ISL Assignment-2 Q2

a) Now consider, from the KC weather data set, just the predictors: Temp.F, Humidity. Percentage, Precip.in. Categorize these three data sets into qualitative predictors. It is up to you to decide on the break points, but you must discuss a rationale for your breakpoints. Now apply, naive Bayes Classifier on the entire data set (with these three qualitative predictors), using 290 of them as a training data set randomly (and the rest as the test data set), over 100 replications. Report on accuracy, precision, and recall.

Solution:

Below is the rationale of break points for conversion of Temperature, precipitation and humidity from being Quantitative values to Qualitative

```
> View(data)
> nbdata=data[,c("Temp.F","Humidity.percentage","Precip.in","Events")]
> head(nbdata)
# A tibble: 6 x 4
  Temp.F Humidity.percentage Precip.in Events
   <int>               <int>     <dbl>  <chr>
1     26                  73      0.03   Snow
2     31                  68      0.01   Snow
3     10                  63      0.02   Snow
4     38                  90      0.00   Rain
5     40                  75      0.00   Rain
6     49                  51      0.00   Rain
```

Break Points Rationale:

Temperature break points:

```
> nbdata$Temp.F[nbdata$Temp.F < 10] <- 'T_1s'
> nbdata$Temp.F[nbdata$Temp.F >=10 & nbdata$Temp.F <20] <- 'T_10s'
> nbdata$Temp.F[nbdata$Temp.F >= 20 & nbdata$Temp.F <30] <- 'T_20s'
> nbdata$Temp.F[nbdata$Temp.F >= 30 & nbdata$Temp.F <40 ] <- 'T_30s'
> nbdata$Temp.F[nbdata$Temp.F >= 40 & nbdata$Temp.F <50 ] <- 'T_40s'
> nbdata$Temp.F[nbdata$Temp.F >= 50 & nbdata$Temp.F <60 ] <- 'T_50s'
> nbdata$Temp.F[nbdata$Temp.F >= 60 & nbdata$Temp.F <70 ] <- 'T_60s'
> nbdata$Temp.F[nbdata$Temp.F >= 70 & nbdata$Temp.F <80 ] <- 'T_70s'
> nbdata$Temp.F[nbdata$Temp.F >= 80 & nbdata$Temp.F <90 ] <- 'T_80s'
```

Humidity break Points:

```
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>= 20 & nbdata$Humidity.percentage <40] <- 'H_20s_30
s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>= 40 & nbdata$Humidity.percentage <50 ] <- 'H_40s_5
0s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage >= 50 & nbdata$Humidity.percentage <70 ] <- 'H_50s_
60s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage >= 70 & nbdata$Humidity.percentage <90 ] <- 'H_70s_
80s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage >= 90 & nbdata$Humidity.percentage <99 ] <- 'H_90s'
```

```
> nbdata$Precip.in[nbdata$Precip.in == 0] <- 'P_0s'
> nbdata$Precip.in[nbdata$Precip.in >0 & nbdata$Precip.in < 1] <- 'P_0.01s'
> nbdata$Precip.in[nbdata$Precip.in >=2 & nbdata$Precip.in <3 ] <- 'P_2s'
> View(nbdata)
```

a)Naive Bayes classifier on entire dataset with three qualitative predictors:

```
> TotalEntries=366
> Training=290
> Testing=TotalEntries-Training
> rep=100
> accuracy=dim(rep)
>
> precision_rain=dim(rep)
> precision_rain_thunderstrom=dim(rep)
> precision_snow=dim(rep)
>
> recall_rain =dim(rep)
> recall_rain_thunderstrom=dim(rep)
> recall_snow=dim(rep)
```

```
> for(k in 1:rep)
+ {
+     train=sample(1:TotalEntries,Training)
+     data.nb_train=naiveBayes(Events~Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in+Visibility.
mi+Wind.mph+Precip.in,data[train,])
+     nbdata.test = nbdata[-train,1:3]
+     predict(data.nb_train, nbdata.test, type="raw")
+     tablin=table(predict(data.nb_train,nbdata.test,type="class"),nbdata[-train,4])
+     errlin[k]=(Testing-sum(diag(tablin)))/Testing
+     accuracy[k]=sum(diag(tablin))/Testing
+     precision_rain[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1]+tablin[3,1])
+     precision_rain_thunderstrom[k] = (tablin[2,2])/(tablin[1,2]+tablin[2,2]+tablin[3,2])
+     precision_snow[k] = (tablin[3,3])/(tablin[1,3]+tablin[2,3]+tablin[3,3])
+     recall_rain[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2]+tablin[1,3])
+     recall_rain_thunderstrom[k]=(tablin[2,2])/(tablin[2,1]+tablin[2,2]+tablin[2,3])
+     recall_snow[k]=(tablin[3,3])/(tablin[3,1]+tablin[3,2]+tablin[3,3])
```

```
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> print(mean(errlin))
[1] 0.5006579
> print(mean(accuracy))
[1] 0.4993421
> print(mean(precision_rain))
[1] 0.9839578
> print(mean(precision_rain_thunderstrom))
[1] 0.04889064
> print(mean(precision_snow))
[1] 0
> print(mean(recall_rain))
[1] 0.4938865
> print(mean(recall_rain_thunderstrom))
[1] NaN
> print(mean(recall_snow))
[1] NaN
> tablin

                    Rain Rain_Thunderstorm Snow
  Rain                38                29    3
  Rain_Thunderstorm    2                 4    0
  Snow                 0                 0    0
```

