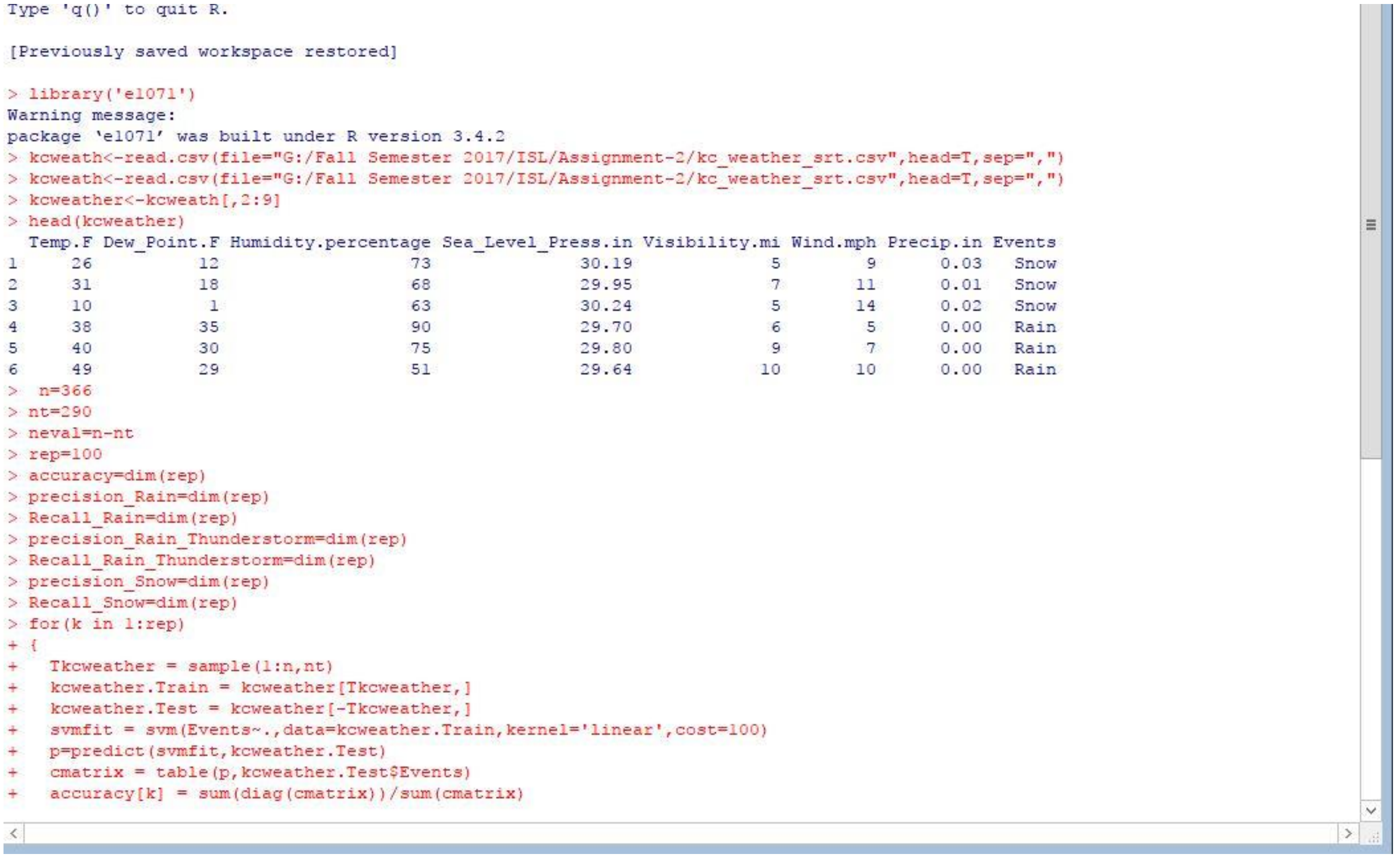
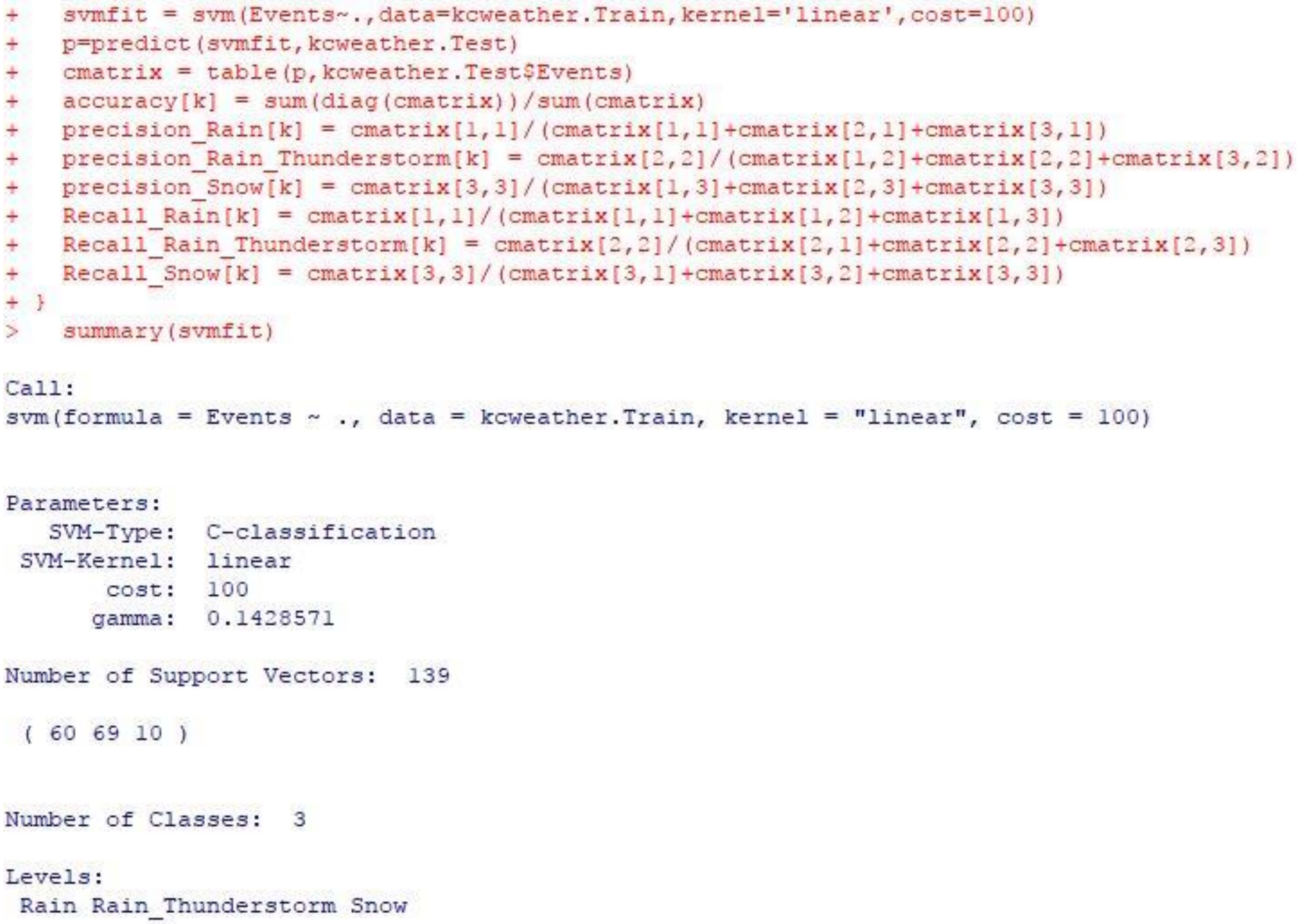
Assignment-3

3_1.

