Problem1:

1.1.i:

Logistic Regression:

Source code:

```
> data<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather <- subset(data,Events=="Snow"|Events=="Rain")
> kcweather$Events<-ifelse(kcweather$Events=="Rain",1,0)
> kcweather$Events<-as.character(kcweather$Events)
> kcweather$Events<-as.numeric(as.character(kcweather$Events))
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
>
> n=226
> nt=180
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision=dim(rep)
> recall=dim(rep)
> |
```

```
> precision=dim(rep)
> recall=dim(rep)
> for (k in 1:rep) {
+   train=sample(1:n,nt)
+   kcweather.train = kcweather[train,1:9]
+   kcweather.test = kcweather[-train,1:9]
+   model=glm(Events~.,kcweather.train,family="binomial")
+   res=predict(model,kcweather.test)
+   tablin=table(Actualvalue=kcweather.test$Events,Predictedvalue=res>0.5)
+   accuracy[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9556522> cat("Precision :  ",mean(precision))
Precision :   0.8789409> cat("Recall: ",mean(recall))
Recall:  0.9217205>
> |
```

The accuracy =0.95, precision=0.87, Recall=0.92

Here the threshold value for probability is chosen manually which is greater than 0.5. This is used to calculate the recall and precision.

LDA:

Sourcecode:

```
> library(MASS)
>  data<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",
> kc_weather<-subset(data,Events=="Snow"|Events=="Rain")
> kc_weather$Events<-as.character(kc_weather$Events)
> kc_weather$Date=as.integer(gsub("-","",kc_weather$Date)
+ n=226
Error: unexpected symbol in:
"kc_weather$Date=as.integer(gsub("-","",kc_weather$Date)
n"
> kc_weather$Date=as.integer(gsub("-","",kc_weather$Date))
> n=226
> nt=180
> neval=n-nt
> rep=100
> errlin=dim(rep)
> accuracy=dim(rep)
> precision=dim(rep)
> recall=dim(rep)
```

```
> recall=dim(rep)
> for(k in 1:rep){
+ train=sample(1:n,nt)
+ kc_weather.lda_train=lda(Events~.,kc_weather[train,])
+ tablin=table(kc_weather$Events[-train],predict(kc_weather.lda_train,kc_weather[-train,])$clas
+      accuracy[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+      precision[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+      recall[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+     errlin[k] = (neval-sum(diag(tablin)))/neval
+   }
>   cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9319565>   cat("Precision :  ",mean(precision))
Precision :   0.9512263>   cat("Recall: ",mean(recall))
Recall:  0.9623049> |
```

Here the accuracy is 0.93, precison=0.95, recall is 0.96

QDA:

```
> library(MASS)
>  data<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather <- subset(data,Events=="Snow"|Events=="Rain")
> kcweather$Events<-as.character(kcweather$Events)
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
> n=226
> nt=180
> neval=n-nt
> rep=100
> errlin=dim(rep)
> accuracy=dim(rep)
> precision=dim(rep)
```

```
> precision=dim(rep)
> recall=dim(rep)
> for  (k in 1:rep) {
+   train=sample(1:n,nt)
+   kcweather.qda_train = qda(Events~.,kcweather[train,])
+   tablin=table(kcweather$Events[-train],predict(kcweather.qda_train,kcweather[-train,])$class)
+   accuracy[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+   errlin[k] = (neval-sum(diag(tablin)))/neval
+ }
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9323913> cat("Precision :  ",mean(precision))
Precision :   0.981388> cat("Recall: ",mean(recall))
Recall:  0.9318323>
> 
```

The Accuracy is 0.93, Precision is 0.98, Recall is 0.931

KNN:

```
> library(class)
> data<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather <- subset(data,Events=="Snow"|Events=="Rain")
> kcweather$Events<-as.character(kcweather$Events)
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
> n=226
> nt=180
> neval=n-nt
> rep=100
> errlin=dim(rep)
> accuracy3=dim(rep)
> precision3=dim(rep)
> recall3=dim(rep)
> accuracy10=dim(rep)
> precision10=dim(rep)
> recall10=dim(rep)
> for (k in 1:rep) {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,1:8]
+   kcweather.Test = kcweather[-Tkcweather,1:8]
+   kcweather.trainLabels <- kcweather[Tkcweather,9]
+   kcweather.testLabels <- kcweather[-Tkcweather,9]
```

```
+   kcweather.knn3 = knn(kcweather.Train,kcweather.Test,kcweather.trainLabels,k=3)
+   kcweather.knn10 = knn(kcweather.Train,kcweather.Test,kcweather.trainLabels,k=10)
+   tablin=table(kcweather.knn3,kcweather.testLabels)
+   tablin10=table(kcweather.knn10,kcweather.testLabels)
+   accuracy3[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision3[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall3[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+   accuracy10[k] = (tablin10[1,1]+tablin10[2,2])/(sum(tablin10))
+   precision10[k] = (tablin10[1,1])/(tablin10[1,1]+tablin10[2,1])
+   recall10[k]=(tablin10[1,1])/(tablin10[1,1]+tablin10[1,2])
+ }
> cat('accuracy-k=3',mean(accuracy3))
accuracy-k=3 0.8708696> cat('precision-k=3',mean(precision3))
precision-k=3 0.9282271> cat('recall-k=3 ',mean(recall3))
recall-k=3  0.9091801> cat('accuracy-k=10 ',mean(accuracy10))
accuracy-k=10  0.8358696> cat('precision-k=10',mean(precision10))
precision-k=10 0.953431> cat('recall-k=10 ',mean(recall10))
recall-k=10  0.8532142>
```

