Classification using Logistic Regression, KNN, Decision Tree

Bishal Neupane, Saugat Gyawali, Spencer Gray, Michael Stinnett

10/08/2022

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
```

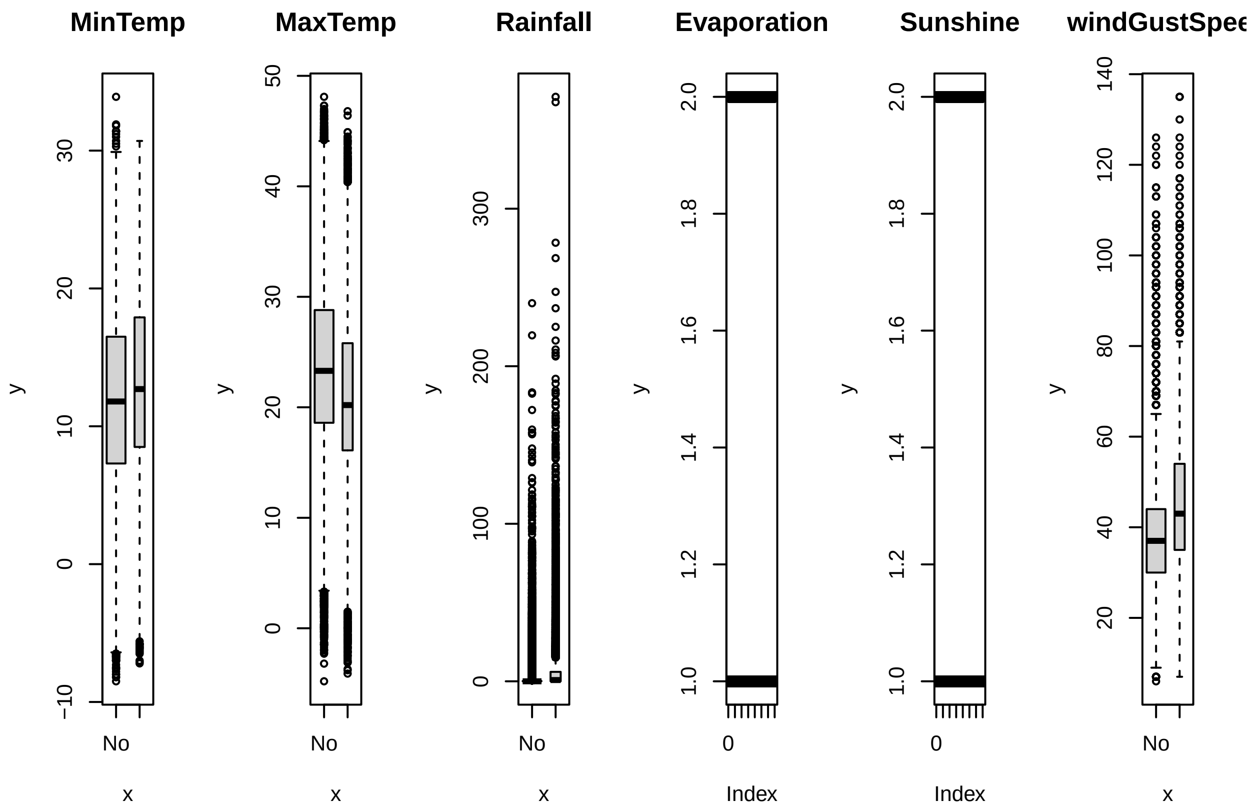
```
## Warning in box(...): "data" is not a graphical parameter
```

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

