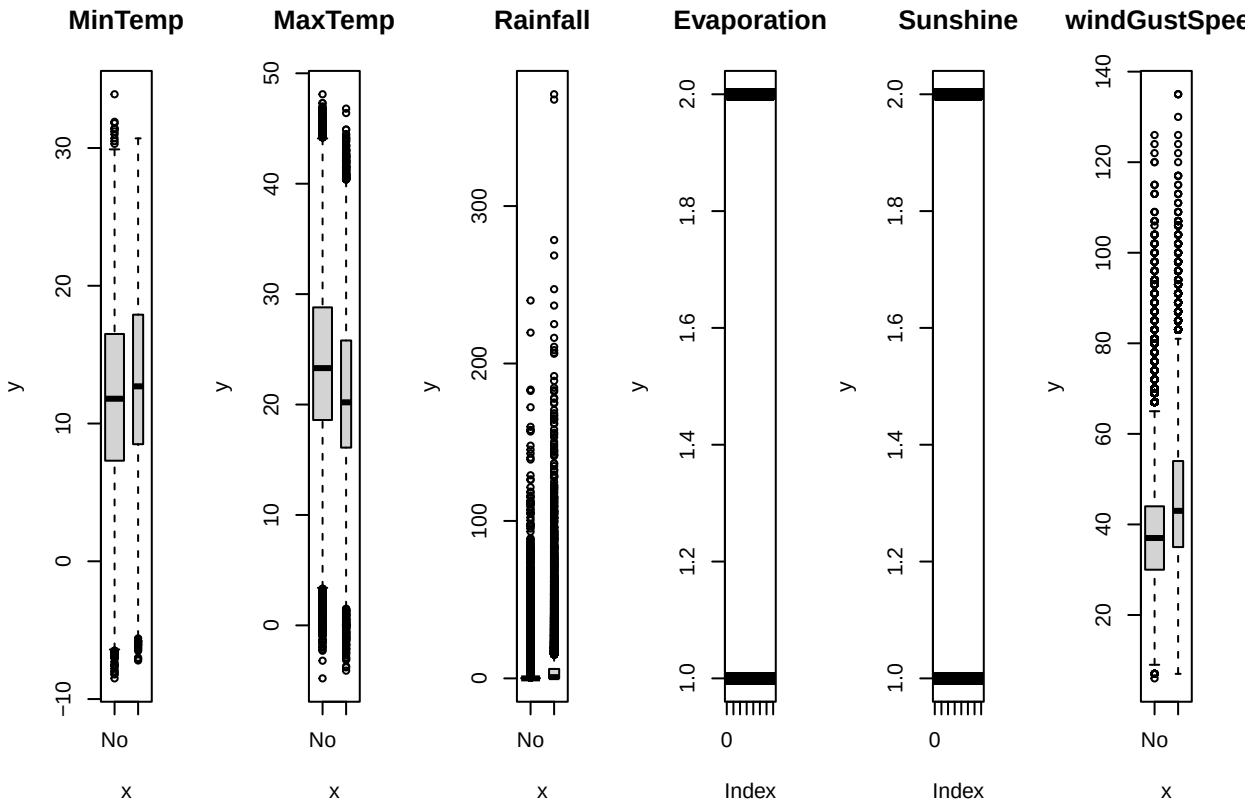
```
par(mfrow=c(1,6))
plot(df$RainTomorrow, df$MinTemp, data=df, main="MinTemp",
varwidth=TRUE)
plot(df$RainTomorrow, df$MaxTemp, data=df, main="MaxTemp", varwidth=TRUE)
plot(df$RainTomorrow, df$Rainfall, data=df, main="Rainfall", varwidth=TRUE)
plot(df$RainTomorrow, df$Evaporation, data=df, main="Evaporation", varwidth=TRUE)
```

```
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.window(...): "varwidth" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "varwidth" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "varwidth" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "varwidth" is not a
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in box(...): "varwidth" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter

## Warning in title(...): "varwidth" is not a graphical parameter
```