accuracy-(k=3) is 0.873, precision-(k=3) is 0.927, recall-(k=3) is 0.914
accuracy-(k=10) is 0.833, precision-(k=10) is 0.9516677, recall-(k=10) is 0.8541957

Result:

1. After all these calculations, I infer that Logistic Regression has a good accuracy for the given data set with low precision.
2. There is a tradeoff between precision and accuracy.

3. We also have QDA which has high accuracy, precision and recall values relatively. Based on the requirement we should choose the model.

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

a.

```
recall-k=10   0.8532142> model=glm(Events~.,kcweather.train,family="binomial")
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(model)

Call:
glm(formula = Events ~ ., family = "binomial", data = kcweather.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.53173   0.00000   0.00001   0.00291   2.61588

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          2.915e+02  1.680e+02   1.735   0.0828 .
Date                 1.099e-07  8.851e-08   1.242   0.2144
Temp.F              -2.667e-01  3.753e-01  -0.711   0.4773
Dew_Point.F          7.892e-01  4.900e-01   1.611   0.1072
Humidity.percentage -2.819e-01  2.014e-01  -1.400   0.1616
Sea_Level_Press.in  -9.418e+00  5.772e+00  -1.632   0.1028
Visibility.mi        3.359e-01  8.015e-01   0.419   0.6751
Wind.mph            -2.577e-01  2.019e-01  -1.276   0.2019
Precip.in            1.866e+02  9.796e+01   1.905   0.0568 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.694  on 179  degrees of freedom
Residual deviance:  21.298  on 171  degrees of freedom
AIC: 39.298

Number of Fisher Scoring iterations: 12
```

Accuracy: 0.9628261
Precision: 0.8980501
Recall: 0.9398147

b.

```
> model=glm(Events~.- Visibility.mi,kcweather.train,family="binomial")
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(model)

Call:
glm(formula = Events ~ . - Visibility.mi, family = "binomial",
    data = kcweather.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.47239   0.00000   0.00001   0.00413   2.68908

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          2.773e+02  1.521e+02   1.823   0.0684 .
Date                 1.257e-07  8.089e-08   1.554   0.1202
Temp.F              -2.944e-01  3.620e-01  -0.813   0.4161
Dew_Point.F          8.255e-01  4.789e-01   1.724   0.0848 .
Humidity.percentage -3.135e-01  1.852e-01  -1.693   0.0905 .
Sea_Level_Press.in  -8.789e+00  5.118e+00  -1.717   0.0859 .
Wind.mph            -2.152e-01  1.707e-01  -1.261   0.2074
Precip.in            1.715e+02  9.083e+01   1.889   0.0589 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.694  on 179  degrees of freedom
Residual deviance:  21.484  on 172  degrees of freedom
AIC: 37.484

Number of Fisher Scoring iterations: 12
```

c.

```
> model=glm(formula = Events ~ . - Temp.F, family = "binomial", data = kcweather.train)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(model)

Call:
glm(formula = Events ~ . - Temp.F, family = "binomial", data = kcweather.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.41710   0.00000   0.00002   0.00352   2.44933

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          3.329e+02  1.801e+02   1.849  0.06452 .
Date                 1.169e-07  8.930e-08   1.309  0.19043
Dew_Point.F          4.756e-01  1.647e-01   2.887  0.00389 **
Humidity.percentage -1.573e-01  9.332e-02  -1.685  0.09196 .
Sea_Level_Press.in  -1.114e+01  6.061e+00  -1.838  0.06608 .
Visibility.mi        4.383e-01  7.634e-01   0.574  0.56586
Wind.mph            -2.958e-01  1.984e-01  -1.491  0.13589
Precip.in            1.823e+02  1.002e+02   1.818  0.06906 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.694  on 179  degrees of freedom
Residual deviance:  21.809  on 172  degrees of freedom
AIC: 37.809

Number of Fisher Scoring iterations: 12
```

