<u>Question 1:</u>

A.Consider first the subset that consists only of Rain and Snow. There are 226 entries with these two categories.
1. 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.

<u>Solution:</u>

Logistic Regression:
<u>Source code:</u>

```
data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-
AS2\\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)

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])

}
cat("Accuracy: ",mean(accuracy))
cat("Precision :  ",mean(precision))
cat("Recall: ",mean(recall))
```

Output:

```
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9576087
> cat("Precision :  ",mean(precision))
Precision :   0.8902743
> cat("Recall: ",mean(recall))
Recall:  0.9285005
```

Note (Calculation of confusion matrix for logistic regression):

In logistic regression, for construction of confusion matrix, I have manually taken the **threshold value of probability = 0.5** as shown below.

tablin=table(Actualvalue=kcweather.test$Events,Predictedvalue=res>0.5)

After the construction of confusion matrix, this is used for calculation of the metrics accuracy, precision and recall

LDA:

Source code:

```
  library(MASS)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-
AS2\\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)
  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])
   errlin[k] = (neval-sum(diag(tablin)))/neval
 }
 cat("Accuracy: ",mean(accuracy))
 cat("Precision :  ",mean(precision))
 cat("Recall: ",mean(recall))
```

Output:

```
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9282609
> cat("Precision :   ",mean(precision))
Precision :   0.9545875
> cat("Recall: ",mean(recall))
Recall:  0.9543134
```

QDA:

Source code:

```
library(MASS)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-
AS2\\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)
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))
cat("Precision :  ",mean(precision))
cat("Recall: ",mean(recall))
```

<u>Output:</u>

```
>
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.925
> cat("Precision :  ",mean(precision))
Precision :   0.9746241
> cat("Recall: ",mean(recall))
Recall:  0.9267973
```

<u>KNN:</u>

<u>Source code:</u>

```
library(class)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-
AS2\\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))
cat('precision-k=3',mean(precision3))
cat('recall-k=3 ',mean(recall3))
cat('accuracy-k=10 ',mean(accuracy10))
cat('precision-k=10',mean(precision10))
cat('recall-k=10 ',mean(recall10))
```

Output:

```
> cat('accuracy-k=3 ',mean(accuracy3))
accuracy-k=3  0.842619
> cat('precision-k=3',mean(precision3))
precision-k=3 0.9142716
> cat('recall-k=3 ',mean(recall3))
recall-k=3  0.8889864
> cat('accuracy-k=10 ',mean(accuracy10))
accuracy-k=10  0.785
> cat('precision-k=10',mean(precision10))
precision-k=10 0.9364937
> cat('recall-k=10 ',mean(recall10))
recall-k=10  0.8171462
>
```

Summary:

Model	**Accuracy**	**Precision**	**Recall**
Logistic Regression	0.9576087	0.8902743	0.9285005
LDA	0.9252609	0.9545875	0.9243134
QDA	0.928	0.9746241	0.9567973
KNN(k=3)	0.842619	0.9142716	0.8889864
KNN(k=10)	0.785	0.9364937	0.8171462

Results Analysis (Text Summarization):

1. As you can see from the summary, the Logistic Regression model has high accuracy for the kc weather dataset and at the same time a less precision value and a recall value.

a) If the business requires to build a model where precision is of not that importance and accuracy, recall need to be high, then logistic regression model is the best fit.

b) If the precision is of main consideration, then Logistic regression is not the best suite model.

2. QDA model also has high accuracy(less value compared to Logistic regression), but it has high precision and recall values compared to Logistic regression and all other models mostly considered.

a) So QDA is the best fit model for a business where Precision is of utmost importance and accuracy, recall are also need to be high values.

