DATA ANALYTICS WITH R, EXCEL AND TABLAEU

ASSIGNMENT decision tree based models 14.2

ANSWERS

By ASHISH S SHANBHAG

ashishshanbhag108@gmail.com

5

a. Read the dataset and identify the right features

Ans In the train data, the basetimes were in the years 2010 and 2011. In the test data the basetimes were in February and March 2012. This simulates the real-world situtation in which training data from the past is available to predict events in the future.
The train data was generated from different basetimes that may temporally overlap. Therefore, if you simply split the train into disjoint partitions, the underlying time intervals may overlap. Therefore, the you should use the provided, temporally disjoint train and test splits in order to ensure that the evaluation is fair

b. Clean dataset, impute missing values and perform exploratory data analysis.

Ans #load train and test file

```
> train <- read.csv("train_Big.csv")
> test <- read.csv("test_Big.csv")

#add a column
> test$Item_Outlet_Sales <- 1

#combine the data set
> combi <- rbind(train, test)

#impute missing values with median
> combi$Item_Weight[is.na(combi$Item_Weight)] <-
median(combi$Item_Weight, na.rm = TRUE)
#impute 0 with median

> combi$Item_Visibility <- ifelse(combi$Item_Visibility == 0,
median(combi$Item_Visibility),combi$Item_Visibility)
```

```
#find mode and impute
> table(combi$Outlet_Size, combi$Outlet_Type)
> levels(combi$Outlet_Size)[1] <- "Other"
```

Till here, we've imputed missing values. Now we are left with removing the dependent (response) variable and other identifier variables (if any). As we said above, we are practicing an unsupervised learning technique, hence response variable must be removed.

```
#remove the dependent and identifier variables
> my_data <- subset(combi, select = -c(Item_Outlet_Sales,
Item_Identifier,                                    Outlet_Identifier))
```

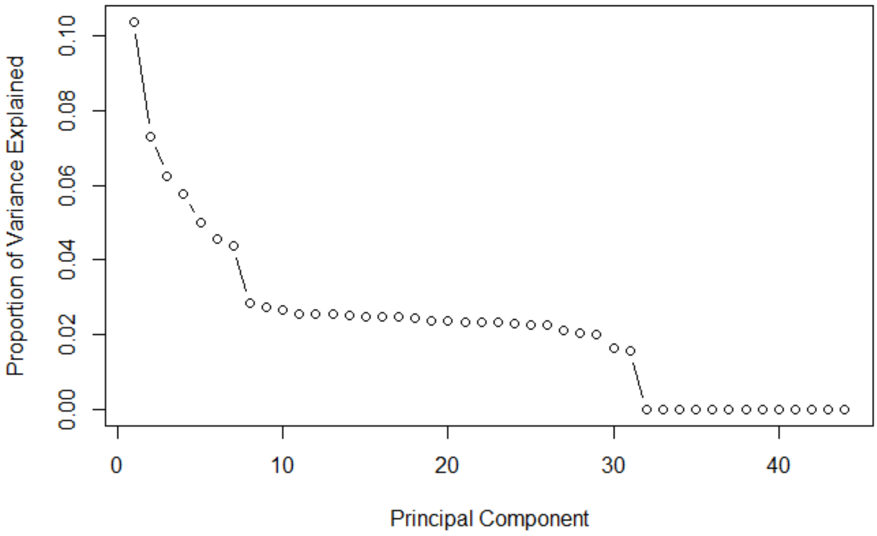
Let's check the available variables (a.k.a predictors) in the data set.

```
#check available variables
> colnames(my_data)
```

