- **Question 1**

Summary:

A(I)

For only Rain and Snow Events.

	Accuracy	Precision	Recall
Logistic Regression	0.9571	0.9742	0.9716
LDA	0.9358	0.9656	0.9525
QDA	0.9319	0.9825	0.93
KNN	0.96391	0.9808	0.9733

1. In this model as, we are considering only two events, the accuracy is high.
2. The precision and recall are also near to 1.

A(II)

1. Performing forward selection for predictor selection. Automatic methods are useful when the number of explanatory variables is large, and it is not feasible to fit all possible models.

2. From the above statistic we can consider the features Dew_Point.F, Sea_Level_Press.in, Humidity.percentage. These features have the p-value higher than the threshold.

3. The accuracy Precision and recall are reduced when we eliminate few features.

Dew_Point.F, Sea_Level_Press.in, Humidity.percentage	Accuracy	Precision	Recall
Logistic Regression	0.960	0.975	0.9747
LDA	0.9556	0.9772	0.966
QDA	0.960	0.972	0.977
KNN	0.961	0.978	0.973

4. When we run the models with above features, the accuracy, precision, and recall are almost the same as the previous.
5. As the features generated are very influential we observe that the accuracy, precision, and recall do not vary much.

B(I)

For only Rain, Rain_Thunderstorm and Snow Events

	Accuracy	Precision	Recall
LDA	0.756	0.818	0.966
QDA	0.743	0.814	0.956
KNN	0.760	0.811	0.983

1.There are three different events for the data, so we observe that the accuracy decreases than when there were only rain and snow events.
2. The recall for these events are close to 1 but the recall is decreased.
3. We need more data and features when we have multiple classes to categorize to give more accuracy.

B(II)

1.From forward selection approach the predictors we are considering are Sea_Level_Press.in, Wind.mph, Precip.in

Sea_Level_Press.in, Wind.mph, Precip.in	Accuracy	Precision	Recall
LDA	0.6584	0.8989	0.9636
QDA	0.598	0.9419	0.852
KNN	0.63	0.81	0.965

2. When we build the model with all features we observe that the accuracy, precision, and recall are better than the selected features.

3. As the model has different events we observe that the accuracy for LDA, QDA, KNN decreases.

4. Precision and recall almost remain the same as all features model.

You're to use the KC Weather Data ("kc_weather_srt.csv", available from Start Here). The data has categorized the weather for each day into three categories ("Events": Rain, Rain_Thunderstorm, Snow) over the three years 2014, 2015, and 2016. You'll note that not all dates are listed because it's a filtered subset where other categories or no events are deleted to have a more manageable subset. The entire dataset has 366 entries. The column labels indicate the units as well such as Temp.F means temperature in Fahrenheit, Visibility.mi means Visibility in miles, etc.
You're to do two level of analysis

Data Analysis:

```
> setwd("C:/Users/putha/Desktop/ISL/Assignment2")
>
> KCWeather_data_csv = read.csv("kc_weather_srt.csv",header = T,na.strings = "?")
> names(KCWeather_data_csv)
[1] "Date"                "Temp.F"              "Dew_Point.F"
[4] "Humidity.percentage" "Sea_Level_Press.in"  "Visibility.mi"
[7] "Wind.mph"            "Precip.in"           "Events"
> dim(KCWeather_data_csv)
[1] 366   9
> summary(KCWeather_data_csv)
        Date          Temp.F       Dew_Point.F    Humidity.percentage
 2014-1-1 :  1   Min.   : 5.00   Min.   :-5.00   Min.   :29.00
 2014-1-10:  1   1st Qu.:46.00   1st Qu.:34.00   1st Qu.:62.00
 2014-1-11:  1   Median :62.50   Median :52.00   Median :72.00
 2014-1-12:  1   Mean   :58.74   Mean   :47.74   Mean   :69.85
 2014-1-14:  1   3rd Qu.:74.00   3rd Qu.:64.00   3rd Qu.:79.00
 2014-1-15:  1   Max.   :88.00   Max.   :77.00   Max.   :95.00
 (Other)  :360
 Sea_Level_Press.in Visibility.mi      Wind.mph        Precip.in
 Min.   :29.33      Min.   : 4.000   Min.   : 1.000   Min.   :0.0000
 1st Qu.:29.83      1st Qu.: 8.000   1st Qu.: 7.000   1st Qu.:0.0000
 Median :29.95      Median :10.000   Median : 9.000   Median :0.0250
 Mean   :29.97      Mean   : 9.014   Mean   : 9.079   Mean   :0.1728
 3rd Qu.:30.11      3rd Qu.:10.000   3rd Qu.:11.000   3rd Qu.:0.1900
 Max.   :30.90      Max.   :10.000   Max.   :19.000   Max.   :2.2700

               Events
 Rain             :176
 Rain_Thunderstorm:140
 Snow             : 50
```