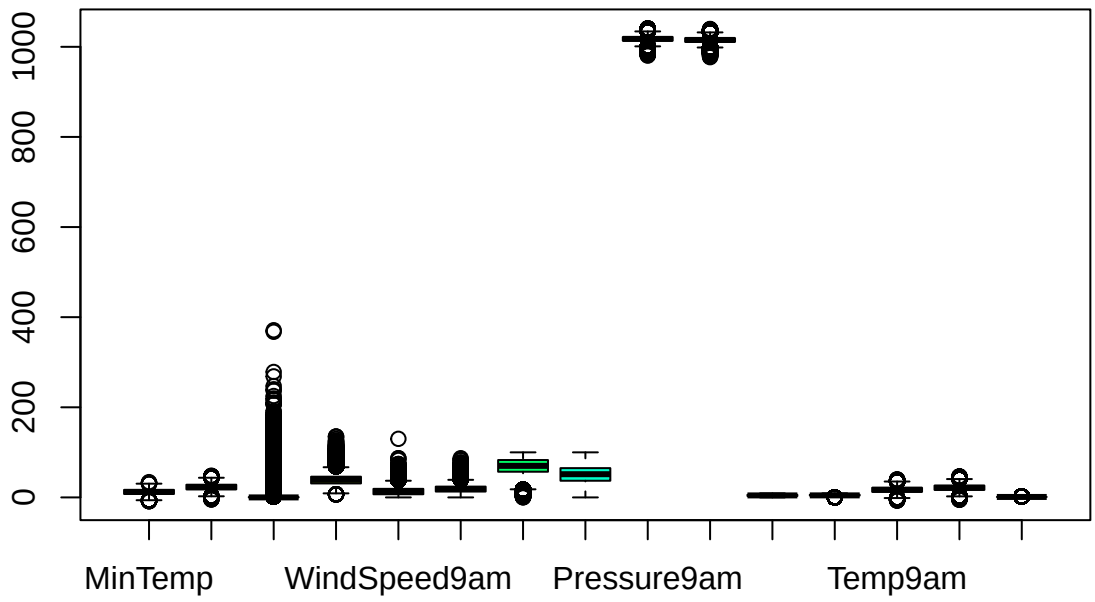
```
plot(df$RainTomorrow, df$Sunshine, data=df, main="Sunshine", varwidth=TRUE)
```

```
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.window(...): "varwidth" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "varwidth" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "varwidth" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "varwidth" is not a
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in box(...): "varwidth" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter

## Warning in title(...): "varwidth" is not a graphical parameter
```

```
plot(df$RainTomorrow, df$WindGustSpeed, data=df, main="windGustSpeed",
varwidth=TRUE)
```




```
boxplot(df, col = rainbow(ncol(df)))
```



Model Building (Logistic Regression)

Building Model and getting summary for all of the 15 predictors

```
set.seed(1234)
i <- sample(1:nrow(df), 0.80*nrow(df), replace=FALSE)
train <-df[i,]
test <- df[-i,]
glm1 <- glm(RainTomorrow~., data=train, family=binomial)
summary(glm1)
```