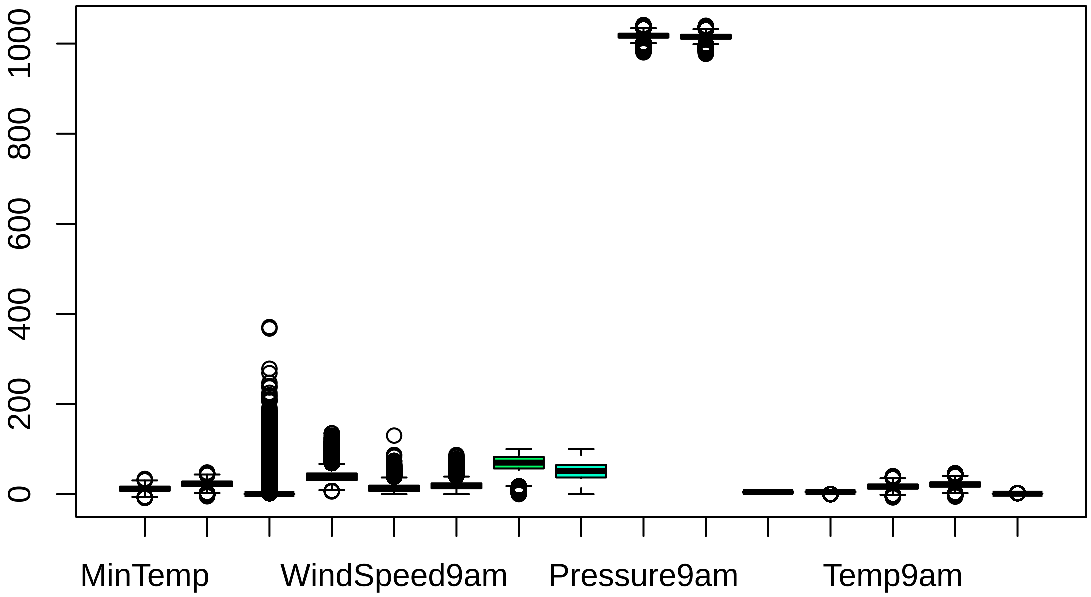
```
## Warning in title(...): "data" is not a graphical parameter
```

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

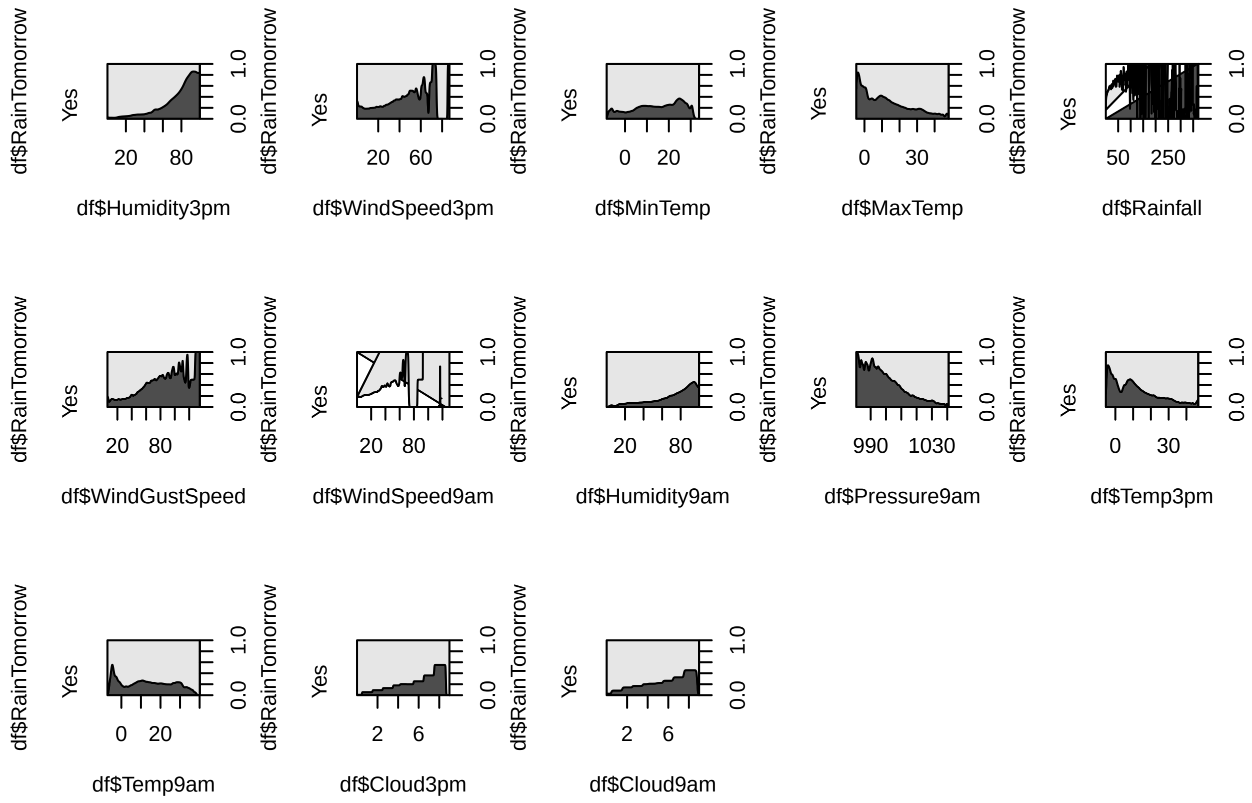
```
plot(df$RainTomorrow, df$WindGustSpeed, data=df, main="windGustSpeed",
varwidth=TRUE)
```



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