Checking for all the pairwise correlation in the predictor sets.

```
> cor(KCWeather_data_csv)
Error in cor(KCWeather_data_csv) : 'x' must be numeric
```

In the dataset the column Events is the qualitative i.e., rain or snow or rain thunderstorm and Date is also not qualitative. So, we are excluding Date and Event from the cor ().

```
> cor(KCWeather_data_csv [2:8])
                       Temp.F Dew_Point.F Humidity.percentage
Temp.F              1.0000000   0.9616863           0.1854993
Dew_Point.F         0.9616863   1.0000000           0.4357141
Humidity.percentage 0.1854993   0.4357141           1.0000000
Sea_Level_Press.in -0.5161536  -0.5193711          -0.1789127
Visibility.mi       0.2806759   0.1338507          -0.5161982
Wind.mph           -0.1877029  -0.2634530          -0.3401291
Precip.in           0.2526478   0.3337662           0.4063828
                   Sea_Level_Press.in Visibility.mi    Wind.mph   Precip.in
Temp.F                    -0.51615357    0.28067587 -0.18770291  0.25264783
Dew_Point.F               -0.51937106    0.13385067 -0.26345299  0.33376616
Humidity.percentage       -0.17891273   -0.51619817 -0.34012909  0.40638281
Sea_Level_Press.in         1.00000000   -0.06376886 -0.15521066 -0.21059474
Visibility.mi             -0.06376886    1.00000000  0.09221678 -0.32739095
Wind.mph                  -0.15521066    0.09221678  1.00000000 -0.01940138
Precip.in                 -0.21059474   -0.32739095 -0.01940138  1.00000000
```

a. Consider first the subset that consists only of Rain and Snow. There are 226 entries with these two categories.

i. Apply logistic regression, LDA, QDA, and knn on this dataset to determine the accuracy, precision, and recall of these models. You're to use randomly 180 days for the training set (approximately 80% of 226) and the rest for the test data set. Conduct your study over 100 replications, and summary the result of your analysis with your conclusion which models you'll recommend to use based on the metrics: accuracy, precision and recall.

- Reading the Events of **only Rain and Snow** and writing into a new csv file.

```
> library(sqldf)
Loading required package: gsubfn
Loading required package: proto
Loading required package: RSQLite
Warning messages:
1: package 'sqldf' was built under R version 3.4.2
2: package 'gsubfn' was built under R version 3.4.2
3: package 'proto' was built under R version 3.4.2
> write.csv(KCWeather_RS, "KCWeather_RS.csv", quote = FALSE, row.names = FALSE)
> KCWeather_RS <- read.csv.sql("kc_weather_srt.csv",
+     sql = "select * from file where Sepal.Events!='Rain_Thunderstorm'", eol = "\n")
Error in rsqlite_send_query(conn@ptr, statement) :
  no such column: Sepal.Events
>     sql = "select * from file where Sepal.Events!='Rain_Thunderstorm'", eol = "\n")
Error: unexpected ',' in "    sql = "select * from file where Sepal.Events!='Rain_Thunderstorm'","
> KCWeather_RS <- read.csv.sql("kc_weather_srt.csv",
+     sql = "select * from file where Events!='Rain_Thunderstorm'", eol = "\n")
Warning messages:
1: In getInlineHandler(name, info$package) :
  closing unused connection 5 (kc_weather_srt.csv)
2: In getInlineHandler(name, info$package) :
  closing unused connection 4 (kc_weather_srt.csv)
3: In getInlineHandler(name, info$package) :
  closing unused connection 3 (kc_weather_srt.csv)
> dim(KCWeather_RS)
[1] 226   9
```