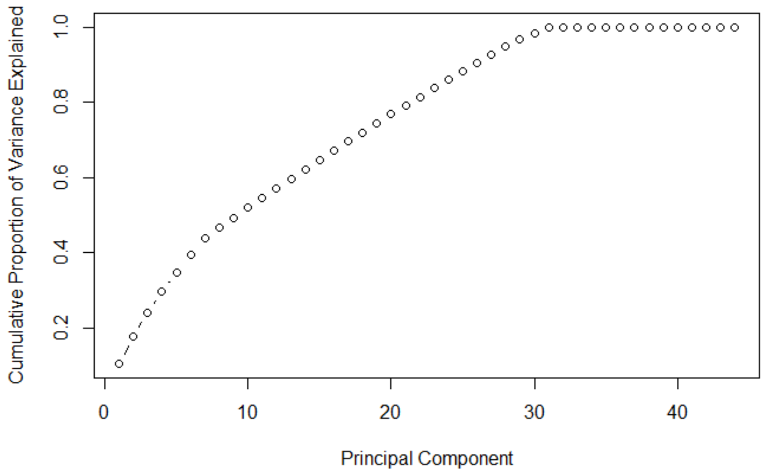
c. Visualize the dataset and make inferences from that

Ans #scree plot

```
          > plot(prop_varex, xlab = "Principal Component",
           ylab = "Proportion of Variance Explained",
           type = "b")
```



```
#cumulative scree plot
> plot
(cumsum(prop_varex), xlab = "PrincipalComponent",ylab"Cumulative
Proportion of Variance Explained",type = "b")
```



d. Perform any 3 hypothesis tests using columns of your choice, make conclusions

Ans

```
    #run a decision tree
> install.packages("rpart")
> library(rpart)
> rpart.model <- rpart(Item_Outlet_Sales ~ .,data = train.data, method =
"anova")
> rpart.model
#transform test into PCA
> test.data <- predict(prin_comp, newdata = pca.test)
> test.data <- as.data.frame(test.data)
#select the first 30 components
> test.data <- test.data[,1:30]
#make prediction on test data
> rpart.prediction <- predict(rpart.model, test.data)
#For fun, finally check your score of leaderboard
> sample <- read.csv("SampleSubmission_TmnO39y.csv")
> final.sub <- data.frame(Item_Identifier = sample$Item_Identifier,
Outlet_Identifier = sample$Outlet_Identifier, Item_Outlet_Sales =
rpart.prediction)
> write.csv(final.sub, "pca.csv",row.names = F)
```

e. Create a linear regression model to predict the number of comments in the next 24 hours (relative to basetime)

Ans

```
#outputs the mean of variables
prin_comp$center
#outputs the standard deviation of variables
prin_comp$scale
```

2. The rotation measure provides the principal component loading. Each column of rotation matrix contains the principal component loading vector. This is the most important measure we should be interested in.

```
> prin_comp$rotation
```

This returns 44 principal components loadings. Is that correct ? Absolutely. In a data set, the maximum number of principal component loadings is a minimum of (n-1, p). Let’s look at first 4 principal components and first 5 rows.