```
par(mfrow=c(3,5))
cdplot(df$RainTomorrow~df$Humidity3pm)
cdplot(df$RainTomorrow~df$WindSpeed3pm)
cdplot(df$RainTomorrow~df$MinTemp)
cdplot(df$RainTomorrow~df$MaxTemp)
cdplot(df$RainTomorrow~df$Rainfall)
cdplot(df$RainTomorrow~df$WindGustSpeed)
cdplot(df$RainTomorrow~df$WindSpeed9am)
cdplot(df$RainTomorrow~df$Humidity9am)
cdplot(df$RainTomorrow~df$Pressure9am)
cdplot(df$RainTomorrow~df$Temp3pm)
cdplot(df$RainTomorrow~df$Temp9am)
cdplot(df$RainTomorrow~df$Cloud3pm)
cdplot(df$RainTomorrow~df$Cloud9am)
```



Model Building (Logistic Regression)

Building Model and getting summary for all of the 15 predictors

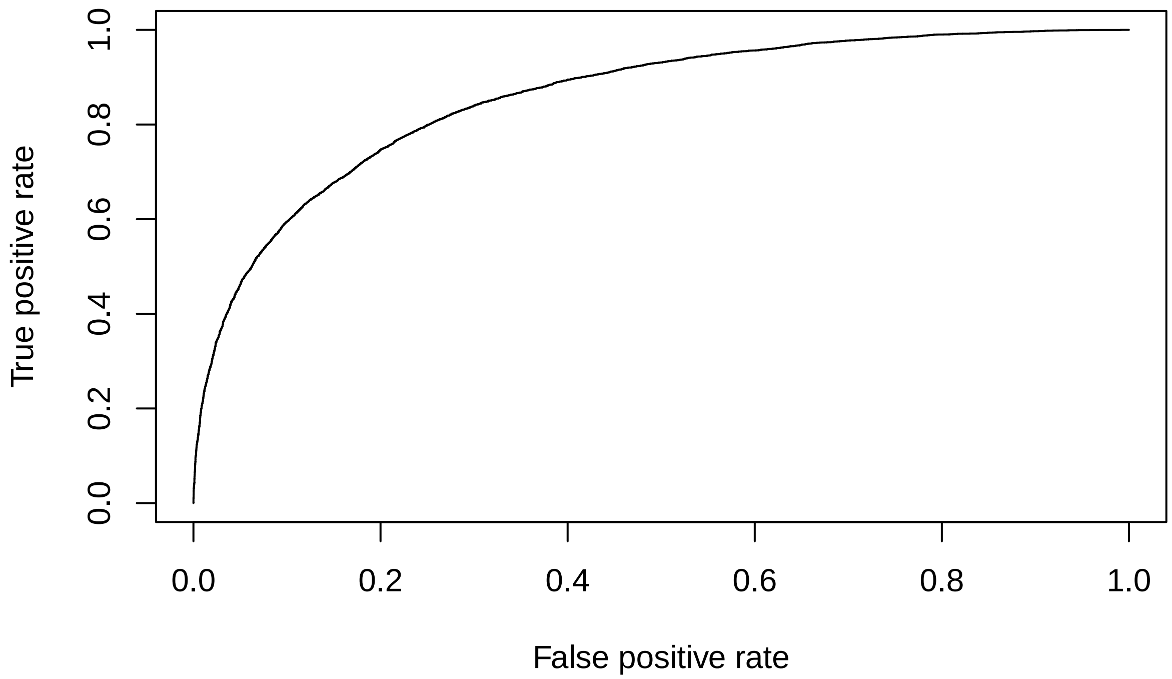
```
set.seed(1234)
i <- sample(1:nrow(df), 0.80*nrow(df), replace=FALSE)
train <-df[i,]
test <- df[-i,]
glm1 <- glm(RainTomorrow~., data=train, family=binomial)
summary(glm1)
```

```
##
## 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
```

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"
```

```
table(pred,as.integer(test$RainTomorrow))
```

```
##
## pred     1     2
##    1 20850  3389
##    2  1155  3045
```

Accuracy Explaination:

The accuracy of the model is about 84 percent.

```
str(test)
```

```
## 'data.frame':    28439 obs. of  15 variables:
##  $ MinTemp      : num  9.2 9.7 9.8 9.8 11.5 19.7 12.3 16.1 13.9 18.6 ...
##  $ MaxTemp      : num  28 31.9 27.7 25.6 29.3 27.2 34.6 38.9 36.6 39.9 ...
##  $ Rainfall     : num  0 0 2.35 0 0 ...
##  $ WindGustSpeed: num  24 80 50 26 24 46 37 57 39 61 ...
##  $ WindSpeed9am : num  11 7 14 17 9 ...
##  $ WindSpeed3pm : num  9 28 22 6 9 30 17 30 15 20 ...
##  $ Humidity9am  : num  45 42 50 45 56 49 41 34 39 36 ...
##  $ Humidity3pm  : num  16 9 28 26 28 22 12 12 10 21 ...
##  $ Pressure9am  : num  1018 1009 1013 1019 1019 ...
##  $ Pressure3pm  : num  1013 1004 1010 1017 1015 ...
##  $ Cloud9am     : num  4.44 4.44 0 4.44 4.44 ...
##  $ Cloud3pm     : num  4.5 4.5 4.5 4.5 4.5 ...
##  $ Temp9am      : num  18.1 18.3 17.3 15.8 19.1 21.6 20.7 25.2 22 26.8 ...
##  $ Temp3pm      : num  26.5 30.2 26.2 23.2 27.3 26.1 33.9 38.4 34.4 37.7 ...
##  $ RainTomorrow : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 1 1 1 ...
```

```
levels(test$RainTomorrow) <- list("1" = "No", "2" = "Yes")
str(test)
```

```
## 'data.frame':    28439 obs. of  15 variables:
##  $ MinTemp      : num  9.2 9.7 9.8 9.8 11.5 19.7 12.3 16.1 13.9 18.6 ...
##  $ MaxTemp      : num  28 31.9 27.7 25.6 29.3 27.2 34.6 38.9 36.6 39.9 ...
##  $ Rainfall     : num  0 0 2.35 0 0 ...
##  $ WindGustSpeed: num  24 80 50 26 24 46 37 57 39 61 ...
##  $ WindSpeed9am : num  11 7 14 17 9 ...
##  $ WindSpeed3pm : num  9 28 22 6 9 30 17 30 15 20 ...
##  $ Humidity9am  : num  45 42 50 45 56 49 41 34 39 36 ...
##  $ Humidity3pm  : num  16 9 28 26 28 22 12 12 10 21 ...
##  $ Pressure9am  : num  1018 1009 1013 1019 1019 ...
##  $ Pressure3pm  : num  1013 1004 1010 1017 1015 ...
##  $ Cloud9am     : num  4.44 4.44 0 4.44 4.44 ...
##  $ Cloud3pm     : num  4.5 4.5 4.5 4.5 4.5 ...
##  $ Temp9am      : num  18.1 18.3 17.3 15.8 19.1 21.6 20.7 25.2 22 26.8 ...
##  $ Temp3pm      : num  26.5 30.2 26.2 23.2 27.3 26.1 33.9 38.4 34.4 37.7 ...
##  $ RainTomorrow : Factor w/ 2 levels "1","2": 1 2 1 1 1 2 1 1 1 1 ...
```

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
confusionMatrix(as.factor(pred),as.factor(test$RainTomorrow))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 20850  3389
##          2  1155  3045
##
##                Accuracy : 0.8402
##                  95% CI : (0.8359, 0.8445)
##     No Information Rate : 0.7738
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4797
##
##  Mcnemar’s Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9475
##             Specificity : 0.4733
##          Pos Pred Value : 0.8602
##          Neg Pred Value : 0.7250
##              Prevalence : 0.7738
##          Detection Rate : 0.7331
##    Detection Prevalence : 0.8523
##       Balanced Accuracy : 0.7104
##
##        ’Positive’ Class : 1
##
```

KNN

```
trainForKNN <- train
trainForKNN$RainTomorrow <- NULL
head(trainForKNN)
```

```
##        MinTemp MaxTemp Rainfall WindGustSpeed WindSpeed9am WindSpeed3pm
## 108993    18.0    20.2      0.0      39.98429           20           20
## 43174     14.2    23.5      4.2      67.00000           17           22
## 34388      8.3    19.1      0.0      74.00000            9           37
## 129197    11.1    26.3      0.2      41.00000           22           15
## 123341    18.3    26.7      0.0      30.00000           13           11
## 85349     18.0    30.3      0.0      20.00000            7            6
##        Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm
## 108993          67          66      1018.0      1017.1 8.000000 8.000000
## 43174           76          26      1007.0      1005.1 4.437189 2.000000
## 34388           73          39      1008.3      1005.9 1.000000 2.000000
## 129197          46          28      1025.3      1021.7 4.437189 4.503167
## 123341          68          58      1011.7      1011.3 6.000000 5.000000
## 85349           65          42      1015.6      1011.9 1.000000 3.000000
##        Temp9am Temp3pm
## 108993    19.2    19.9
## 43174     17.3    23.0
## 34388     12.1    16.4
## 129197    18.3    26.2
## 123341    23.5    25.2
## 85349     23.8    29.8
```