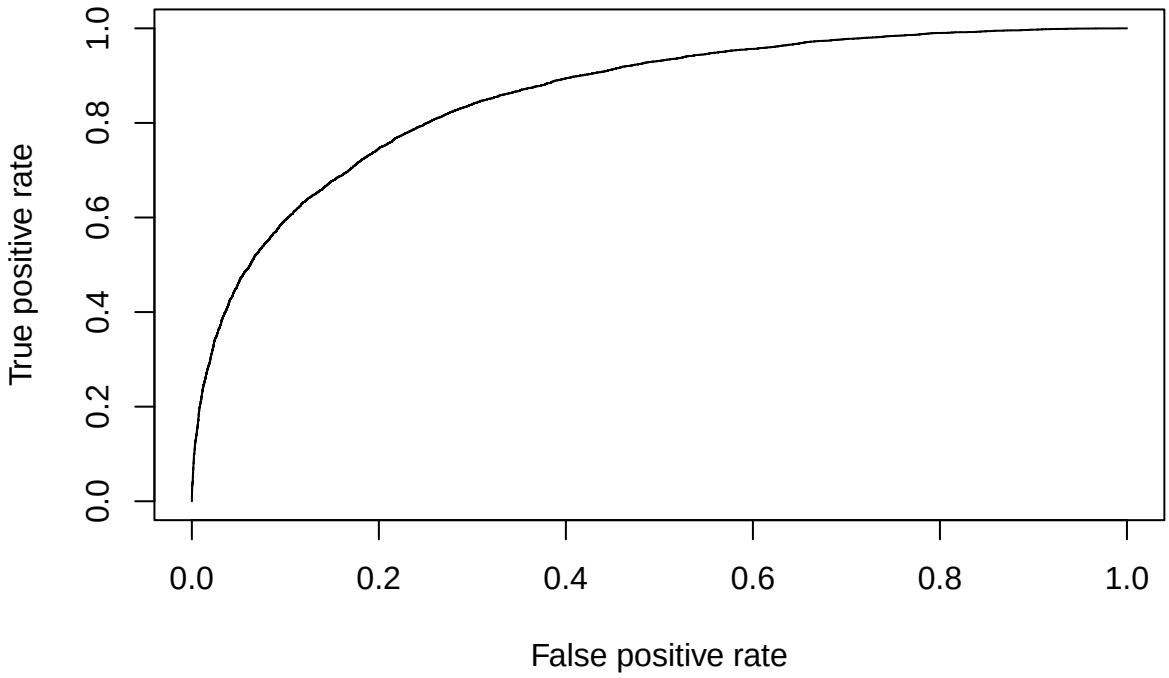
```
##
## Call:
## glm(formula = RainTomorrow ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2931  -0.5709  -0.3325  -0.1304   3.2242
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   56.3125905  1.6039308  35.109  < 2e-16 ***
## MinTemp        0.0193635  0.0042659   4.539 5.65e-06 ***
```

```
## MaxTemp       -0.0461163  0.0052193  -8.836  < 2e-16 ***
## Rainfall       0.0227846  0.0012042  18.921  < 2e-16 ***
## WindGustSpeed  0.0544559  0.0009793  55.607  < 2e-16 ***
## WindSpeed9am  -0.0103997  0.0013181  -7.890 3.03e-15 ***
## WindSpeed3pm  -0.0260557  0.0013288 -19.608  < 2e-16 ***
## Humidity9am    0.0069509  0.0008930   7.784 7.05e-15 ***
## Humidity3pm    0.0537375  0.0009197  58.429  < 2e-16 ***
## Pressure9am    0.1069824  0.0049713  21.520  < 2e-16 ***
## Pressure3pm   -0.1699806  0.0050141 -33.900  < 2e-16 ***
## Cloud9am       0.0417745  0.0051255   8.150 3.63e-16 ***
## Cloud3pm       0.1798765  0.0054734  32.864  < 2e-16 ***
## Temp9am        0.0120170  0.0060172   1.997  0.04581 *
## Temp3pm        0.0171757  0.0055089   3.118  0.00182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 120923  on 113753  degrees of freedom
## Residual deviance:  83519  on 113739  degrees of freedom
## AIC: 83549
##
## Number of Fisher Scoring iterations: 5
```

Prediction and result summary

Predicting Test Set and plotting ROC

```
#install.packages("ROCR")
library(ROCR)
p <- predict(glm1, newdata=test, type="response")
pr <- prediction(p, test$RainTomorrow)
# TPR = sensitivity, FPR=specificity
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
# compute AUC
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
print(auc)
```

```
## [1] 0.8566688
```

Explaination of Summary

we are using glm() generalized linear function.For the logistic regression, the residuals are deviancce residuals.The deviance residual is a mathematical transformation of loss function.The null deviance measures the lack of fit of the model with only intercept while residual deviance measures lack of fit of the entire model. In our case residual deviance is lower than the Null deviance. The Fisher scoring algorithm is a modified form of Newton's method of solving a maximum likelihood problem. In logistic regression, the coefficient quantifies the difference in the log odds of the target variable rather than measuring difference in target variable. ROC curves goes up from 0 to 1 which means that the model is performing pretty well. AUC value is also 0.86

Dimension of Test Case

```
dim(test)
```

```
## [1] 28439    15
```

Predicting on Test data and print accuracy

```
probs <- predict(glm1, newdata=test, type="response")

pred <- ifelse(probs>0.5, 2, 1)
acc1 <- mean(pred==as.integer(test$RainTomorrow))
print(paste("glm1 accuracy = ", acc1))
```

```
## [1] "glm1 accuracy =  0.840219416997785"
```

Accuracy Explaination:

The accuracy of the model is about 84 percent.