- Building the model for 100 replications.

```
> library(MASS)
> data=read.csv("KCWeather_RS.csv")
>
> logisticAccuracy = dim(100)
> logisticPrecision = dim(100)
> logisticRecall = dim(100)
> ldaAccuracy=dim(100)
> ldaPrecision=dim(100)
> ldaRecall=dim(100)
> qdaAccuracy=dim(100)
> qdaPrecision=dim(100)
> qdaRecall=dim(100)
> knnAccuracy=dim(100)
> knnPrecision=dim(100)
> knnRecall=dim(100)
> for(k in 1:100){
+   index <- createDataPartition(data$Events, p = .79,list = FALSE)
+   trainData <- data[ index, ]
+   test <- data[-index, ]
+
+    #log
+    glm_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                     +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="glm", family="binomial")
+   result_log=predict(glm_fit, newdata=test)
+   cm_log=confusionMatrix(result_log, test$Events)
+   logisticAccuracy[k]=cm_log$overall[1]
+   logisticPrecision[k]=cm_log$byClass[5]
+   logisticRecall[k]=cm_log$byClass[6]
+
+   #lda
+   lda_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                     +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="lda", family="binomial")
+   result_lda=predict(lda_fit, newdata=test)
+   cm_lda=confusionMatrix(result_lda, test$Events)
+   ldaAccuracy[k]=cm_lda$overall[1]
+   ldaPrecision[k]=cm_lda$byClass[5]
+   ldaRecall[k]=cm_lda$byClass[6]
+   #qda
+   qda_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                     +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="qda")
+   result_qda=predict(qda_fit, newdata=test)
+   cm_qda=confusionMatrix(result_qda, test$Events)
+  qdaAccuracy[k]=cm_qda$overall[1]
+   qdaPrecision[k]=cm_qda$byClass[5]
+   qdaRecall[k]=cm_qda$byClass[6]
+
+
+   #knn
+   knn_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                     +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="knn")
+   result_knn=predict(knn_fit, newdata=test)
+   cm_knn=confusionMatrix(result_knn, test$Events)
+   knnAccuracy[k]=cm_knn$overall[1]
+   knnPrecision[k]=cm_knn$byClass[5]
+   knnRecall[k]=cm_knn$byClass[6]
+
+ }
```

```
> length(index)
[1] 180
```

- **Logistic Regression:**

```
> #log statistics
> meanlogisticAccuracy = mean(logisticAccuracy)
> meanlogisticAccuracy
[1] 0.9571739
> meanlogisticPrecision = mean(logisticPrecision)
> meanlogisticPrecision
[1] 0.9742915
> meanlogisticRecall = mean(logisticRecall)
> meanlogisticRecall
[1] 0.9716667
```

```
> cm_log
Confusion Matrix and Statistics

          Reference
Prediction Rain Snow
      Rain   34    2
      Snow    2    8

               Accuracy : 0.913
                 95% CI : (0.7921, 0.9758)
    No Information Rate : 0.7826
    P-Value [Acc > NIR] : 0.01763

                  Kappa : 0.7444
 Mcnemar's Test P-Value : 1.00000

            Sensitivity : 0.9444
            Specificity : 0.8000
         Pos Pred Value : 0.9444
         Neg Pred Value : 0.8000
             Prevalence : 0.7826
         Detection Rate : 0.7391
   Detection Prevalence : 0.7826
      Balanced Accuracy : 0.8722

       'Positive' Class : Rain
```

- **Linear Discriminant Analysis:**

```
> #lda statistics
> meanldaAccuracy=mean(ldaAccuracy)
> meanldaAccuracy
[1] 0.9358696
> meanldaPrecision=mean(ldaPrecision)
> meanldaPrecision
[1] 0.9656779
> meanldaRecall=mean(ldaRecall)
> meanldaRecall
[1] 0.9525
```

```
> cm_lda
Confusion Matrix and Statistics

          Reference
Prediction Rain Snow
      Rain   34    3
      Snow    2    7

               Accuracy : 0.8913
                 95% CI : (0.7643, 0.9638)
    No Information Rate : 0.7826
    P-Value [Acc > NIR] : 0.04637

                  Kappa : 0.6686
 Mcnemar's Test P-Value : 1.00000

            Sensitivity : 0.9444
            Specificity : 0.7000
         Pos Pred Value : 0.9189
         Neg Pred Value : 0.7778
             Prevalence : 0.7826
         Detection Rate : 0.7391
   Detection Prevalence : 0.8043
      Balanced Accuracy : 0.8222

       'Positive' Class : Rain
```