```
trainForKNNLabels <- train$RainTomorrow
testForKNN <- test
testForKNN$RainTomorrow <- NULL
testLabelForKNN <- test$RainTomorrow
head(testForKNN)
```

```
##    MinTemp MaxTemp Rainfall WindGustSpeed WindSpeed9am WindSpeed3pm Humidity9am
## 4      9.2    28.0 0.000000            24     11.00000            9          45
## 9      9.7    31.9 0.000000            80      7.00000           28          42
## 16     9.8    27.7 2.349974            50     14.00199           22          50
## 20     9.8    25.6 0.000000            26     17.00000            6          45
## 21    11.5    29.3 0.000000            24      9.00000            9          56
## 29    19.7    27.2 0.000000            46     19.00000           30          49
##    Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm
## 4           16      1017.6      1012.8 4.437189 4.503167    18.1    26.5
## 9            9      1008.9      1003.6 4.437189 4.503167    18.3    30.2
## 16          28      1013.4      1010.3 0.000000 4.503167    17.3    26.2
## 20          26      1019.2      1017.1 4.437189 4.503167    15.8    23.2
## 21          28      1019.3      1014.8 4.437189 4.503167    19.1    27.3
## 29          22      1004.8      1004.2 4.437189 4.503167    21.6    26.1
```

```
library(class)
knnPred <- knn(train = trainForKNN, test = testForKNN, cl=trainForKNNLabels, k=3)
```

```
levels(knnPred) <- list("1" = "No", "2" = "Yes")
str(knnPred)
```

```
##  Factor w/ 2 levels "1","2": 1 1 2 1 1 1 1 1 1 1 ...
```

```
acc <- length(which(knnPred == testLabelForKNN)) /length(knnPred)
print(acc)
```

```
## [1] 0.8196491
```

```
library(caret)
confusionMatrix(as.factor(knnPred),as.factor(test$RainTomorrow))
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction     1     2
##          1 20049  3173
##          2  1956  3261
##
##                Accuracy : 0.8196
##                  95% CI : (0.8151, 0.8241)
##     No Information Rate : 0.7738
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4479
##
##  Mcnemar’s Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9111
##             Specificity : 0.5068
##          Pos Pred Value : 0.8634
##          Neg Pred Value : 0.6251
##              Prevalence : 0.7738
##          Detection Rate : 0.7050
##    Detection Prevalence : 0.8166
##       Balanced Accuracy : 0.7090
##
##        ’Positive’ Class : 1
##
```

```
#install.packages("tree")
library(tree)
trainForDT <- trainForKNN
head(trainForDT)
```

```
##        MinTemp MaxTemp Rainfall WindGustSpeed WindSpeed9am WindSpeed3pm
## 108993    18.0    20.2      0.0      39.98429           20           20
## 43174     14.2    23.5      4.2      67.00000           17           22
## 34388      8.3    19.1      0.0      74.00000            9           37
## 129197    11.1    26.3      0.2      41.00000           22           15
## 123341    18.3    26.7      0.0      30.00000           13           11
## 85349     18.0    30.3      0.0      20.00000            7            6
##        Humidity9am Humidity3pm Pressure9am Pressure3pm Cloud9am Cloud3pm
## 108993          67          66      1018.0      1017.1 8.000000 8.000000
## 43174           76          26      1007.0      1005.1 4.437189 2.000000
## 34388           73          39      1008.3      1005.9 1.000000 2.000000
## 129197          46          28      1025.3      1021.7 4.437189 4.503167
## 123341          68          58      1011.7      1011.3 6.000000 5.000000
## 85349           65          42      1015.6      1011.9 1.000000 3.000000
##        Temp9am Temp3pm
## 108993    19.2    19.9
## 43174     17.3    23.0
## 34388     12.1    16.4
## 129197    18.3    26.2
## 123341    23.5    25.2
## 85349     23.8    29.8
```