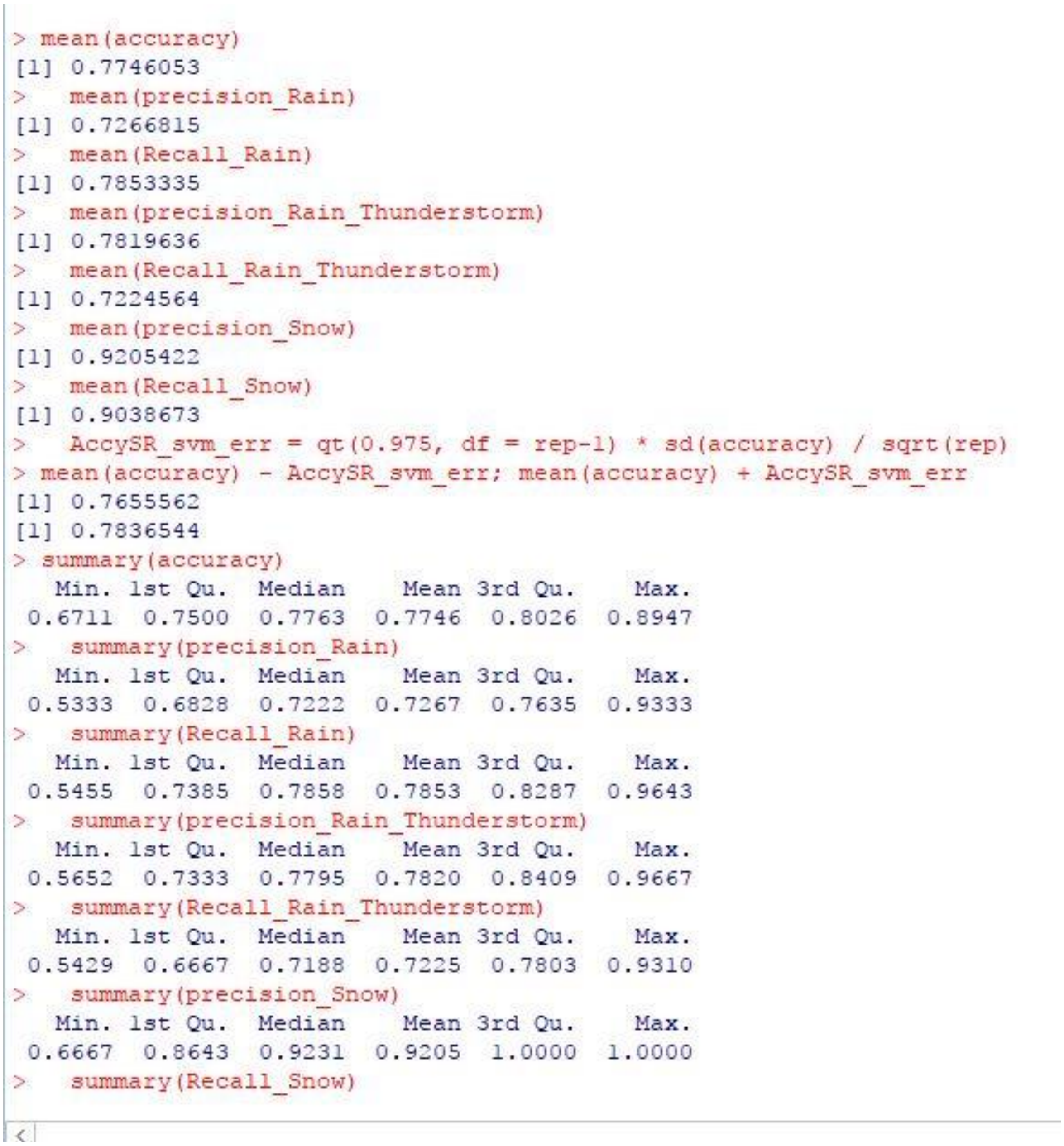
a. Implementing linear kernel model:



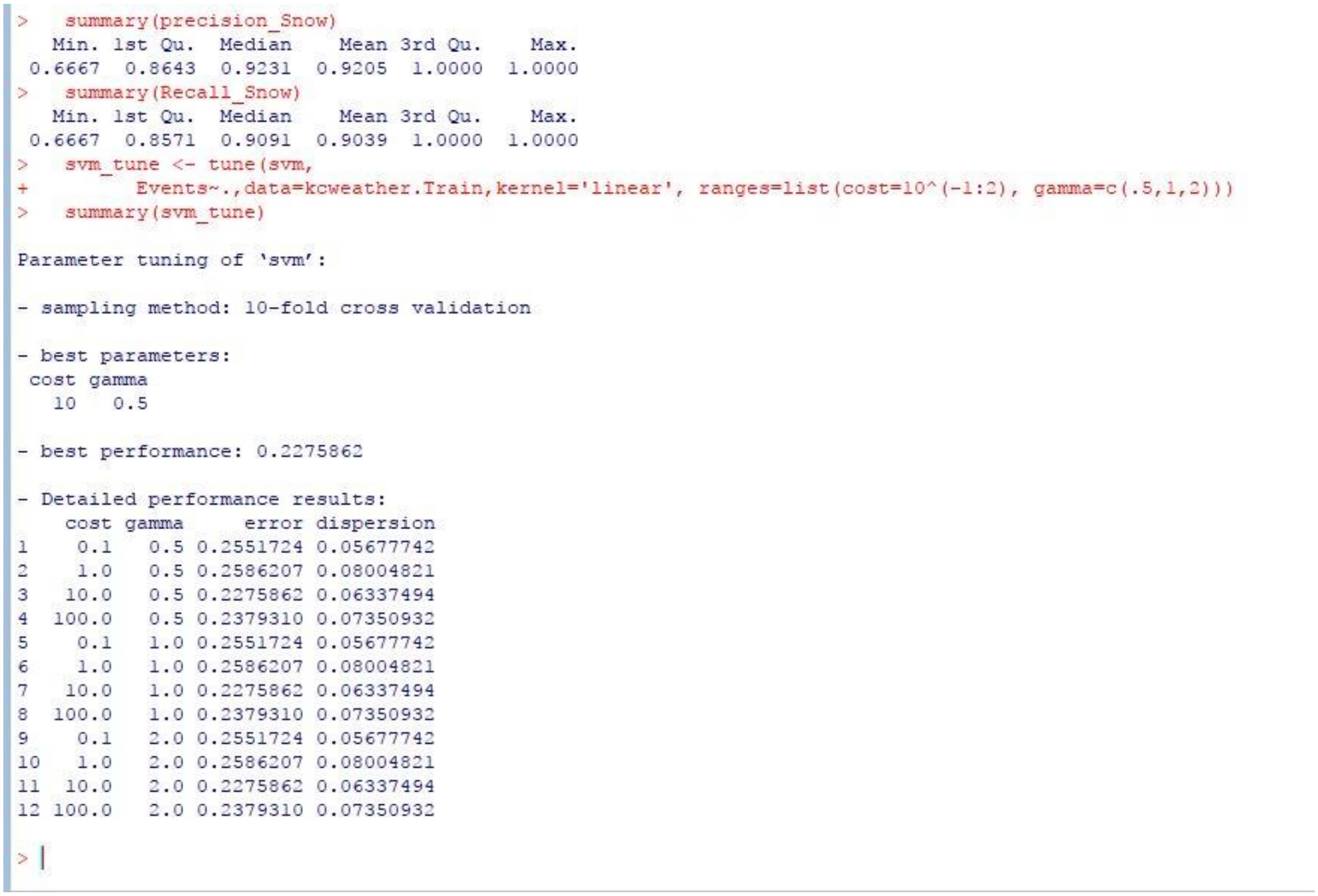
```
Type 'q()' to quit R.

[Previously saved workspace restored]

> library('e1071')
Warning message:
package ‘e1071’ was built under R version 3.4.2
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather<-kcweath[,2:9]
> head(kcweather)
  Temp.F Dew_Point.F Humidity.percentage Sea_Level_Press.in Visibility.mi Wind.mph Precip.in Events
1     26          12                  73              30.19             5        9      0.03   Snow
2     31          18                  68              29.95             7       11      0.01   Snow
3     10           1                  63              30.24             5       14      0.02   Snow
4     38          35                  90              29.70             6        5      0.00   Rain
5     40          30                  75              29.80             9        7      0.00   Rain
6     49          29                  51              29.64            10       10      0.00   Rain
>  n=366
> nt=290
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision_Rain=dim(rep)
> Recall_Rain=dim(rep)
> precision_Rain_Thunderstorm=dim(rep)
> Recall_Rain_Thunderstorm=dim(rep)
> precision_Snow=dim(rep)
> Recall_Snow=dim(rep)
> for(k in 1:rep)
+ {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,]
+   kcweather.Test = kcweather[-Tkcweather,]
+   svmfit = svm(Events~.,data=kcweather.Train,kernel='linear',cost=100)
+   p=predict(svmfit,kcweather.Test)
+   cmatrix = table(p,kcweather.Test$Events)
+   accuracy[k] = sum(diag(cmatrix))/sum(cmatrix)
```



```
+   svmfit = svm(Events~.,data=kcweather.Train,kernel='linear',cost=100)
+   p=predict(svmfit,kcweather.Test)
+   cmatrix = table(p,kcweather.Test$Events)
+   accuracy[k] = sum(diag(cmatrix))/sum(cmatrix)
+   precision_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[2,1]+cmatrix[3,1])
+   precision_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[1,2]+cmatrix[2,2]+cmatrix[3,2])
+   precision_Snow[k] = cmatrix[3,3]/(cmatrix[1,3]+cmatrix[2,3]+cmatrix[3,3])
+   Recall_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[1,2]+cmatrix[1,3])
+   Recall_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[2,1]+cmatrix[2,2]+cmatrix[2,3])
+   Recall_Snow[k] = cmatrix[3,3]/(cmatrix[3,1]+cmatrix[3,2]+cmatrix[3,3])
+ }
>   summary(svmfit)

Call:
svm(formula = Events ~ ., data = kcweather.Train, kernel = "linear", cost = 100)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  100
      gamma:  0.1428571

Number of Support Vectors:  139

 ( 60 69 10 )


Number of Classes:  3

Levels:
 Rain Rain_Thunderstorm Snow
```



```
> mean(accuracy)
[1] 0.7746053
>   mean(precision_Rain)
[1] 0.7266815
>   mean(Recall_Rain)
[1] 0.7853335
>   mean(precision_Rain_Thunderstorm)
[1] 0.7819636
>   mean(Recall_Rain_Thunderstorm)
[1] 0.7224564
>   mean(precision_Snow)
[1] 0.9205422
>   mean(Recall_Snow)
[1] 0.9038673
>   AccySR_svm_err = qt(0.975, df = rep-1) * sd(accuracy) / sqrt(rep)
> mean(accuracy) - AccySR_svm_err; mean(accuracy) + AccySR_svm_err
[1] 0.7655562
[1] 0.7836544
> summary(accuracy)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6711  0.7500  0.7763  0.7746  0.8026  0.8947 
>   summary(precision_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5333  0.6828  0.7222  0.7267  0.7635  0.9333 
>   summary(Recall_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5455  0.7385  0.7858  0.7853  0.8287  0.9643 
>   summary(precision_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5652  0.7333  0.7795  0.7820  0.8409  0.9667 
>   summary(Recall_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5429  0.6667  0.7188  0.7225  0.7803  0.9310 
>   summary(precision_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6667  0.8643  0.9231  0.9205  1.0000  1.0000 
>   summary(Recall_Snow)
```



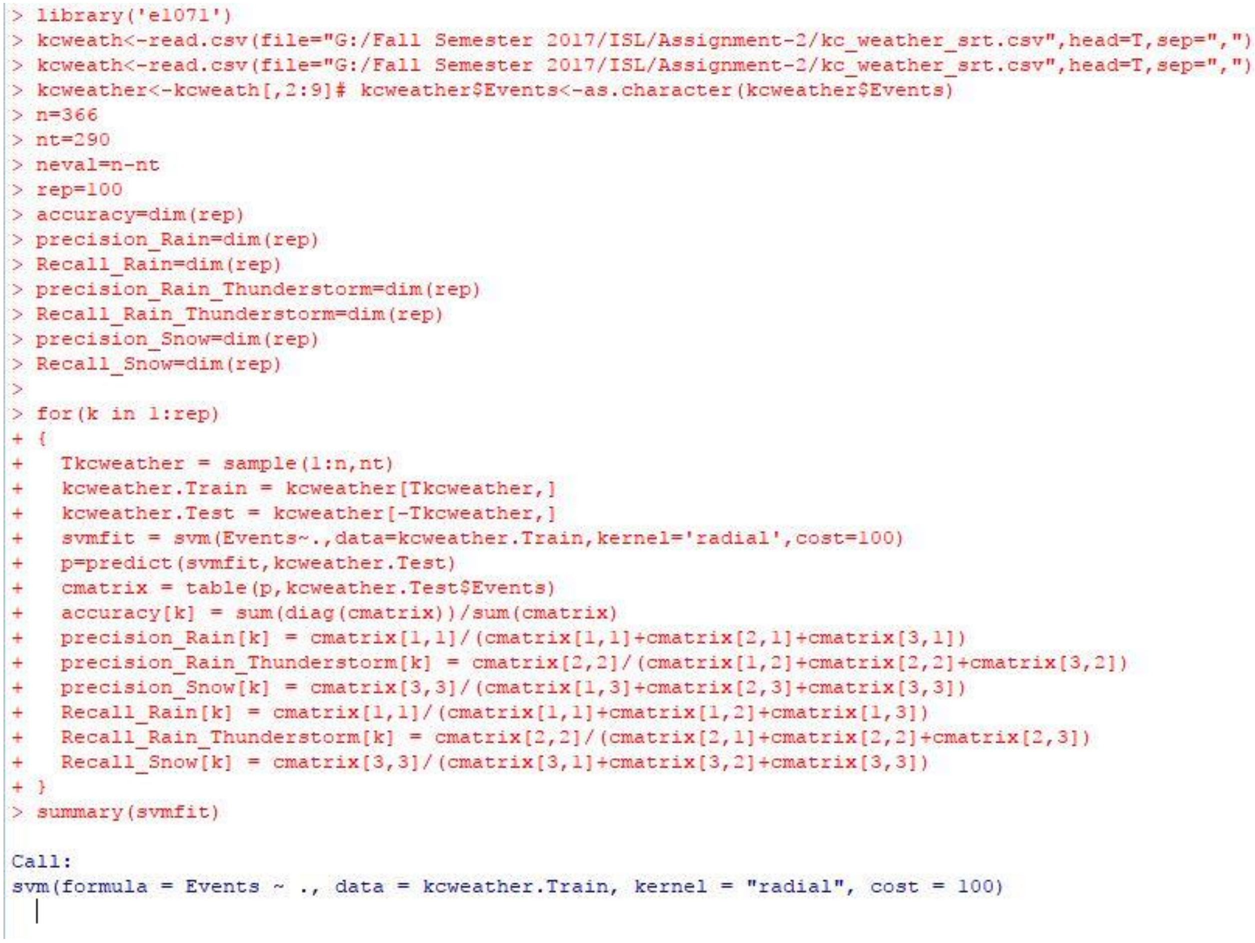
```
>   summary(precision_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6667  0.8643  0.9231  0.9205  1.0000  1.0000 
>   summary(Recall_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6667  0.8571  0.9091  0.9039  1.0000  1.0000 
>   svm_tune <- tune(svm,
+         Events~.,data=kcweather.Train,kernel='linear', ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))
>   summary(svm_tune)

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation 

- best parameters:
 cost gamma
   10   0.5

- best performance: 0.2275862 

- Detailed performance results:
    cost gamma     error dispersion
1    0.1   0.5 0.2551724 0.05677742
2    1.0   0.5 0.2586207 0.08004821
3   10.0   0.5 0.2275862 0.06337494
4  100.0   0.5 0.2379310 0.07350932
5    0.1   1.0 0.2551724 0.05677742
6    1.0   1.0 0.2586207 0.08004821
7   10.0   1.0 0.2275862 0.06337494
8  100.0   1.0 0.2379310 0.07350932
9    0.1   2.0 0.2551724 0.05677742
10   1.0   2.0 0.2586207 0.08004821
11  10.0   2.0 0.2275862 0.06337494
12 100.0   2.0 0.2379310 0.07350932

> 
```