- **Quadratic Discriminant Analysis:**

```
> #qda statistics
> meanqdaAccuracy=mean(qdaAccuracy)
> meanqdaAccuracy
[1] 0.9319565
> meanqdaPrecision=mean(qdaPrecision)
> meanqdaPrecision
[1] 0.9825555
> meanqdaRecall=mean(qdaRecall)
> meanqdaRecall
[1] 0.93
```

```
> cm_qda
Confusion Matrix and Statistics

          Reference
Prediction Rain Snow
      Rain   34    1
      Snow    2    9

               Accuracy : 0.9348
                 95% CI : (0.821, 0.9863)
    No Information Rate : 0.7826
    P-Value [Acc > NIR] : 0.005312

                  Kappa : 0.815
 Mcnemar's Test P-Value : 1.000000

            Sensitivity : 0.9444
            Specificity : 0.9000
         Pos Pred Value : 0.9714
         Neg Pred Value : 0.8182
             Prevalence : 0.7826
         Detection Rate : 0.7391
   Detection Prevalence : 0.7609
      Balanced Accuracy : 0.9222

       'Positive' Class : Rain
```

- **K-nearest neighbor:**

```
> #knn statistics
> meanknnAccuracy=mean(knnAccuracy)
> meanknnAccuracy
[1] 0.963913
> meanknnPrecision=mean(knnPrecision)
> meanknnPrecision
[1] 0.9808364
> meanknnRecall=mean(knnRecall)
> meanknnRecall
[1] 0.9733333
```

```
> cm_knn
Confusion Matrix and Statistics

          Reference
Prediction Rain Snow
      Rain   35    0
      Snow    1   10

               Accuracy : 0.9783
                 95% CI : (0.8847, 0.9994)
    No Information Rate : 0.7826
    P-Value [Acc > NIR] : 0.0001747

                  Kappa : 0.9383
 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.9722
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9091
             Prevalence : 0.7826
         Detection Rate : 0.7609
   Detection Prevalence : 0.7609
      Balanced Accuracy : 0.9861

       'Positive' Class : Rain
```

ii. Discuss and analysis in a systematic way you would consider eliminating some of the predictors and see if your accuracy, precision and recall improves.