```
trainLabelsForDT <- trainForKNNLabels
testForDt <- testForKNN
```

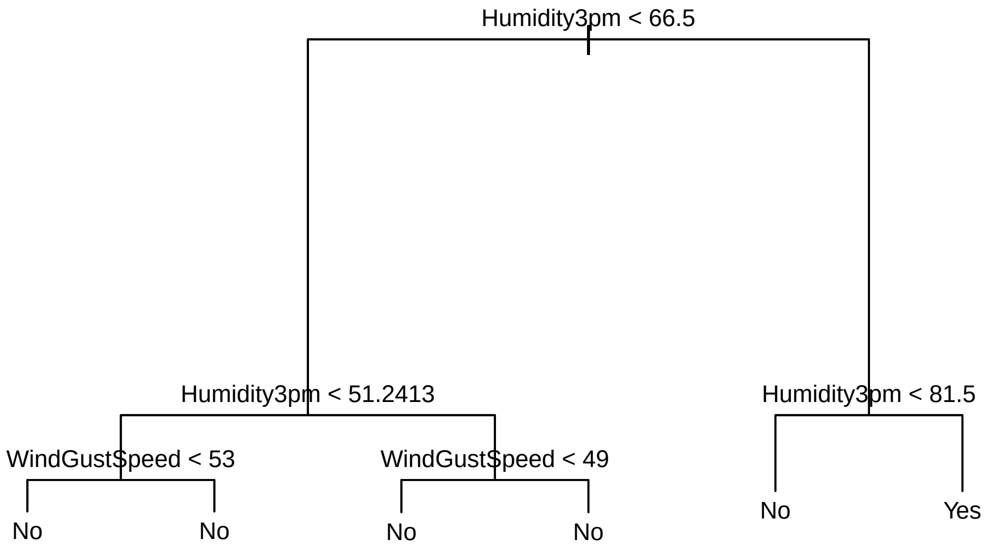
```
head(trainLabelsForDT)
```

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

```
treeWeather <- tree(trainLabelsForDT~., data=trainForDT)
treeWeather
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 113754 120900 No ( 0.77633 0.22367 )
##    2) Humidity3pm < 66.5 87611  67380 No ( 0.87096 0.12904 )
##      4) Humidity3pm < 51.2413 54549  30350 No ( 0.92022 0.07978 )
##        8) WindGustSpeed < 53 46706  20990 No ( 0.94082 0.05918 ) *
##        9) WindGustSpeed > 53 7843   7903 No ( 0.79753 0.20247 ) *
##      5) Humidity3pm > 51.2413 33062  34010 No ( 0.78970 0.21030 )
##       10) WindGustSpeed < 49 27563  25040 No ( 0.83104 0.16896 ) *
##       11) WindGustSpeed > 49 5499   7473 No ( 0.58247 0.41753 ) *
##    3) Humidity3pm > 66.5 26143  36070 Yes ( 0.45921 0.54079 )
##      6) Humidity3pm < 81.5 17012  23030 No ( 0.59011 0.40989 ) *
##      7) Humidity3pm > 81.5 9131   9513 Yes ( 0.21531 0.78469 ) *
```

```
plot(treeWeather)
text(treeWeather, cex=0.75, pretty=0)
```



```
summary(treeWeather)
```

```
##
## Classification tree:
## tree(formula = trainLabelsForDT ~ ., data = trainForDT)
## Variables actually used in tree construction:
## [1] "Humidity3pm"   "WindGustSpeed"
## Number of terminal nodes:  6
## Residual mean deviance:  0.8259 = 93950 / 113700
## Misclassification error rate: 0.178 = 20244 / 113754
```

```
prediction <- predict(treeWeather, newdata = testForDt, type = "class")
table(prediction, test$RainTomorrow)
```

```
##
## prediction     1     2
##        No  21515  4592
##        Yes   490  1842
```

```
levels(prediction) <- list("1" = "No", "2" = "Yes")
library(caret)
confusionMatrix(as.factor(prediction),as.factor(test$RainTomorrow))
```

```
## Confusion Matrix and Statistics
```

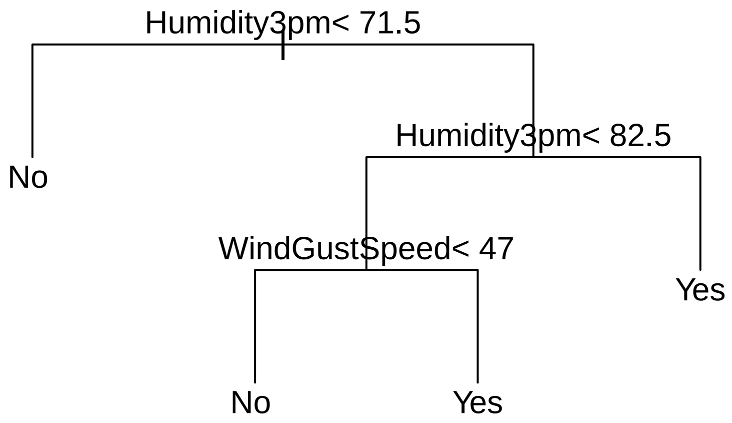
```
##
##           Reference
## Prediction     1     2
##          1 21515  4592
##          2   490  1842
##
##                Accuracy : 0.8213
##                  95% CI : (0.8168, 0.8257)
##     No Information Rate : 0.7738
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3409
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9777
##             Specificity : 0.2863
##          Pos Pred Value : 0.8241
##          Neg Pred Value : 0.7899
##              Prevalence : 0.7738
##          Detection Rate : 0.7565
##    Detection Prevalence : 0.9180
##       Balanced Accuracy : 0.6320
##
##        'Positive' Class : 1
##
```

Repeating Experiment with rpart