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) Let us use the logistic regression model for eliminating predictors

a) model=glm(Events~.,kcweather.train,family="binomial")
summary(model)

WEATHER

Call:

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

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.43533   0.00000   0.00001   0.00232   3.02241

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.636e+02  1.450e+02   1.128   0.2592
Date                 1.193e-07  8.144e-08   1.465   0.1430
Temp.F              -3.415e-01  3.795e-01  -0.900   0.3681
Dew_Point.F          9.118e-01  5.011e-01   1.820   0.0688 .
Humidity.percentage -3.113e-01  2.159e-01  -1.442   0.1494
Sea_Level_Press.in  -5.010e+00  4.948e+00  -1.012   0.3113
Visibility.mi       -1.158e-01  6.217e-01  -0.186   0.8522
Wind.mph            -1.827e-01  1.939e-01  -0.942   0.3462
Precip.in            1.442e+02  1.177e+02   1.225   0.2204
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 185.553  on 179  degrees of freedom
Residual deviance:  20.994  on 171  degrees of freedom
AIC: 38.994

Number of Fisher Scoring iterations: 12

> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9491304
> cat("Precision :  ",mean(precision))
Precision :   0.8742776>
>
> cat("Recall: ",mean(recall))
Recall:  0.9043733>
```

b) model=glm(Events~.- Visibility.mi,kcweather.train,family="binomial")

VISIBILITY

Call:

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

```
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.956087>
> cat("Precision :  ",mean(precision))
Precision :   0.8787629
> cat("Recall: ",mean(recall))
Recall:  0.9289836
>
> summary(model)

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.48354   0.00000   0.00000   0.00078   2.89797

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.443e+02  1.496e+02   0.964   0.3349
Date                 7.384e-08  7.322e-08   1.008   0.3132
Temp.F              -3.732e-01  3.829e-01  -0.975   0.3297
Dew_Point.F          1.020e+00  5.375e-01   1.897   0.0578 .
Humidity.percentage -3.487e-01  2.071e-01  -1.684   0.0922 .
Sea_Level_Press.in  -4.327e+00  5.009e+00  -0.864   0.3876
Wind.mph            -2.327e-01  1.791e-01  -1.299   0.1940
Precip.in            1.639e+02  1.220e+02   1.344   0.1790
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 197.926  on 179  degrees of freedom
Residual deviance:  20.526  on 172  degrees of freedom
AIC: 36.526

Number of Fisher Scoring iterations: 13
```

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

TEMPERATURE

Call:

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

```
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9515217
> cat("Precision :  ",mean(precision))
Precision :   0.8701204
> cat("Recall: ",mean(recall))
Recall:  0.9189152
>
> summary(model)

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

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.51223   0.00000   0.00002   0.00410   2.50219  

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)   
(Intercept)          3.106e+02  1.766e+02   1.759  0.07861 . 
Date                 9.246e-08  8.880e-08   1.041  0.29777   
Dew_Point.F          4.553e-01  1.522e-01   2.991  0.00278 **
Humidity.percentage -1.178e-01  8.180e-02  -1.441  0.14970   
Sea_Level_Press.in  -1.049e+01  5.966e+00  -1.759  0.07857 . 
Visibility.mi        6.305e-01  7.199e-01   0.876  0.38114   
Wind.mph            -4.088e-01  2.031e-01  -2.013  0.04414 * 
Precip.in            1.831e+02  1.099e+02   1.666  0.09572 . 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 200.214  on 179  degrees of freedom
Residual deviance:  21.791  on 172  degrees of freedom
AIC: 37.791

Number of Fisher Scoring iterations: 13
```

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

PRECIPITATION

Call:

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

```
> cat("Accuracy: ",mean(accuracy))
Accuracy:  0.9606522
> cat("Precision :  ",mean(precision))
Precision :   0.9094401
> cat("Recall: ",mean(recall))
Recall:  0.9211164
>
> summary(model)

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

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.57130   0.00000   0.00051   0.02278   2.94780

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          2.388e+02  1.459e+02   1.636    0.102
Date                 1.047e-07  8.028e-08   1.305    0.192
Temp.F              -1.950e-01  3.696e-01  -0.528    0.598
Dew_Point.F          7.178e-01  4.601e-01   1.560    0.119
Humidity.percentage -1.880e-01  1.956e-01  -0.961    0.337
Sea_Level_Press.in  -7.801e+00  4.934e+00  -1.581    0.114
Visibility.mi        2.574e-03  4.932e-01   0.005    0.996
Wind.mph            -1.842e-01  1.571e-01  -1.172    0.241

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 182.883  on 179  degrees of freedom
Residual deviance:  26.736  on 172  degrees of freedom
AIC: 42.736