Inference:

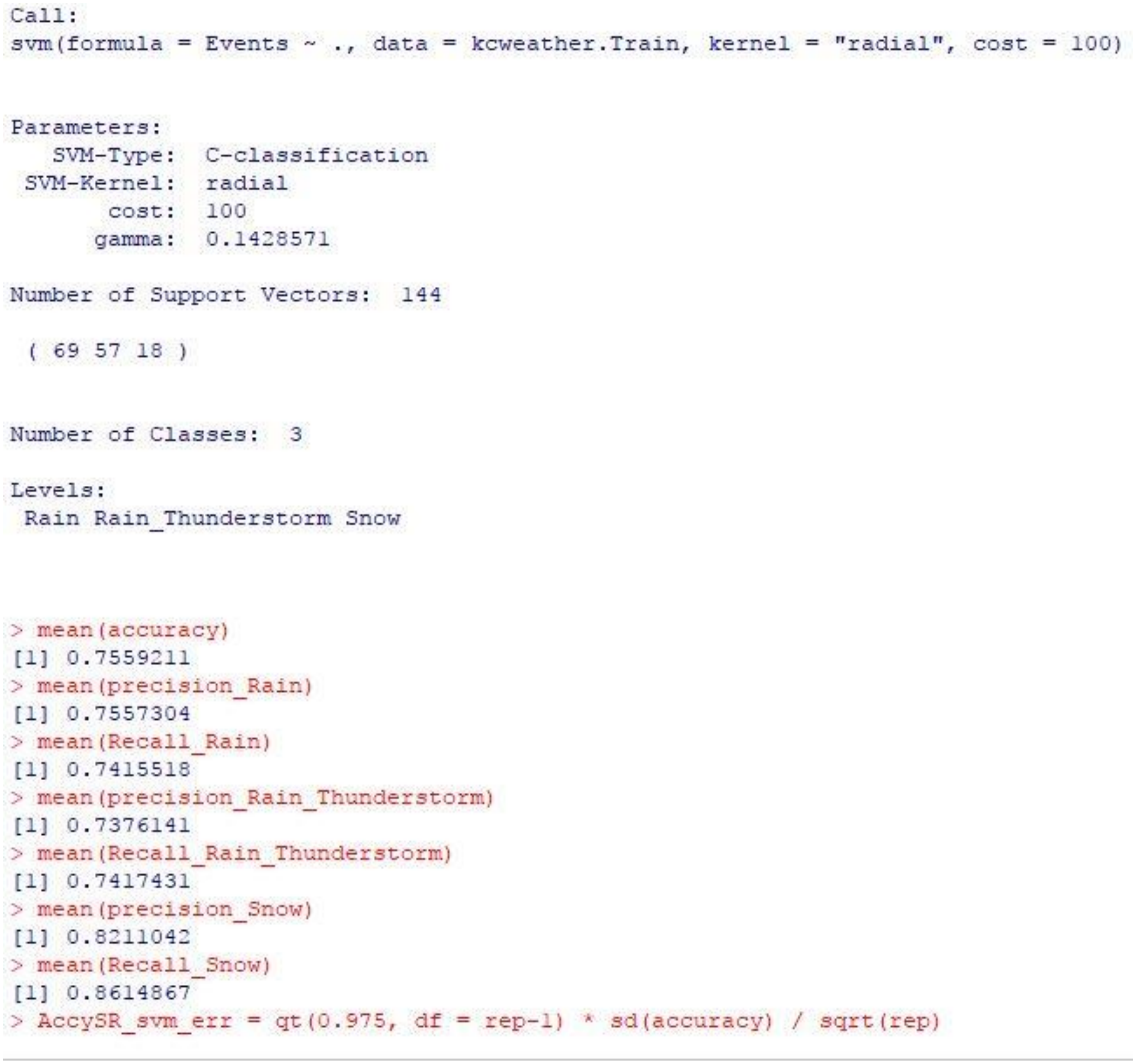
The total number of support vectors are 139 with three classes.