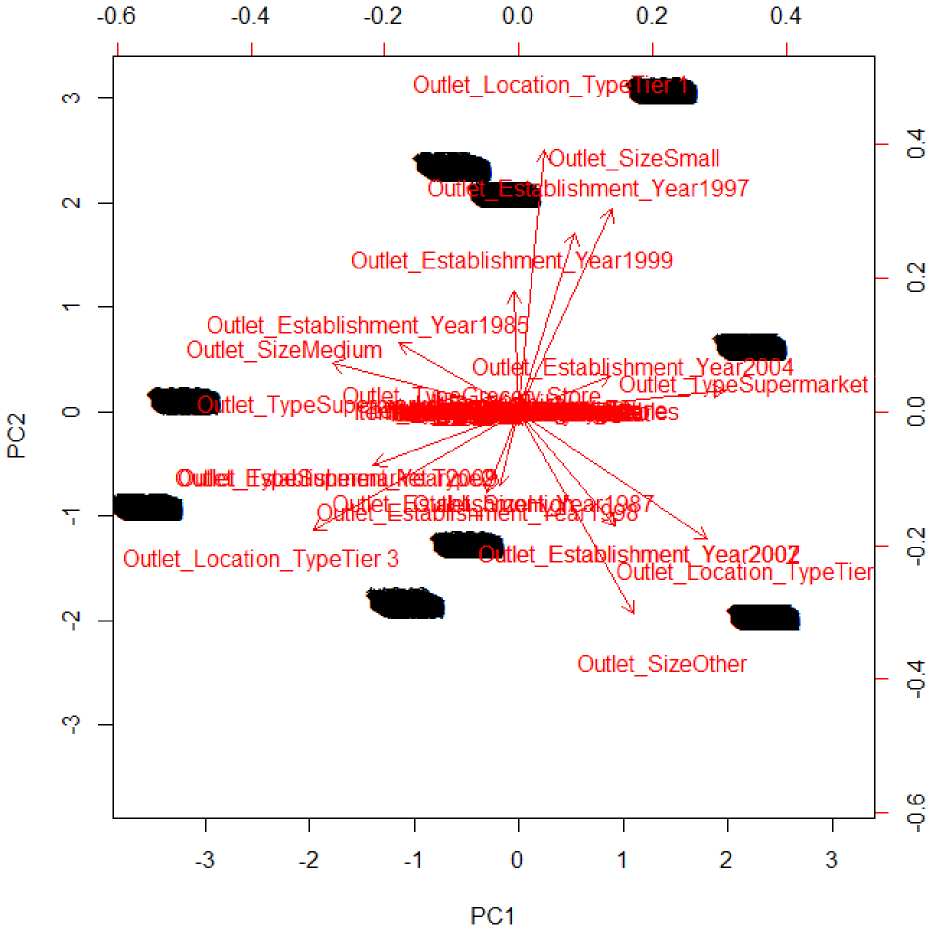
```
> prin_comp$rotation[1:5,1:4]
                              PC1           PC2           PC3            PC4
Item_Weight   0.0054429225   -0.001285666   0.011246194   0.011887106
Item_Fat_ContentLF -0.0021983314    0.003768557  -0.009790094  -
0.016789483
Item_Fat_Contentlow fat   -0.0019042710    0.001866905  -0.003066415  -
0.018396143
Item_Fat_ContentLow Fat    0.0027936467   -
0.002234328   0.028309811   0.056822747
Item_Fat_Contentreg       0.0002936319    0.001120931   0.009033254  -
0.001026615
```

3. In order to compute the principal component score vector, we don’t need to multiply the loading with data. Rather, the matrix x has the principal component score vectors in a 8523 × 44 dimension.

```
> dim(prin_comp$x)
[1] 8523    44
```

Let’s plot the resultant principal components.

```
> biplot(prin_comp, scale = 0)
```



f. Fine tune the model and represent important features

Ans The train data was generated from different basetimes that may temporally overlap. Therefore, if you simply split the train into disjoint partitions, the underlying time intervals may overlap. Therefore, the you should use the provided, temporally disjoint train and test splits in order to ensure that the evaluation is fair

g. Interpret the summary of the linear model

Ans We infer than first principal component corresponds to a measure of Outlet_TypeSupermarket, Outlet_Establishment_Year 2007. Similarly, it can be said that the second component corresponds to a measure of Outlet_Location_TypeTier1, Outlet_Sizeother. For exact measure of a variable in a component, you should look at rotation matrix(above) again.

h. Report the test accuracy vs. the training accuracy

Ans Sadly, 6 out of 9 variables are categorical in nature. We have some additional work to do now. We'll convert these categorical variables into numeric using one hot encoding.