```
#install.packages("rpart")
library(rpart)
treeR <- rpart(trainLabelsForDT~., data =trainForDT, method ="class" )
treeR
```

```
## n= 113754
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 113754 25443 No (0.7763331 0.2236669)
##    2) Humidity3pm< 71.5 94998 13856 No (0.8541443 0.1458557) *
##    3) Humidity3pm>=71.5 18756  7169 Yes (0.3822244 0.6177756)
##      6) Humidity3pm< 82.5 10289  4837 No (0.5298863 0.4701137)
##       12) WindGustSpeed< 47 7445  2948 No (0.6040296 0.3959704) *
##       13) WindGustSpeed>=47 2844   955 Yes (0.3357947 0.6642053) *
##      7) Humidity3pm>=82.5 8467  1717 Yes (0.2027873 0.7972127) *
```

```
plot(treeR, uniform=TRUE, margin =0.2)
text(treeR)
```



```
summary(treeR)
```

```
## Call:
## rpart(formula = trainLabelsForDT ~ ., data = trainForDT, method = "class")
##   n= 113754
##
##           CP nsplit rel error    xerror        xstd
## 1 0.17364305      0 1.0000000 1.0000000 0.005523825
## 2 0.03044059      1 0.8263570 0.8263570 0.005145459
## 3 0.01000000      3 0.7654758 0.7660653 0.004995006
##
## Variable importance
##   Humidity3pm      Cloud3pm       Temp3pm WindGustSpeed       MaxTemp
##            79             5             5             3             3
##      Rainfall   Humidity9am  WindSpeed3pm  WindSpeed9am
##             2             1             1             1
##
## Node number 1: 113754 observations,    complexity param=0.173643
##   predicted class=No   expected loss=0.2236669  P(node) =1
##     class counts: 88311 25443
##    probabilities: 0.776 0.224
##   left son=2 (94998 obs) right son=3 (18756 obs)
##   Primary splits:
##       Humidity3pm < 71.5    to the left,  improve=6976.774, (0 missing)
##       Rainfall    < 0.75    to the left,  improve=3933.318, (0 missing)
##       Cloud3pm    < 6.5     to the left,  improve=2813.442, (0 missing)
##       Humidity9am < 76.5    to the left,  improve=2173.394, (0 missing)
##       Cloud9am    < 6.5     to the left,  improve=1712.699, (0 missing)
##   Surrogate splits:
##       Cloud3pm < 7.5     to the left,  agree=0.844, adj=0.057, (0 split)
##       Temp3pm  < 10.55   to the right, agree=0.844, adj=0.052, (0 split)
##       MaxTemp  < 10.55   to the right, agree=0.840, adj=0.028, (0 split)
##       Rainfall < 29.85   to the left,  agree=0.838, adj=0.015, (0 split)
##       Temp9am  < 0.05    to the right, agree=0.836, adj=0.006, (0 split)
## 
## Node number 2: 94998 observations
##   predicted class=No   expected loss=0.1458557  P(node) =0.8351179
##     class counts: 81142 13856
##    probabilities: 0.854 0.146 
## 
## Node number 3: 18756 observations,    complexity param=0.03044059
##   predicted class=Yes  expected loss=0.3822244  P(node) =0.1648821
##     class counts:  7169 11587
##    probabilities: 0.382 0.618 
##   left son=6 (10289 obs) right son=7 (8467 obs)
##   Primary splits:
##       Humidity3pm   < 82.5    to the left,  improve=993.9188, (0 missing)
##       Rainfall      < 2.05    to the left,  improve=514.1706, (0 missing)
##       Pressure3pm   < 1012.65 to the right, improve=423.8869, (0 missing)
##       Pressure9am   < 1015.05 to the right, improve=396.1862, (0 missing)
##       WindGustSpeed < 45      to the left,  improve=393.3555, (0 missing)
##   Surrogate splits:
##       Cloud3pm    < 7.5     to the left,  agree=0.612, adj=0.140, (0 split)
##       Humidity9am < 89.5    to the left,  agree=0.609, adj=0.134, (0 split)
##       Temp3pm     < 12.35   to the right, agree=0.598, adj=0.110, (0 split)
##       MaxTemp     < 13.25   to the right, agree=0.586, adj=0.083, (0 split)
##       Rainfall    < 6.65    to the left,  agree=0.579, adj=0.068, (0 split)
## 
## Node number 6: 10289 observations,    complexity param=0.03044059
##   predicted class=No   expected loss=0.4701137  P(node) =0.09044957
##     class counts:  5452  4837
##    probabilities: 0.530 0.470 
##   left son=12 (7445 obs) right son=13 (2844 obs)
##   Primary splits:
##       WindGustSpeed < 47      to the left,  improve=296.1295, (0 missing)
##       Rainfall      < 2.05    to the left,  improve=242.9582, (0 missing)
##       Pressure3pm   < 1012.65 to the right, improve=242.3097, (0 missing)
##       Pressure9am   < 1015.25 to the right, improve=235.2660, (0 missing)
##       Cloud3pm      < 6.5     to the left,  improve=139.7565, (0 missing)
##   Surrogate splits:
##       WindSpeed3pm < 27      to the left,  agree=0.802, adj=0.283, (0 split)
##       WindSpeed9am < 23      to the left,  agree=0.794, adj=0.254, (0 split)
##       Pressure9am  < 1007.65 to the right, agree=0.737, adj=0.050, (0 split)
##       Pressure3pm  < 1003.75 to the right, agree=0.735, adj=0.043, (0 split)
##       Humidity9am  < 56.5    to the right, agree=0.728, adj=0.014, (0 split)
## 
## Node number 7: 8467 observations
##   predicted class=Yes  expected loss=0.2027873  P(node) =0.07443255
##     class counts:  1717  6750
##    probabilities: 0.203 0.797
##
## Node number 12: 7445 observations
##   predicted class=No   expected loss=0.3959704  P(node) =0.06544825
##     class counts:  4497  2948
##    probabilities: 0.604 0.396
##
## Node number 13: 2844 observations
##   predicted class=Yes  expected loss=0.3357947  P(node) =0.02500132
##     class counts:   955  1889
##    probabilities: 0.336 0.664
```