The svm tuning results shows the best parameters are with cost 10 and gamma=0.5

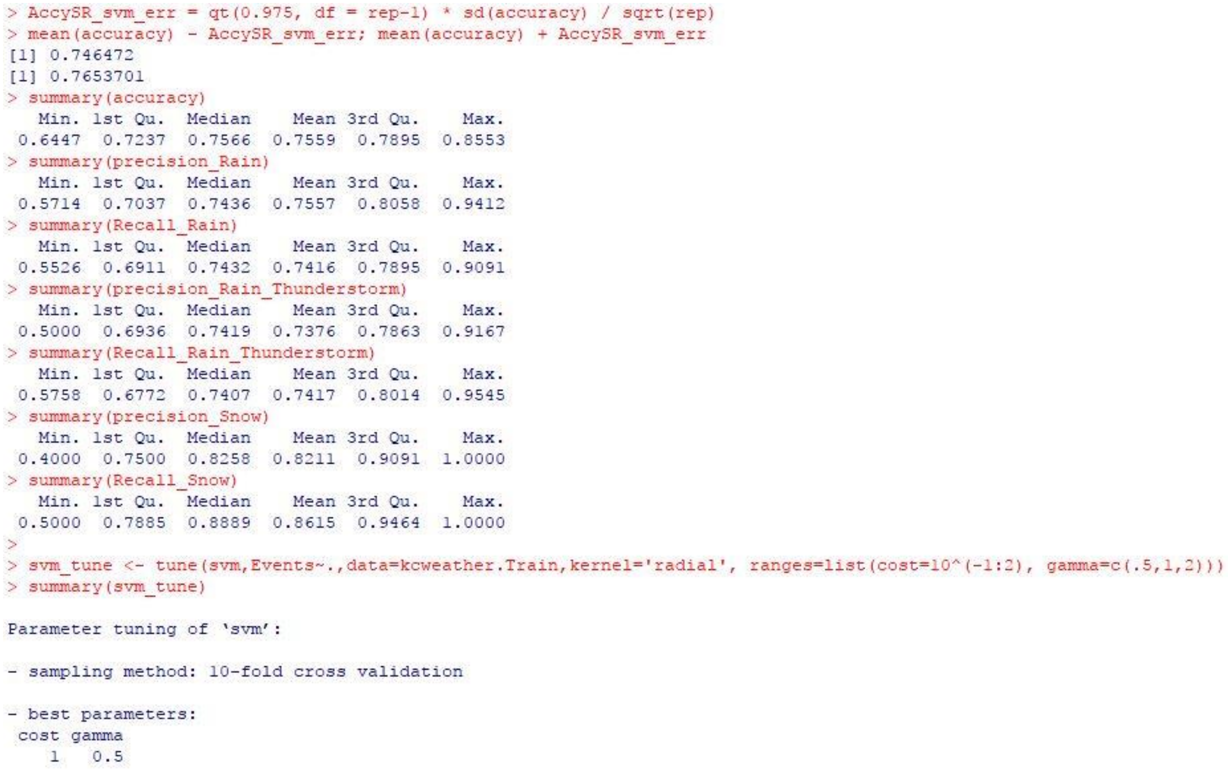
b. Implementing Radial Kernel:



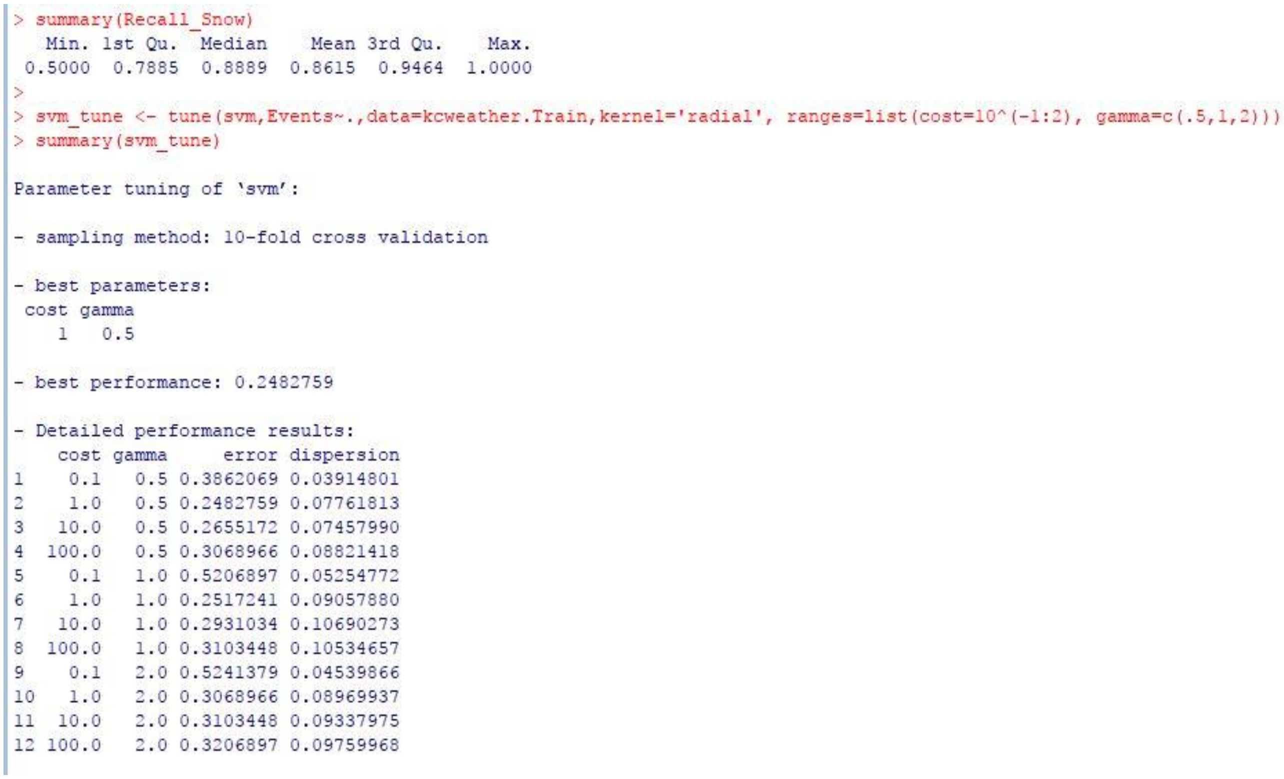
```
> library('e1071')
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather<-kcweath[,2:9]# kcweather$Events<-as.character(kcweather$Events)
> n=366
> nt=290
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision_Rain=dim(rep)
> Recall_Rain=dim(rep)
> precision_Rain_Thunderstorm=dim(rep)
> Recall_Rain_Thunderstorm=dim(rep)
> precision_Snow=dim(rep)
> Recall_Snow=dim(rep)
>
> for(k in 1:rep)
+ {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,]
+   kcweather.Test = kcweather[-Tkcweather,]
+   svmfit = svm(Events~.,data=kcweather.Train,kernel='radial',cost=100)
+   p=predict(svmfit,kcweather.Test)
+   cmatrix = table(p,kcweather.Test$Events)
+   accuracy[k] = sum(diag(cmatrix))/sum(cmatrix)
+   precision_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[2,1]+cmatrix[3,1])
+   precision_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[1,2]+cmatrix[2,2]+cmatrix[3,2])
+   precision_Snow[k] = cmatrix[3,3]/(cmatrix[1,3]+cmatrix[2,3]+cmatrix[3,3])
+   Recall_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[1,2]+cmatrix[1,3])
+   Recall_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[2,1]+cmatrix[2,2]+cmatrix[2,3])
+   Recall_Snow[k] = cmatrix[3,3]/(cmatrix[3,1]+cmatrix[3,2]+cmatrix[3,3])
+ }
> summary(svmfit)

Call:
svm(formula = Events ~ ., data = kcweather.Train, kernel = "radial", cost = 100)
```



```
Call:
svm(formula = Events ~ ., data = kcweather.Train, kernel = "radial", cost = 100)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  100
      gamma:  0.1428571

Number of Support Vectors:  144

 ( 69 57 18 )


Number of Classes:  3

Levels:
 Rain Rain_Thunderstorm Snow



> mean(accuracy)
[1] 0.7559211
> mean(precision_Rain)
[1] 0.7557304
> mean(Recall_Rain)
[1] 0.7415518
> mean(precision_Rain_Thunderstorm)
[1] 0.7376141
> mean(Recall_Rain_Thunderstorm)
[1] 0.7417431
> mean(precision_Snow)
[1] 0.8211042
> mean(Recall_Snow)
[1] 0.8614867
> AccySR_svm_err = qt(0.975, df = rep-1) * sd(accuracy) / sqrt(rep)
```



