Anomaly Detection

Final Project

(30 points)

A company has multiple salespersons, each having a unique ID, selling multiple products, each with a unique PROD code. The data consists of records of such sales. For each sale, besides the salesperson ID and the product code PROD, the number of units sold (QUANT) and the total amount of the sale (VAL) is also recorded. The price of a product can be vary depending on the salesperson involved, the number of units involved and the actual sales event. This introduces a lot of variability in the sales records and the company is interested in anomalous sale events, which might indicate fraudulent transactions.

Some of the available data of past sales has been reviewed by experts within the company and each such sale has been tagged as 'fraud' or 'ok'. The unreviewed sales are simply tagged as 'unkn'. This extra information is provided in the data as a label (INSP) for each sale.

Your task is to develop and evaluate strategies for detecting anomalies in this dataset.

```
> sales<-read.csv("sales.csv", sep = ",", stringsAsFactors = FALSE)

> head(sales)

  X ID Prod Quant   Val Insp

1 1 v1   p1   182  1665 unkn

2 2 v2   p1  3072  8780 unkn

3 3 v3   p1 20393 76990 unkn

4 4 v4   p1   112  1100 unkn

5 5 v3   p1  6164 20260 unkn

6 6 v5   p2   104  1155 unkn

> summary(sales)

      X             ID           Prod          Quant             Val           Insp

 Min.   :     1   v431   : 10159   p1125  :  3923   Min.   :      100   Min.   :   1005   fraud:  1270

 1st Qu.:102795   v54    :  6017   p3774  :  1824   1st Qu.:      107   1st Qu.:   1345   ok   : 14462

 Median :205896   v426   :  3902   p1437  :  1720   Median :      168   Median :   2675   unkn :385414

 Mean   :205932   v1679  :  3016   p1917  :  1702   Mean   :     8442   Mean   :  14617
 3rd Qu.:309013   v1085  :  3001   p4089  :  1598   3rd Qu.:      738   3rd Qu.:   8680
 Max.   :411818   v1183  :  2642   p2742  :  1519   Max.   :473883883   Max.   :4642955
                  (Other):372409   (Other):388860   NA's   :13842       NA's   :1182
> dim(sales) # total 401146 raw records
[1] 401146      6
> nlevels(sales$ID)
[1] 6016
> nlevels(sales$Prod)
[1] 4548
> anyNA(sales$Quant)
[1] TRUE
> anyNA(sales$Val)
[1] TRUE
> anyNA(sales$Insp)
[1] FALSE
> length(which(is.na(sales$Quant) & is.na(sales$Val)))
[1] 888
> sales$UnitPrice<-sales$Val/sales$Quant
> summary(sales$UnitPrice) # mean unit price is $20.30.
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
    0.00     8.46    11.89    20.30    19.11 26460.70    14136
> t(t(names(sales)))
     [,1]
[1,] "X"
[2,] "ID"
```

```
[3,] "Prod"
[4,] "Quant"
[5,] "Val"
[6,] "Insp"
[7,] "UnitPrice"
> str(sales)
'data.frame':   401146 obs. of  7 variables:
 $ X        : int  1 2 3 4 5 6 7 8 9 10 ...
 $ ID       : chr  "v1" "v2" "v3" "v4" ...
 $ Prod     : chr  "p1" "p1" "p1" "p1" ...
 $ Quant    : int  182 3072 20393 112 6164 104 350 200 233 118 ...
 $ Val      : int  1665 8780 76990 1100 20260 1155 5680 4010 2855 1175 ...
 $ Insp     : chr  "unkn" "unkn" "unkn" "unkn" ...
 $ UnitPrice: num  9.15 2.86 3.78 9.82 3.29 ...
> unitPriceProd<-aggregate(sales$UnitPrice, list(sales$Prod), median, na.rm=T)
> topP <- sapply(c(T,F),function(o)unitPriceProd[order(unitPriceProd[,2],decreasing=o)[1:5],1])