b) Analyze Temp.F only as quantitative predictor in naive Bayes

Only Temp is considered as a quantitative predictor for this model.

```
> nbdata=data[,c("Temp.F","Humidity.percentage","Precip.in","Events")]
>
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>=20  & nbdata$Humidity.percentage <40] <- 'H_20s_30
s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>=40  & nbdata$Humidity.percentage <50 ] <- 'H_40s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>=50  & nbdata$Humidity.percentage <70 ] <- 'H_50s_6
0s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>=70  & nbdata$Humidity.percentage <90 ] <- 'H_70s_8
0s'
> nbdata$Humidity.percentage[nbdata$Humidity.percentage>=90  & nbdata$Humidity.percentage <99 ] <- 'H_90s'
> nbdata$Precip.in[nbdata$Precip.in == 0] <- 'P_0s'
> nbdata$Precip.in[nbdata$Precip.in >0 & nbdata$Precip.in < 1] <- 'P_0.01s'
> nbdata$Precip.in[nbdata$Precip.in >=2 & nbdata$Precip.in <3 ] <- 'P_2s'
>
> TotalEntries=366
> Training=290
> Testing=TotalEntries-Training
> rep=100
> accuracy=dim(rep)
>
> precision_rain=dim(rep)
> precision_rain_thunderstrom=dim(rep)
> precision_snow=dim(rep)
>
> recall_rain =dim(rep)
> recall_rain_thunderstrom=dim(rep)
> recall_snow=dim(rep)
>
> for(k in 1:rep)
+ {
+     train=sample(1:TotalEntries,Training)
+     data.nb_train=naiveBayes(Events~Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in+Visibility.
mi+Wind.mph+Precip.in,data[train,])
+     nbdata.test = nbdata[-train,1:3]
+     predict(data.nb_train, nbdata.test, type="raw")
+     tablin=table(predict(data.nb_train,nbdata.test,type="class"),nbdata[-train,4])
+     errlin[k]=(Testing-sum(diag(tablin)))/Testing
+     accuracy[k]=sum(diag(tablin))/Testing
+     precision_rain[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1]+tablin[3,1])
+     precision_rain_thunderstrom[k] = (tablin[2,2])/(tablin[1,2]+tablin[2,2]+tablin[3,2])
+     precision_snow[k] = (tablin[3,3])/(tablin[1,3]+tablin[2,3]+tablin[3,3])
+     recall_rain[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2]+tablin[1,3])
+     recall_rain_thunderstrom[k]=(tablin[2,2])/(tablin[2,1]+tablin[2,2]+tablin[2,3])
+     recall_snow[k]=(tablin[3,3])/(tablin[3,1]+tablin[3,2]+tablin[3,3])
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> print(mean(errlin))
[1] 0.2705263
> print(mean(accuracy))
[1] 0.7294737
> print(mean(precision_rain))
[1] 0.6390509
> print(mean(precision_rain_thunderstrom))
[1] 0.7902483
> print(mean(precision_snow))
[1] 0.8833406
> print(mean(recall_rain))
[1] 0.7606657
> print(mean(recall_rain_thunderstrom))
[1] 0.6545778
> print(mean(recall_snow))
[1] 0.8965715
```

c) All predictors (Temperature, Humidity, Precision) as quantitative predictors:

The numerical values of Temperature, Humidity and Precision given in the data set are used for building the Naïve Bayes model.

```
> nbdata=data[,c("Temp.F","Humidity.percentage","Precip.in","Events")]
> View(nbdata)
> TotalEntries=366
> Training=290
> Testing=TotalEntries-Training
> rep=100
> accuracy=dim(rep)
>
> precision_rain=dim(rep)
> precision_rain_thunderstrom=dim(rep)
> precision_snow=dim(rep)
>
> recall_rain =dim(rep)
> recall_rain_thunderstrom=dim(rep)
> recall_snow=dim(rep)
>
> for(k in 1:rep)
+ {
+     train=sample(1:TotalEntries,Training)
+     data.nb_train=naiveBayes(Events~Temp.F+Dew_Point.F+Humidity.percentage+Sea_Level_Press.in+Visibility.
mi+Wind.mph+Precip.in,data[train,])
+     nbdata.test = nbdata[-train,1:3]
+     predict(data.nb_train, nbdata.test, type="raw")
+     tablin=table(predict(data.nb_train,nbdata.test,type="class"),nbdata[-train,4])
+     errlin[k]=(Testing-sum(diag(tablin)))/Testing
+     accuracy[k]=sum(diag(tablin))/Testing
+     precision_rain[k] = (tablin[1,1])/(tablin[1,1]+tablin[2,1]+tablin[3,1])
+     precision_rain_thunderstrom[k] = (tablin[2,2])/(tablin[1,2]+tablin[2,2]+tablin[3,2])
+     precision_snow[k] = (tablin[3,3])/(tablin[1,3]+tablin[2,3]+tablin[3,3])
+     recall_rain[k]=(tablin[1,1])/(tablin[1,1]+tablin[1,2]+tablin[1,3])
+     recall_rain_thunderstrom[k]=(tablin[2,2])/(tablin[2,1]+tablin[2,2]+tablin[2,3])
+     recall_snow[k]=(tablin[3,3])/(tablin[3,1]+tablin[3,2]+tablin[3,3])
+ }
> print(mean(errlin))
[1] 0.2603947
> print(mean(accuracy))
[1] 0.7396053
> print(mean(precision_rain))
[1] 0.6953179
> print(mean(precision_rain_thunderstrom))
[1] 0.7296526
> print(mean(precision_snow))
[1] 0.9415376
> print(mean(recall_rain))
[1] 0.7583772
> print(mean(recall_rain_thunderstrom))
[1] 0.7406478
> print(mean(recall_snow))
[1] 0.6942448
```

Summary:

Naïve Bayes	Error	Accuracy	Precision			Recall		
			Rain	Rain_Thunderstorm	Snow	Rain	Rain_Thunderstorm	Snow
Three Qualitative Predictors	0.5006579	0.4993421	0.9839578	0.04889064	0	0.4938865	Nan	Nan
Only temperature Quantitative	0.2705263	0.7294737	0.6390509	0.7902483	0.8833406	0.7606657	0.6545778	0.8965715
All Quantitative predictors	0.2603947	0.7396053	0.6953179	0.7296526	0.9415376	0.7583772	0.7406478	0.6942448

Analysis Summary:

1. Naïve Bayes model for the Kansas City weather data set given has high accuracy, and low error rate when we consider all the predictors as quantitative.
2. When only the temperature is quantitative the recall value is high compared to the remaining models and a decent precision values. So, when recall and accuracy are of higher importance then this model is the best fit.

Model	Error	Accuracy	Precision			Recall		
			Rain	Rain_Thunderstorm	Snow	Rain	Rain_Thunderstorm	Snow
LDA	0.2519737	0.7480263	0.7523023	0.7237916	0.8021281	0.7087325	0.7499575	0.8945227
QDA	0.255	0.745	0.7498849	0.7184631	0.7964458	0.7016425	0.7264472	0.9515166
KNN(k=3)	0.847623	0.733815	0.7262863	0.6946528	0.8897852	0.7250768	0.6982034	0.8713335

3. When Navie bayes model is compared to KNN thought the accuracy level is slightly different, but there is a more variance in the error rate. So Navie Bayes better perform than the KNN.

4.LDA and QDA outperforms than all the Navie bayes models when all the metrices has been taken in to consideration**.**