d.

```
> model=glm(formula = Events ~ . - Precip.in, family = "binomial", data = kcweather.train)
> summary(model)

Call:
glm(formula = Events ~ . - Precip.in, family = "binomial", data = kcweather.train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.39762   0.00000   0.00039   0.01719   2.94182

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         2.499e+02  1.417e+02   1.763   0.0779 .
Date                1.488e-07  9.051e-08   1.644   0.1003
Temp.F             -2.125e-01  3.435e-01  -0.619   0.5361
Dew_Point.F         7.441e-01  4.400e-01   1.691   0.0908 .
Humidity.percentage -2.487e-01  1.812e-01  -1.372   0.1699
Sea_Level_Press.in -8.000e+00  4.829e+00  -1.657   0.0976 .
Visibility.mi      -2.272e-01  5.301e-01  -0.429   0.6682
Wind.mph           -1.500e-01  1.674e-01  -0.896   0.3702
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 190.694  on 179  degrees of freedom
Residual deviance:  24.558  on 172  degrees of freedom
AIC: 40.558

Number of Fisher Scoring iterations: 10
```

From all the above summaries obtained from different predictors and also using all predictors, I could see that visibility and temperature are not significant predictors but precipitation is slightly important as we could see that AIC value is higher when all predictors are used but relatively high for precipitation. So, it is a good model.

2. 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

LDA:

```
> data<-read.csv(file="G:Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",header=T,sep=",")
> kcweather<-subset(data,Events=="Snow"|Events=="Rain")
> kcweather$Events<-as.character(kcweather$Events)
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
> n=366
> nt=290
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision=dim(rep)
> recall=dim(rep)
> for (k in 1:rep) {
+   train=sample(1:n,nt)
+   kcweather.lda_train = lda(Events~.,kcweather[train,])
+   tablin=table(kcweather$Events[-train],predict(kcweather.lda_train,kcweather[-train,])$class)
+   accuracy[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+ }
> cat("Accuracy:",mean(accuracy))
Accuracy: 0.9333495> cat("Precision :  ",mean(precision))
Precision :   0.9597941> cat("Recall: ",mean(recall))
Recall:  0.9539534>
```

QDA:

```
> kcweather$Events<-as.character(kcweather$Events)
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
> n=366
> nt=290
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision=dim(rep)
> recall=dim(rep)
> for (k in 1:rep) {
+   train=sample(1:n,nt)
+   kcweather.qda_train = qda(Events~.,kcweather[train,])
+   tablin=table(kcweather$Events[-train],predict(kcweather.qda_train,kcweather[-train,])$class)
+   accuracy[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+ }
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9235978> cat("Precision :  ",mean(precision))
Precision :   0.9801153> cat("Recall: ",mean(recall))
Recall:  0.9218103> |
```

KNN:

```
> kcweather<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",header=T,sep=",")
> kcweather$Date=as.integer(gsub("-", "",kcweather$Date))
> kcweather$Events<-as.character(kcweather$Events)
> n=366
> nt=290
> neval=n-nt
> rep=100
> errlin=dim(rep)
> accuracy3=dim(rep)
> precision3=dim(rep)
> recall3=dim(rep)
> accuracy10=dim(rep)
> precision10=dim(rep)
> recall10=dim(rep)
> for (k in 1:rep) {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,1:8]
+   kcweather.Test = kcweather[-Tkcweather,1:8]
+   kcweather.trainLabels <- kcweather[Tkcweather,9]
+   kcweather.testLabels <- kcweather[-Tkcweather,9]
+   kcweather.knn3 = knn(kcweather.Train,kcweather.Test,kcweather.trainLabels,k=3)
+   kcweather.knn10 = knn(kcweather.Train,kcweather.Test,kcweather.trainLabels,k=10)
+   tablin=table(kcweather.knn3,kcweather.testLabels)
+   tablin10=table(kcweather.knn10,kcweather.testLabels)
+   accuracy3[k] = (tablin[1,1]+tablin[2,2])/(sum(tablin))
+   precision3[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1])
+   recall3[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2])
+   accuracy10[k] = (tablin10[1,1]+tablin10[2,2])/(sum(tablin10))
+   precision10[k] = (tablin10[1,1])/(tablin10[1,1]+tablin10[2,1])
+   recall10[k]=(tablin10[1,1])/(tablin10[1,1]+tablin10[1,2])
+ }
> cat('accuracy-k=3 ',mean(accuracy3))
accuracy-k=3  0.5676316> cat('precision-k=3',mean(precision3))
precision-k=3 0.6831014> cat('recall-k=3 ',mean(recall3))
recall-k=3  0.7161054> cat('accuracy-k=10 ',mean(accuracy10))
accuracy-k=10  0.5496053> cat('precision-k=10',mean(precision10))
precision-k=10 0.6830243> cat('recall-k=10 ',mean(recall10))
recall-k=10  0.6863537> |
```

Summary:

1.From the above calculations we could see that QDA has a good accuracy with high precision and if precision is our prime factor for the modeling then we should go with the QDA model.

2. But LDA has the highest accuracy and with a bit less precision than QDA so if the model needs high accuracy then LDA is better. There is always a trade-off