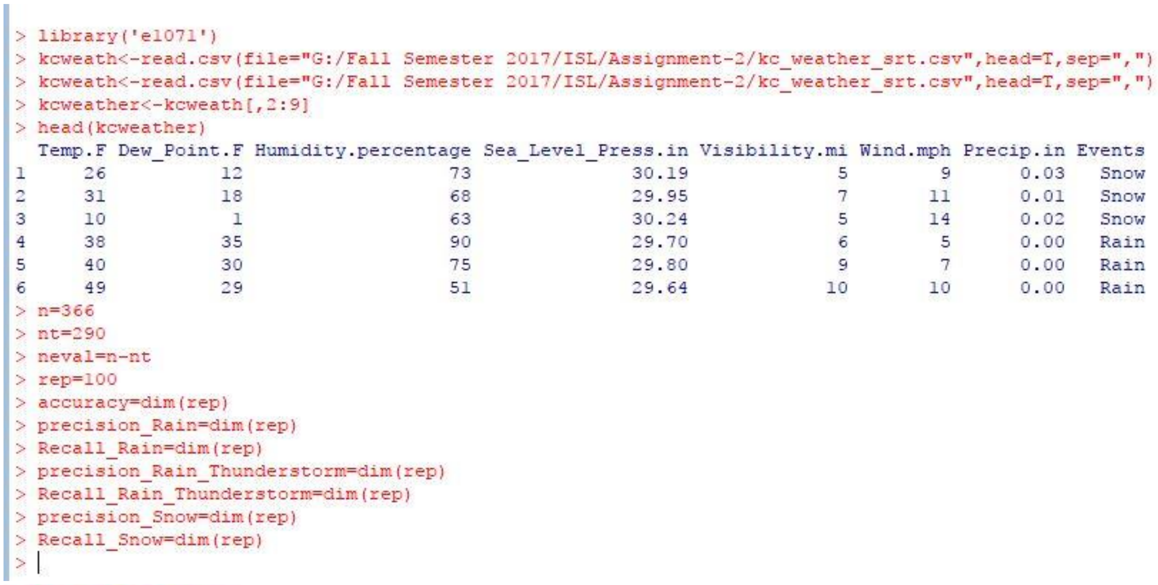
```
> AccySR_svm_err = qt(0.975, df = rep-1) * sd(accuracy) / sqrt(rep)
> mean(accuracy) - AccySR_svm_err; mean(accuracy) + AccySR_svm_err
[1] 0.746472
[1] 0.7653701
> summary(accuracy)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6447  0.7237  0.7566  0.7559  0.7895  0.8553
> summary(precision_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5714  0.7037  0.7436  0.7557  0.8058  0.9412
> summary(Recall_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5526  0.6911  0.7432  0.7416  0.7895  0.9091
> summary(precision_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5000  0.6936  0.7419  0.7376  0.7863  0.9167
> summary(Recall_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5758  0.6772  0.7407  0.7417  0.8014  0.9545
> summary(precision_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4000  0.7500  0.8258  0.8211  0.9091  1.0000
> summary(Recall_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5000  0.7885  0.8889  0.8615  0.9464  1.0000
>
> svm_tune <- tune(svm,Events~.,data=kcweather.Train,kernel='radial', ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))
> summary(svm_tune)

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
    1   0.5
```



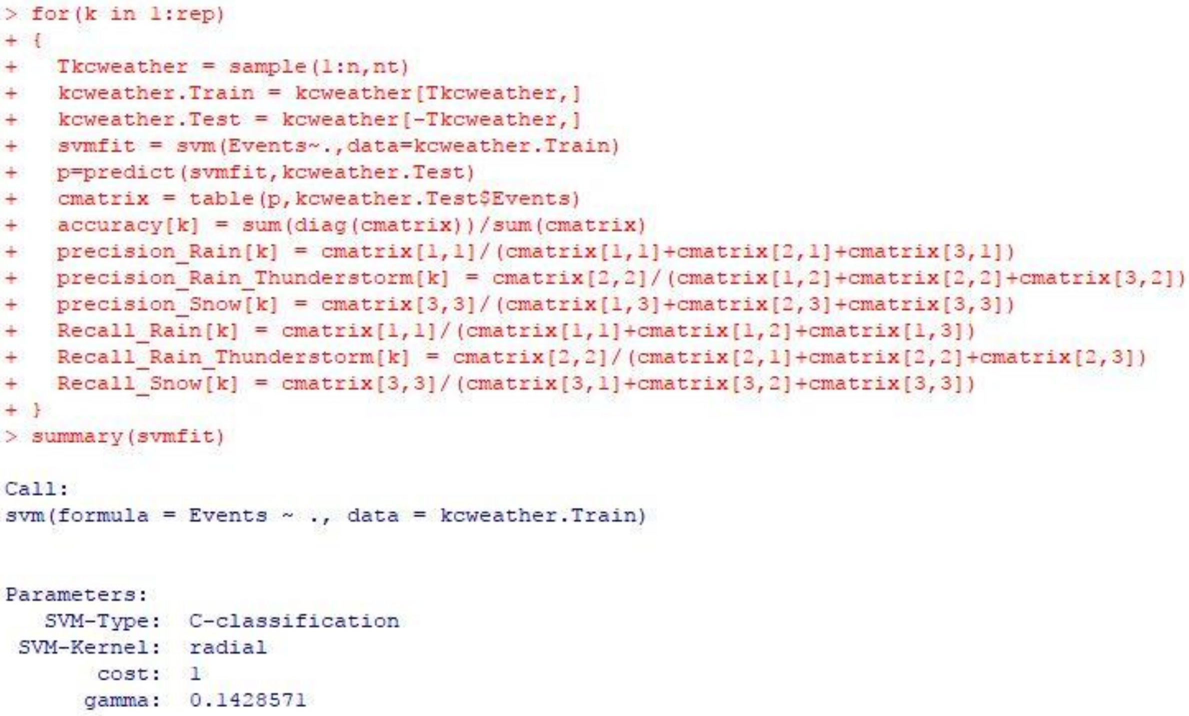
```
> summary(Recall_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5000  0.7885  0.8889  0.8615  0.9464  1.0000
>
> svm_tune <- tune(svm,Events~.,data=kcweather.Train,kernel='radial', ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))
> summary(svm_tune)

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
    1   0.5

- best performance: 0.2482759

- Detailed performance results:
    cost gamma     error dispersion
1    0.1   0.5 0.3862069 0.03914801
2    1.0   0.5 0.2482759 0.07761813
3   10.0   0.5 0.2655172 0.07457990
4  100.0   0.5 0.3068966 0.08821418
5    0.1   1.0 0.5206897 0.05254772
6    1.0   1.0 0.2517241 0.09057880
7   10.0   1.0 0.2931034 0.10690273
8  100.0   1.0 0.3103448 0.10534657
9    0.1   2.0 0.5241379 0.04539866
10   1.0   2.0 0.3068966 0.08969937
11  10.0   2.0 0.3103448 0.09337975
12 100.0   2.0 0.3206897 0.09759968
```