Number of Fisher Scoring iterations: 10
```

	Residual deviance	AIC	Accuracy	Precision	Recall
All predictors	20.994	38.994	0.9491304	0.8742776	0.9043733
All Predictors-Visibility	20.526	36.526	0.956087	0.8787629	0.9289836
All Predictors-Temperature	21.791	37.791	0.9515217	0.8701204	0.9189152
All Predictors-Precipitation	26.736	42.736	0.960622	0.9094401	0.9211164

In the case of excluding the predictors Visibility, Temperature and Precipitation all the cases Residual deviance and AIC values are lesser when compared considering all the predictors. But the values of accuracy, precision and recall are slightly lesser.

Hence, **Visibility, Temperature and precipitation are not significant predictors** for the model as the **AIC and residual value decreases** when compared to model where all predictors are included for building model.

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.

a)

LDA:

Source code:

```
library(MASS)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-AS2\\kc_weather_srt.csv",head=T,sep=",")

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))

cat("Precision :  ",mean(precision))

cat("Recall: ",mean(recall))
```

Output:

```
> library(MASS)
> data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-AS2\\kc_weather_srt.
=T,sep=",")
> 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$Events[-train],predict(kcweather.lda_train,kcweat
n,])$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:  0.9238136
> cat("Precision :  ",mean(precision))
Precision :   0.8292378
> cat("Recall: ",mean(recall))
Recall:  0.8236089
```

QDA:

Source code:

```
library(MASS)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-AS2\\kc_weather_srt.csv",head=T,sep=",")

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))

cat("Precision :  ",mean(precision))

cat("Recall: ",mean(recall))
```

Output:

```
> library(MASS)
> data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-AS2\\kc_wea
=T,sep=",")
> 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$Events[-train],predict(kcweather.qda_tra
n,])$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:  0.9284047
> cat("Precision: ",mean(precision))
Precision:  0.7934406
> cat("Recall: ",mean(recall))
Recall:  0.9310367
```

KNN:

Source code:

```
library(class)

data<-read.csv(file="C:\\Users\\sudheesha\\Desktop\\ISL-
AS2\\kc_weather_srt.csv",head=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))
cat('precision-k=3',mean(precision3))
cat('recall-k=3 ',mean(recall3))
cat('accuracy-k=10 ',mean(accuracy10))
cat('precision-k=10',mean(precision10))
cat('recall-k=10 ',mean(recall10))
```

Output:

```
> kcw$Date=as.integer(gsub("-", "",kcw$Date))
> kcw$Events<-as.character(kcw$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) {
+     Tkcw = sample(1:n,nt)
+     kcw.Train = kcw[Tkcw,1:8]
+     kcw.Test = kcw[-Tkcw,1:8]
+     kcw.trainLabels <- kcw[Tkcw,9]
+     kcw.testLabels <- kcw[-Tkcw,9]
+     kcw.knn3 = knn(kcw.Train,kcw.Test,kcw.trainLabels,k=3)
+     kcw.knn10 = knn(kcw.Train,kcw.Test,kcw.trainLabels,k=10)
+     tablin=table(kcw.knn3,kcw.testLabels)
+     tablin10=table(kcw.knn10,kcw.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  0.5596053> cat('precision-k=3',mean(precision3))
precision-k=3 0.677384> cat('recall-k=3 ',mean(recall3))
recall-k=3  0.7164928> cat('accuracy-k=10 ',mean(accuracy10))
accuracy-k=10  0.5410526> cat('precision-k=10',mean(precision10))
precision-k=10 0.6760336> cat('recall-k=10 ',mean(recall10))
recall-k=10  0.6876788
```

Summary:

Model	**Accuracy**	**Precision**	**Recall**
LDA	0.9238136	0.8292378	0.8236089
QDA	0.9284047	0.7934406	0.9310367
KNN(k=3)	0.5596053	0.677384	0.7164928
KNN(k=10)	0.5410526	0.6760336	0.6876788

Analysis (Text Summarization):

- The three metrics that we will be considering is the Accuracy, Precision and Recall
- QDA has the highest accuracy and the least precision and whereas the LDA has the least accuracy and highest precision
- When compared to these two models KNN has the less the values
- So , based on the metrics and their importance any of the two models can be considered
- According to me, LDA is the best fit to the model when compared to that of the QDA has the accuracy gives an accurate result when it is tested on the symmetric data sets only.