```
prediction1 <- predict(treeR, newdata = testForDt, type = "class")
table(prediction, test$RainTomorrow)
```

```
##
## prediction     1     2
##          1 21515  4592
##          2   490  1842
```

```
levels(prediction1) <- list("1" = "No", "2" = "Yes")
library(caret)
confusionMatrix(as.factor(prediction1),as.factor(test$RainTomorrow))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     1     2
##          1 21309  4237
##          2   696  2197
##
##                Accuracy : 0.8265
##                  95% CI : (0.8221, 0.8309)
##     No Information Rate : 0.7738
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3848
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9684
##             Specificity : 0.3415
##          Pos Pred Value : 0.8341
##          Neg Pred Value : 0.7594
##              Prevalence : 0.7738
##          Detection Rate : 0.7493
##    Detection Prevalence : 0.8983
##       Balanced Accuracy : 0.6549
##
##        'Positive' Class : 1
##
```

Working of Algorithms:

Logistic Regression: It is the statistical analysis method which predicts the output based on the prior observation of a data set. Logistic regression focuses on decreasing the loss function on each iteration using the concept of gradient descent and learning rate. It will adjust the value of w. It tries to minimize the loss as long as it can for the given data and output the log odd and this can be later converted to probability.

KNN Classification: KNN is the machine learning algorithm which can be used for both regression and classification but I am going to focus on classification. It tries to classify different categories based on the distance. It tries to create the group of K data based on the euclidian distance or other distances.

Decision Tree: This is the recursive, top-down, greedy algorithm used for classification. Decision tree works by classifying the features into two or more branches based on the features. Entropy and Gini index are used as the metric in decision trees.

Summary of Results.

Looking at the result of Logistic regression, we got accuracy of about 84% and sensitivity was about 0.94, specificity was 0.4733 which are the preety good metrices. Also, for KNN classification,accuracy is 0.8196 and sensitivity is 0.91, specificity is 0.51. The metrices for logistic regression and KNN was kind of similar. For the KNN I have taken the value of K as 3. The accuracy and other metrices of KNN can be changed by changing the value of K. Usually, it is okay to take square root of no. of obseration but, since I have a lot of data set I have used k as 3 but can be changed and see how it will affect the metrices.The accuracy of Decision tree is 84 % and sensitivity and specificity are 0.96 and 0.34 respectively. The metrics of Decision tree is almost similar to the other two. The accuracy and other metrics of decision tree can be changed by using the concept of tree pruning.