The number of support vectors are 144 with the best parameters cost=1, gamma=0.5

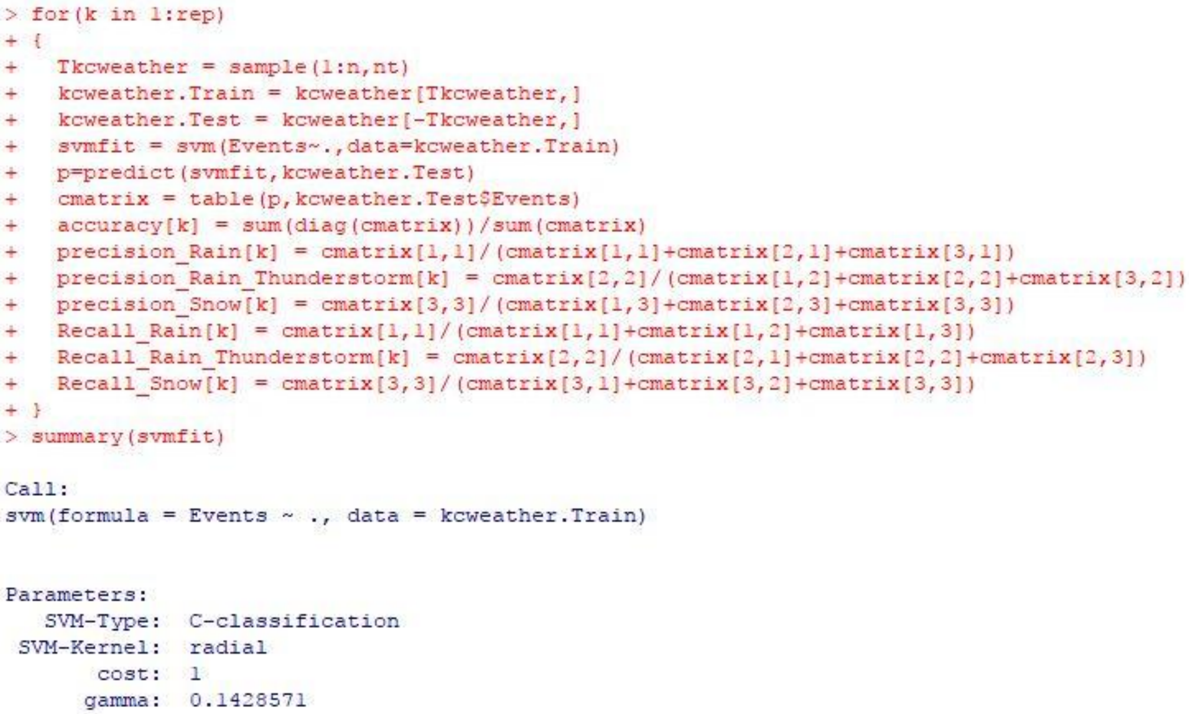
c. Let the svm choose the kernel:



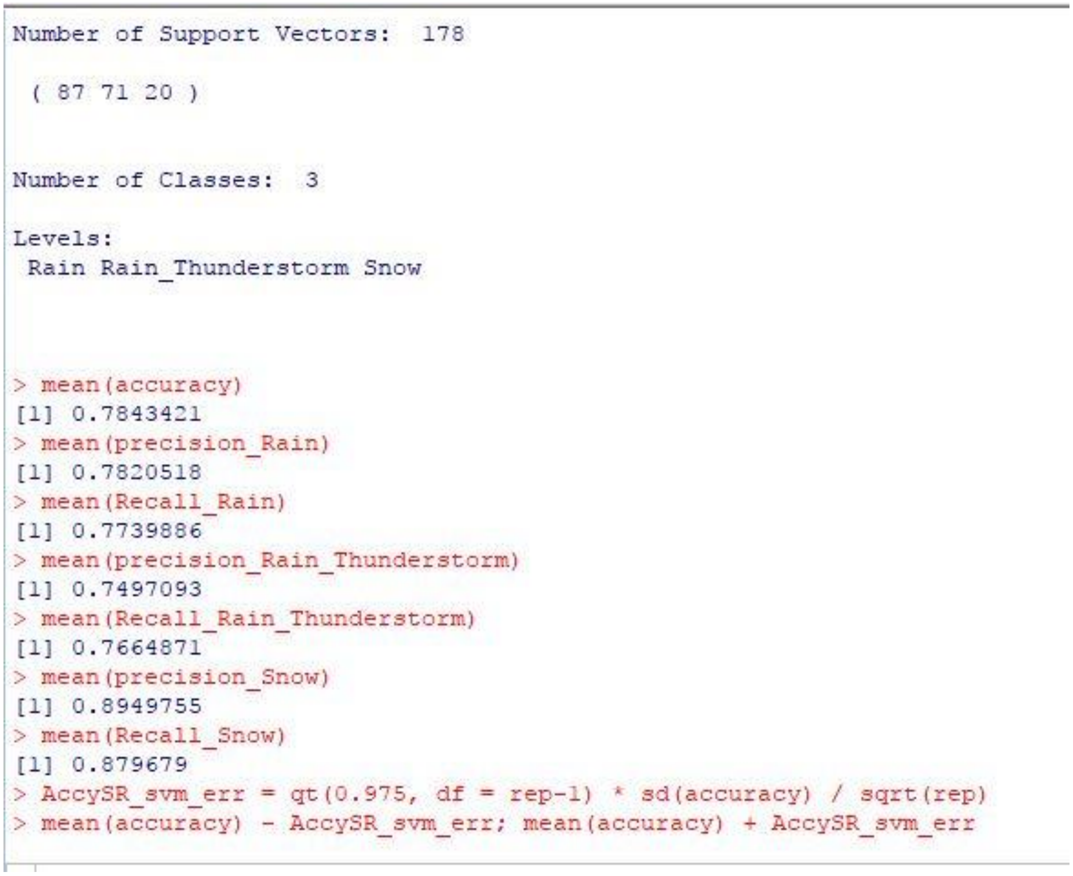
```
> library('e1071')
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweath<-read.csv(file="G:/Fall Semester 2017/ISL/Assignment-2/kc_weather_srt.csv",head=T,sep=",")
> kcweather<-kcweath[,2:9]
> head(kcweather)
  Temp.F Dew_Point.F Humidity.percentage Sea_Level_Press.in Visibility.mi Wind.mph Precip.in Events
1     26          12                  73              30.19             5        9      0.03   Snow
2     31          18                  68              29.95             7       11      0.01   Snow
3     10           1                  63              30.24             5       14      0.02   Snow
4     38          35                  90              29.70             6        5      0.00   Rain
5     40          30                  75              29.80             9        7      0.00   Rain
6     49          29                  51              29.64            10       10      0.00   Rain
> n=366
> nt=290
> neval=n-nt
> rep=100
> accuracy=dim(rep)
> precision_Rain=dim(rep)
> Recall_Rain=dim(rep)
> precision_Rain_Thunderstorm=dim(rep)
> Recall_Rain_Thunderstorm=dim(rep)
> precision_Snow=dim(rep)
> Recall_Snow=dim(rep)
> |
```