Using the forward selection for eliminating the predictors. And checking accuracy.

```
> null=glm(Events~1, data=data, family="binomial")
> full=glm(Events~., data=data, family="binomial")
Warning message:
glm.fit: algorithm did not converge
> step(null, scope=list(lower=null, upper=full), direction="forward")
Start:  AIC=240.87
Events ~ 1
```

```
+ Precip.in            1   36.864  44.86
+ Wind.mph             1   38.046  46.05
+ Visibility.mi        1   38.505  46.51
+ Date               223    0.000 452.00

Step:  AIC=44.03
Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in

                 Df Deviance    AIC
<none>                36.030  44.03
+ Temp.F          1   34.113  44.11
+ Precip.in       1   34.380  44.38
+ Wind.mph        1   34.978  44.98
+ Visibility.mi   1   35.808  45.81
+ Date          222    0.000 452.00

Call:  glm(formula = Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in,
    family = "binomial", data = data)

Coefficients:
        (Intercept)          Dew_Point.F  Humidity.percentage   Sea_Level_Press.in
          -136.5993              -0.4523               0.0714               4.7220

Degrees of Freedom: 225 Total (i.e. Null);  222 Residual
Null Deviance:      238.9
Residual Deviance: 36.03        AIC: 44.03
```

```
 #log
 glm_fit <- train(Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in,  data=trainData, method="glm", family="binomial")
result_log=predict(glm_fit, newdata=test)
cm_log=confusionMatrix(result_log, test$Events)
logisticAccuracy[k]=cm_log$overall[1]
logisticPrecision[k]=cm_log$byClass[5]
logisticRecall[k]=cm_log$byClass[6]

#lda
lda_fit <- train(Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in,  data=trainData, method="lda", family="binomial")
result_lda=predict(lda_fit, newdata=test)
cm_lda=confusionMatrix(result_lda, test$Events)
ldaAccuracy[k]=cm_lda$overall[1]
ldaPrecision[k]=cm_lda$byClass[5]
ldaRecall[k]=cm_lda$byClass[6]

#qda
qda_fit <- train(Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in,  data=trainData, method="qda")
result_qda=predict(qda_fit, newdata=test)
cm_qda=confusionMatrix(result_qda, test$Events)
qdaAccuracy[k]=cm_qda$overall[1]
qdaPrecision[k]=cm_qda$byClass[5]
qdaRecall[k]=cm_qda$byClass[6]


#knn
knn_fit <- train(Events ~ Dew_Point.F + Humidity.percentage + Sea_Level_Press.in,  data=trainData, method="knn")
result_knn=predict(knn_fit, newdata=test)
cm_knn=confusionMatrix(result_knn, test$Events)
knnAccuracy[k]=cm_knn$overall[1]
knnPrecision[k]=cm_knn$byClass[5]
knnRecall[k]=cm_knn$byClass[6]
```

```
> #log statistics
> meanlogisticAccuracy = mean(logisticAccuracy)
> meanlogisticAccuracy
[1] 0.9606522
> meanlogisticPrecision = mean(logisticPrecision)
> meanlogisticPrecision
[1] 0.9755195
> meanlogisticRecall = mean(logisticRecall)
> meanlogisticRecall
[1] 0.9747222
>
> #lda statistics
> meanldaAccuracy=mean(ldaAccuracy)
> meanldaAccuracy
[1] 0.9556522
> meanldaPrecision=mean(ldaPrecision)
> meanldaPrecision
[1] 0.9772727
> meanldaRecall=mean(ldaRecall)
> meanldaRecall
[1] 0.9663889
> #qda statistics
> meanqdaAccuracy=mean(qdaAccuracy)
> meanqdaAccuracy
[1] 0.9608696
> meanqdaPrecision=mean(qdaPrecision)
> meanqdaPrecision
[1] 0.9728722
> meanqdaRecall=mean(qdaRecall)
> meanqdaRecall
[1] 0.9777778
>
> #knn statistics
> meanknnAccuracy=mean(knnAccuracy)
> meanknnAccuracy
[1] 0.9619565
> meanknnPrecision=mean(knnPrecision)
> meanknnPrecision
[1] 0.9787285
> meanknnRecall=mean(knnRecall)
> meanknnRecall
[1] 0.9730556
```

b. Consider next the entire dataset consisting of 366 entries. Now logistics regression cannot be applied, but you can apply the rest of them. Repeat the above studies in i) and ii) with LDA, QDA, and knn on the entire data set (using 290 of them in a training set). Do not forget randomization and 100 replications for this study.

- Building the model

```
> setwd("C:/Users/putha/Desktop/ISL/Assignment2")
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
Warning messages:
1: package ‘caret’ was built under R version 3.4.2
2: package ‘ggplot2’ was built under R version 3.4.2
> library(MASS)
> data=read.csv("kc_weather_srt.csv")
>
> ldaAccuracy=dim(100)
> ldaPrecision=dim(100)
> ldaRecall=dim(100)
> qdaAccuracy=dim(100)
> qdaPrecision=dim(100)
> qdaRecall=dim(100)
> knnAccuracy=dim(100)
> knnPrecision=dim(100)
> knnRecall=dim(100)
>
> for(k in 1:100){
+   index <- createDataPartition(data$Events, p = .789,list = FALSE)
+   length(index)
+   trainData <- data[ index, ]
+   test <- data[-index, ]

+   #lda
+   lda_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                 +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="lda", family="binomial")
+   result_lda=predict(lda_fit, newdata=test)
+   cm_lda=confusionMatrix(result_lda, test$Events)
+   ldaAccuracy[k]=cm_lda$overall[1]
+   ldaPrecision[k]=cm_lda$byClass[5]
+   ldaRecall[k]=cm_lda$byClass[6]
```

```
+   #qda
+   qda_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                    +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="qda")
+   result_qda=predict(qda_fit, newdata=test)
+   cm_qda=confusionMatrix(result_qda, test$Events)
+  qdaAccuracy[k]=cm_qda$overall[1]
+   qdaPrecision[k]=cm_qda$byClass[5]
+   qdaRecall[k]=cm_qda$byClass[6]
+
+
+   #knn
+   knn_fit <- train(Events ~ Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in
+                    +Visibility.mi+Wind.mph+Precip.in,  data=trainData, method="knn")
+   result_knn=predict(knn_fit, newdata=test)
+   cm_knn=confusionMatrix(result_knn, test$Events)
+   knnAccuracy[k]=cm_knn$overall[1]
+   knnPrecision[k]=cm_knn$byClass[5]
+   knnRecall[k]=cm_knn$byClass[6]
+
+ }
```