Model Building (Naive Bayes)

Installing package and using it to train

```
#install.packages("e1071")
library(e1071)
nb1 <- naiveBayes(RainTomorrow~., data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##        No       Yes
## 0.7763331 0.2236669
##
## Conditional probabilities:
##      MinTemp
## Y         [,1]     [,2]
##   No  11.90616 6.396925
##   Yes 13.21459 6.319109
##
##      MaxTemp
## Y         [,1]     [,2]
##   No  23.84033 7.062661
##   Yes 21.16600 6.942200
##
##      Rainfall
## Y         [,1]      [,2]
##   No  1.284295  5.377816
##   Yes 6.032321 14.107786
```

```
##
##      WindGustSpeed
## Y         [,1]     [,2]
##   No  38.38022 12.02649
##   Yes 45.54337 15.25268
##
##      WindSpeed9am
## Y         [,1]     [,2]
##   No  13.57903 8.598428
##   Yes 15.50484 9.594962
##
##      WindSpeed3pm
## Y         [,1]     [,2]
##   No  18.21943 8.393907
##   Yes 20.05846 9.713400
##
##      Humidity9am
## Y         [,1]     [,2]
##   No  66.24121 18.92812
##   Yes 77.74624 16.04076
##
##      Humidity3pm
## Y         [,1]     [,2]
##   No  46.61060 18.29663
##   Yes 68.24655 18.97738
##
##      Pressure9am
## Y         [,1]     [,2]
##   No  1018.501 6.429970
##   Yes 1014.688 7.009321
##
##      Pressure3pm
## Y         [,1]     [,2]
##   No  1016.027 6.380775
##   Yes 1012.558 7.033430
##
##      Cloud9am
## Y         [,1]     [,2]
##   No  4.129147 2.275001
##   Yes 5.511903 1.951483
##
##      Cloud3pm
## Y         [,1]     [,2]
##   No  4.160614 2.067661
##   Yes 5.685443 1.792424
##
##      Temp9am
## Y         [,1]     [,2]
##   No  17.08196 6.512454
##   Yes 16.71863 6.374956
##
##      Temp3pm
## Y         [,1]     [,2]
##   No  22.39961 6.800710
```

```
##   Yes 19.29338 6.611687
```

Explaination of Result:

The prior and likelihood is calculated from the training set.The prior is shown in the form of A-priori which is 0.77 and 0.22 in our case. Likelihood is shown as the conditional probability. Each row sums upto one and each shows the likelihood of occuring each events.

```
p2_raw <- predict(nb1, newdata=test, type="raw")
head(p2_raw, n=2)
```

```
##             No         Yes
## [1,] 0.9996001 0.000399853
## [2,] 0.9538949 0.046105099
```

Explaination on test

The prediction of test for two rows of test data set is shown above which is 99 percent and 95 percent no.

Comparison of Models:

The result of both models seems to be pretty similar. The ROC of logistic regression shows that the model is pretty good. The accuracy was also almost similar.I have used all of the 15 features for both logistic regression and naive bayes.

Strength of Logistic Regressions:

1) Logistic regression is easier to implement, interpret and very efficient to train
2) It can easily extend to multiple classes.
3) It provides a measure of how appropriate is a predictor.

Weakness of Logistic Regression:

1) If number of rows is less than the number of attributes then it will lead to over fitting.
2) It can only be used to predict descrete function
3) Non linear problems cannot be solved with logistic regression.

Strength of Naive Bayes Classifier:

1) It is simple to implement.
2) It is very fast because probabilities can be directly calculated without loops.
3) It works well with both continuous and discrete data.

Weakness of Naive Bayes Classifier:

1) This algorithm assumes that all features are independent which rarely happens in real life.
2) It would create problem when the categorical variable is only seen in test dataset. It will assign the zero probability which can create problem to the result.

Explaination of benefits and drawbacks of each Classification metrics used:

1) Accuracy:

Accuracy is the ratio of correctly classified to the total number of rows.

Advantages of Accuracy

1) Easy to use, understand and relate.
2) Give the proper effectiveness of model if data is balanced.

Drawbacks of Accuracy

1) Not as interreptable as confusion matrix
2) It doesn't take wrong prediction into consideration

2) Confusion Matrix:

Advantages of Confusion Matrix:

1) It specifies for which label model is confused.
2) It shows the correct and incorrect prediction.

Disadvantages:

1) Checking for over and under fitting is difficult.
2) It doesn't give a class probabilities.

3) ROC curves and AUC:

Advantages:

1) It shows the graphical representation of accuracy of test
2) It allows more complex and more exact measure of accuracy.

Disadvantages:

1) Actual decision threshold is not displayed.
2) It is not easily interreptable from business prospective.

5) MCC:

Advantages:

1) It accounts for difference in class distribution.