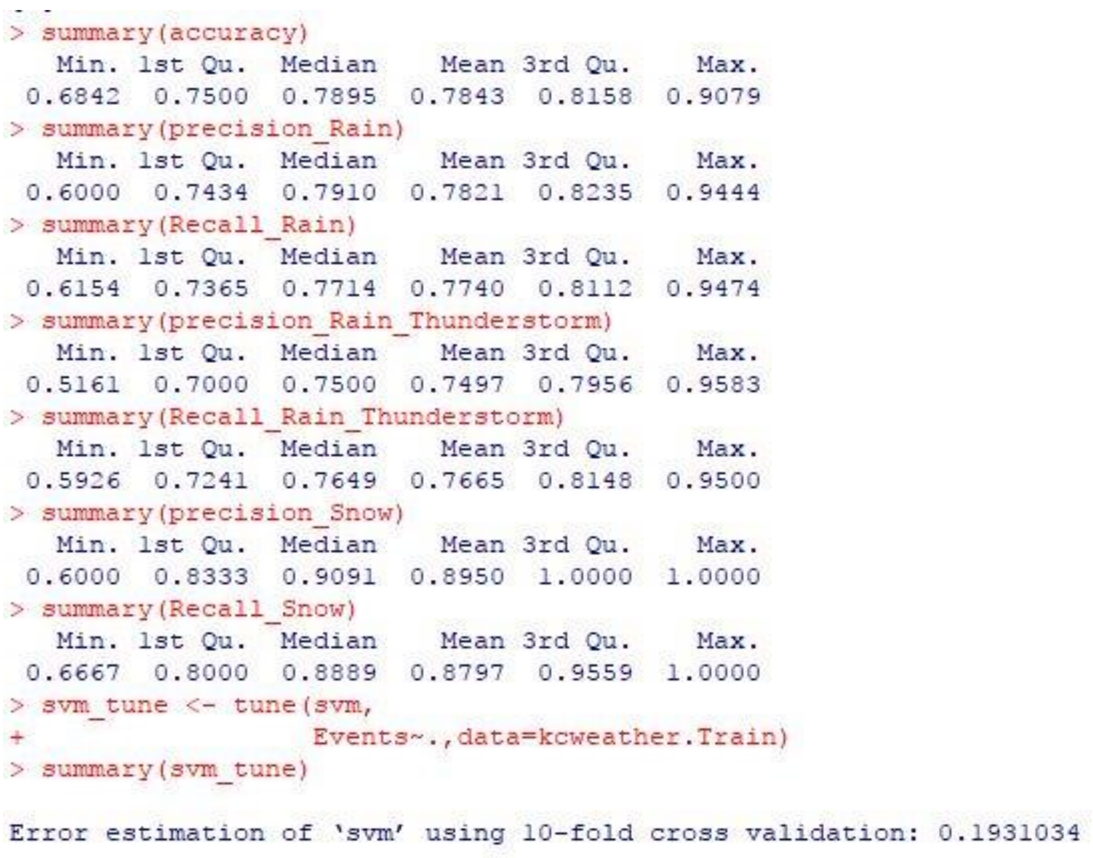
```
> for(k in 1:rep)
+ {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,]
+   kcweather.Test = kcweather[-Tkcweather,]
+   svmfit = svm(Events~.,data=kcweather.Train)
+   p=predict(svmfit,kcweather.Test)
+   cmatrix = table(p,kcweather.Test$Events)
+   accuracy[k] = sum(diag(cmatrix))/sum(cmatrix)
+   precision_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[2,1]+cmatrix[3,1])
+   precision_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[1,2]+cmatrix[2,2]+cmatrix[3,2])
+   precision_Snow[k] = cmatrix[3,3]/(cmatrix[1,3]+cmatrix[2,3]+cmatrix[3,3])
+   Recall_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[1,2]+cmatrix[1,3])
+   Recall_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[2,1]+cmatrix[2,2]+cmatrix[2,3])
+   Recall_Snow[k] = cmatrix[3,3]/(cmatrix[3,1]+cmatrix[3,2]+cmatrix[3,3])
+ }
> summary(svmfit)

Call:
svm(formula = Events ~ ., data = kcweather.Train)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.1428571
```



```
> for(k in 1:rep)
+ {
+   Tkcweather = sample(1:n,nt)
+   kcweather.Train = kcweather[Tkcweather,]
+   kcweather.Test = kcweather[-Tkcweather,]
+   svmfit = svm(Events~.,data=kcweather.Train)
+   p=predict(svmfit,kcweather.Test)
+   cmatrix = table(p,kcweather.Test$Events)
+   accuracy[k] = sum(diag(cmatrix))/sum(cmatrix)
+   precision_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[2,1]+cmatrix[3,1])
+   precision_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[1,2]+cmatrix[2,2]+cmatrix[3,2])
+   precision_Snow[k] = cmatrix[3,3]/(cmatrix[1,3]+cmatrix[2,3]+cmatrix[3,3])
+   Recall_Rain[k] = cmatrix[1,1]/(cmatrix[1,1]+cmatrix[1,2]+cmatrix[1,3])
+   Recall_Rain_Thunderstorm[k] = cmatrix[2,2]/(cmatrix[2,1]+cmatrix[2,2]+cmatrix[2,3])
+   Recall_Snow[k] = cmatrix[3,3]/(cmatrix[3,1]+cmatrix[3,2]+cmatrix[3,3])
+ }
> summary(svmfit)

Call:
svm(formula = Events ~ ., data = kcweather.Train)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.1428571
```



```
Number of Support Vectors:  178

 ( 87 71 20 )


Number of Classes:  3

Levels:
 Rain Rain_Thunderstorm Snow



> mean(accuracy)
[1] 0.7843421
> mean(precision_Rain)
[1] 0.7820518
> mean(Recall_Rain)
[1] 0.7739886
> mean(precision_Rain_Thunderstorm)
[1] 0.7497093
> mean(Recall_Rain_Thunderstorm)
[1] 0.7664871
> mean(precision_Snow)
[1] 0.8949755
> mean(Recall_Snow)
[1] 0.879679
> AccySR_svm_err = qt(0.975, df = rep-1) * sd(accuracy) / sqrt(rep)
> mean(accuracy) - AccySR_svm_err; mean(accuracy) + AccySR_svm_err
```



```
> summary(accuracy)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6842  0.7500  0.7895  0.7843  0.8158  0.9079 
> summary(precision_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6000  0.7434  0.7910  0.7821  0.8235  0.9444 
> summary(Recall_Rain)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6154  0.7365  0.7714  0.7740  0.8112  0.9474 
> summary(precision_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5161  0.7000  0.7500  0.7497  0.7956  0.9583 
> summary(Recall_Rain_Thunderstorm)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.5926  0.7241  0.7649  0.7665  0.8148  0.9500 
> summary(precision_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6000  0.8333  0.9091  0.8950  1.0000  1.0000 
> summary(Recall_Snow)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.6667  0.8000  0.8889  0.8797  0.9559  1.0000 
> svm_tune <- tune(svm,
+                  Events~.,data=kcweather.Train)
> summary(svm_tune)

Error estimation of ‘svm’ using 10-fold cross validation: 0.1931034
```

Here we could see that the accuracy is 0.784. with the total number of svms=178.

Inference:
From the previous assignment comparing the values of different models like LDA, QDA, KNN with SVM, we understand that SVM is a better model than KNN and all others.