- **Linear Discriminant Analysis:**

```
> #lda statistics
> meanldaAccuracy=mean(ldaAccuracy)
> meanldaAccuracy
[1] 0.7565789
> meanldaPrecision=mean(ldaPrecision)
> meanldaPrecision
[1] 0.8182979
> meanldaRecall=mean(ldaRecall)
> meanldaRecall
[1] 0.9668182
>
```

```
> cm_lda
Confusion Matrix and Statistics

                   Reference
Prediction           Rain Rain_Thunderstorm Snow
  Rain                 27                 5    0
  Rain_Thunderstorm     6                24    0
  Snow                  4                 0   10

Overall Statistics

               Accuracy : 0.8026
                 95% CI : (0.6954, 0.8851)
    No Information Rate : 0.4868
    P-Value [Acc > NIR] : 1.407e-08

                  Kappa : 0.6817
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Rain Class: Rain_Thunderstorm Class: Snow
Sensitivity               0.7297                   0.8276      1.0000
Specificity               0.8718                   0.8723      0.9394
Pos Pred Value            0.8438                   0.8000      0.7143
Neg Pred Value            0.7727                   0.8913      1.0000
Prevalence                0.4868                   0.3816      0.1316
Detection Rate            0.3553                   0.3158      0.1316
Detection Prevalence      0.4211                   0.3947      0.1842
Balanced Accuracy         0.8008                   0.8500      0.9697
```

- **Quadratic Discriminant Analysis:**

```
> #qda statistics
> meanqdaAccuracy=mean(qdaAccuracy)
> meanqdaAccuracy
[1] 0.7436842
> meanqdaPrecision=mean(qdaPrecision)
> meanqdaPrecision
[1] 0.8140426
> meanqdaRecall=mean(qdaRecall)
> meanqdaRecall
[1] 0.9569697
> cm_qda
Confusion Matrix and Statistics

                   Reference
Prediction          Rain Rain_Thunderstorm Snow
  Rain                26                10    0
  Rain_Thunderstorm    8                19    0
  Snow                 3                 0   10

Overall Statistics

               Accuracy : 0.7237
                 95% CI : (0.6091, 0.8201)
    No Information Rate : 0.4868
    P-Value [Acc > NIR] : 2.336e-05

                  Kappa : 0.548
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Rain Class: Rain_Thunderstorm Class: Snow
Sensitivity               0.7027                   0.6552      1.0000
Specificity               0.7436                   0.8298      0.9545
Pos Pred Value            0.7222                   0.7037      0.7692
Neg Pred Value            0.7250                   0.7959      1.0000
Prevalence                0.4868                   0.3816      0.1316
Detection Rate            0.3421                   0.2500      0.1316
Detection Prevalence      0.4737                   0.3553      0.1711
Balanced Accuracy         0.7231                   0.7425      0.9773
```

- **K-nearest neighbor:**

```
> #knn statistics
> meanknnAccuracy=mean(knnAccuracy)
> meanknnAccuracy
[1] 0.7602632
> meanknnPrecision=mean(knnPrecision)
> meanknnPrecision
[1] 0.8112766
> meanknnRecall=mean(knnRecall)
> meanknnRecall
[1] 0.9834848
> cm_knn
Confusion Matrix and Statistics

                   Reference
Prediction          Rain Rain_Thunderstorm Snow
  Rain                30                 9    0
  Rain_Thunderstorm    5                20    0
  Snow                 2                 0   10

Overall Statistics

               Accuracy : 0.7895
                 95% CI : (0.6808, 0.8746)
    No Information Rate : 0.4868
    P-Value [Acc > NIR] : 5.791e-08

                  Kappa : 0.6514
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Rain Class: Rain_Thunderstorm Class: Snow
Sensitivity               0.8108                   0.6897      1.0000
Specificity               0.7692                   0.8936      0.9697
Pos Pred Value            0.7692                   0.8000      0.8333
Neg Pred Value            0.8108                   0.8235      1.0000
Prevalence                0.4868                   0.3816      0.1316
Detection Rate            0.3947                   0.2632      0.1316
Detection Prevalence      0.5132                   0.3289      0.1579
Balanced Accuracy         0.7900                   0.7916      0.9848
```

. Discuss and analysis in a systematic way you would consider eliminating some of the predictors and see if your accuracy, precision and recall improves.