> topP # top 5 and bottom 5 median unit price aggregated by Prod.
     [,1]    [,2]
[1,] "p3689" "p560"
[2,] "p2453" "p559"
[3,] "p2452" "p4195"
[4,] "p2456" "p601"
[5,] "p2459" "p563"
> valuePerID <- aggregate(sales$Val,list(sales$ID),sum,na.rm=T)
> topS<-sapply(c(T,F),function(o)valuePerID[order(valuePerID$x,decreasing=o)[1:5],1])
> topS # top 5 and bottom 5 median total sales value aggregated by ID.
     [,1]    [,2]
[1,] "v431"  "v3355"
[2,] "v54"   "v6069"
[3,] "v19"   "v5876"
[4,] "v4520" "v6058"
[5,] "v955"  "v4515"
> quantProd <- aggregate(sales$Quant,list(sales$Prod),sum,na.rm=T)
> topQuantProd<-sapply(c(T,F),function(o)quantProd[order(quantProd$x,decreasing=o)[1:5],1])
> topQuantProd # top 5 and bottom 5 total number of units saled aggregated by Prod.
     [,1]    [,2]
[1,] "p2516" "p2442"
[2,] "p3599" "p2443"
[3,] "p314"  "p1653"
[4,] "p569"  "p4101"
[5,] "p319"  "p3678"
> sales <- sales[complete.cases(sales), ] #removing missing value records
> dim(sales) #there are total of 387010 records after removing the missings
[1] 387010      7
sales$Insp <- factor(sales$Insp)
salesOK <- sales[sales$Insp=="ok", ]
> dim(salesOK)
[1] 14347     7
> salesF <- sales[sales$Insp=="fraud", ]
> dim(salesF)
```

```
[1] 1199    7
> outliers<-tapply(sales$UnitPrice, list(sales$Prod), function(x) length(boxplot.stats(x)$out))
> length(outliers)
[1] 4546
> salesvalid<-rbind(salesOK,salesF)
> table(salesvalid$Insp)


fraud    ok  unkn
 1199 14347     0
> salesvalid$Prod<-gsub("p", "", salesvalid$Prod)
> head(salesvalid)
    X  ID Prod Quant    Val Insp UnitPrice
49 53 v42   11 51097 310780   ok  6.082157
52 56 v45   11   260   1925   ok  7.403846
64 68 v42   11 51282 278770   ok  5.436020
73 77 v50   11 46903 281485   ok  6.001428
78 82 v46   12   475   2600   ok  5.473684
80 84 v48   12   433   3395   ok  7.840647
> salesvalid$Prod<-as.double(salesvalid$Prod)
> index<-createDataPartition(salesvalid$Insp, p=0.7, list=FALSE, times=1)
Warning message:
In createDataPartition(salesvalid$Insp, p = 0.7, list = FALSE, times = 1) :
  Some classes have no records ( unkn ) and these will be ignored
> train<-salesvalid[index,]
> test<-salesvalid[-index,]
> salesknn<-knn(train=train[c("Prod", "Quant", "Val")], test=test[c("Prod", "Quant", "Val")],
cl=train$Insp, k=4)

> CrossTable(x = test$Insp, y = salesknn, prop.chisq=FALSE)

   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  4663

             | salesknn
   test$Insp |     fraud |        ok | Row Total |
-------------|-----------|-----------|-----------|
       fraud |       175 |       184 |       359 |
             |     0.487 |     0.513 |     0.077 |
             |     0.788 |     0.041 |           |
             |     0.038 |     0.039 |           |
-------------|-----------|-----------|-----------|
          ok |        47 |      4257 |      4304 |
             |     0.011 |     0.989 |     0.923 |
             |     0.212 |     0.959 |           |
             |     0.010 |     0.913 |           |
-------------|-----------|-----------|-----------|
Column Total |       222 |      4441 |      4663 |
             |     0.048 |     0.952 |           |
-------------|-----------|-----------|-----------|


> accuracy<-sum(1*(salesknn==test$Insp))/length(salesknn)

> accuracy

[1] 0.9504611
```