i. Interpret the final model coefficients

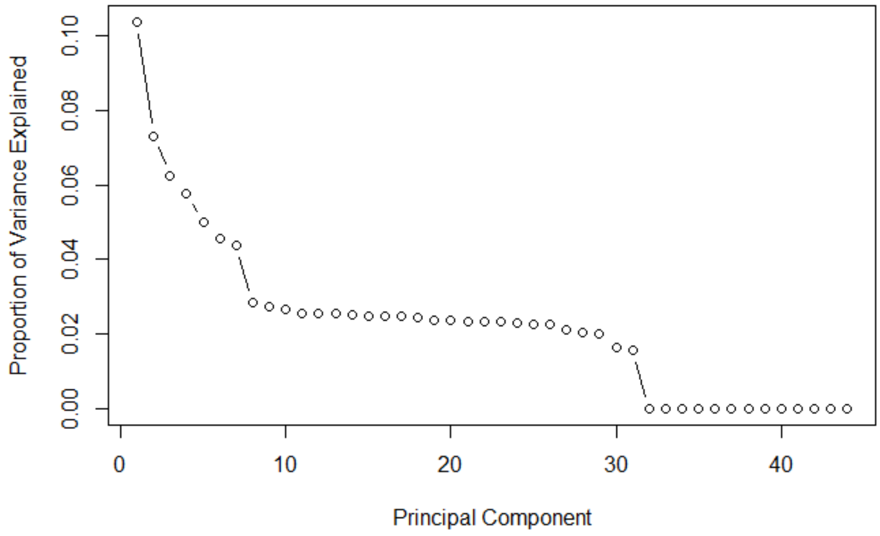
Ans This returns 44 principal components loadings. Is that correct ? Absolutely. In a data set, the maximum number of principal component loadings is a minimum of (n-1, p). Let's look at first 4 principal components and first 5 rows.

```
> prin_comp$rotation[1:5,1:4]
                                    PC1            PC2            PC3              PC4
Item_Weight              0.0054429225      -0.001285666    0.011246194      0.011887106
Item_Fat_ContentLF       -0.0021983314     0.003768557    -0.009790094     -0.016789483
Item_Fat_Contentlow fat  -0.0019042710     0.001866905    -0.003066415     -0.018396143
Item_Fat_ContentLow Fat   0.0027936467    -0.002234328     0.028309811      0.056822747
Item_Fat_Contentreg       0.0002936319     0.001120931     0.009033254     -0.001026615
```

j. Plot the model result and compare it with assumptions of the model

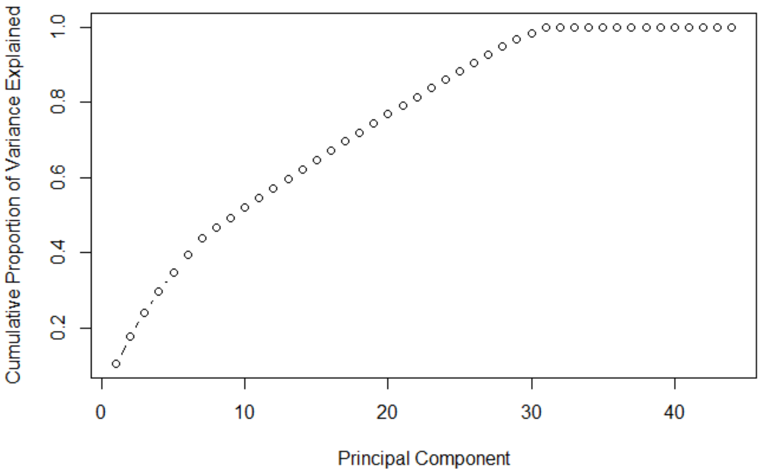
Ans #scree plot

```
        > plot(prop_varex, xlab = "Principal Component",
         ylab = "Proportion of Variance Explained",
         type = "b")
```



#cumulative scree plot

```
> plot
(cumsum(prop_varex), xlab = "PrincipalComponent",ylab"Cumulative
Proportion of Variance Explained",type = "b")
```


Cumulative Proportion of Variance Explained
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
Principal Component