Using the forward selection. We select only Precip.in, Sea_Level_Press.in, Wind.mph

```
> data=read.csv("kc_weather_srt.csv",header = TRUE, sep = ",")
>
> null=glm(Events~1, data=data, family="binomial")
> full=glm(Events~., data=data, family="binomial")
Warning message:
glm.fit: algorithm did not converge
> step(null, scope=list(lower=null, upper=full), direction="forward")
```

```
Step:  AIC=470.87
Events ~ Precip.in + Sea_Level_Press.in + Wind.mph

                      Df Deviance    AIC
<none>                     462.87 470.87
+ Dew_Point.F          1   461.93 471.93
+ Humidity.percentage  1   462.24 472.24
+ Temp.F               1   462.30 472.30
+ Visibility.mi        1   462.86 472.86
+ Date               362     0.00 732.00

Call:  glm(formula = Events ~ Precip.in + Sea_Level_Press.in + Wind.mph,
    family = "binomial", data = data)

Coefficients:
       (Intercept)          Precip.in  Sea_Level_Press.in            Wind.mph
         -42.47577            3.00582             1.38199             0.07796

Degrees of Freedom: 365 Total (i.e. Null);  362 Residual
Null Deviance:      506.8
Residual Deviance: 462.9        AIC: 470.9
```

```
+   #lda
+   lda_fit <- train(Events ~ Sea_Level_Press.in+Wind.mph+Precip.in,  data=trainData, method="lda", family="binomial")
+   result_lda=predict(lda_fit, newdata=test)
+   cm_lda=confusionMatrix(result_lda, test$Events)
+   ldaAccuracy[k]=cm_lda$overall[1]
+   ldaPrecision[k]=cm_lda$byClass[5]
+   ldaRecall[k]=cm_lda$byClass[6]
+
+   #qda
+   qda_fit <- train(Events ~ Sea_Level_Press.in+Wind.mph+Precip.in,  data=trainData, method="qda")
+   result_qda=predict(qda_fit, newdata=test)
+   cm_qda=confusionMatrix(result_qda, test$Events)
+  qdaAccuracy[k]=cm_qda$overall[1]
+   qdaPrecision[k]=cm_qda$byClass[5]
+   qdaRecall[k]=cm_qda$byClass[6]
+
+
+   #knn
+   knn_fit <- train(Events ~ Sea_Level_Press.in+Wind.mph+Precip.in,  data=trainData, method="knn")
+   result_knn=predict(knn_fit, newdata=test)
+   cm_knn=confusionMatrix(result_knn, test$Events)
+   knnAccuracy[k]=cm_knn$overall[1]
+   knnPrecision[k]=cm_knn$byClass[5]
+   knnRecall[k]=cm_knn$byClass[6]
+
```

```
> #lda statistics
> meanldaAccuracy=mean(ldaAccuracy)
> meanldaAccuracy
[1] 0.6584211
> meanldaPrecision=mean(ldaPrecision)
> meanldaPrecision
[1] 0.8989362
> meanldaRecall=mean(ldaRecall)
> meanldaRecall
[1] 0.9636364
>
> #qda statistics
> meanqdaAccuracy=mean(qdaAccuracy)
> meanqdaAccuracy
[1] 0.5982895
> meanqdaPrecision=mean(qdaPrecision)
> meanqdaPrecision
[1] 0.9419149
> meanqdaRecall=mean(qdaRecall)
> meanqdaRecall
[1] 0.8527273
>
> #knn statistics
> meanknnAccuracy=mean(knnAccuracy)
> meanknnAccuracy
[1] 0.6396053
> meanknnPrecision=mean(knnPrecision)
> meanknnPrecision
[1] 0.8193617
> meanknnRecall=mean(knnRecall)
> meanknnRecall
[1] 0.9654545
```

References:

1. http://www.stat.columbia.edu/~martin/W2024/R10.pdf
2. https://rstudio-pubs-static.s3.amazonaws.com/64455_df98186f15a64e0ba37177de8b4191fa.html