Dimensionality Reduction

Benton Fariss

2023-03-23

The Dataset: https://www.kaggle.com/datasets/ujjwalchowdhury/walmartcleaned?datasetId=2169207&language=R #Read in dataset

```
data <- read.csv('walmart_cleaned.csv')
set.seed(1234)
data$IsHoliday <- as.factor(data$IsHoliday)
print(nrow(data))
```

```
## [1] 421570
```

#Load Libraries

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(class)
```

#Data Cleaning Remove Unimportant Columns

```
data_remove <- c(1, 3)
data <- data[,-data_remove]
str(data)
```

```
## 'data.frame':    421570 obs. of  15 variables:
```

```
##  $ Store       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ IsHoliday   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Dept        : num  1 26 17 45 28 79 55 5 58 7 ...
##  $ Weekly_Sales: num  24924.5 11737.1 13223.8 37.4 1085.3 ...
##  $ Temperature : num  42.3 42.3 42.3 42.3 42.3 ...
##  $ Fuel_Price  : num  2.57 2.57 2.57 2.57 2.57 ...
##  $ MarkDown1   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MarkDown2   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MarkDown3   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MarkDown4   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ MarkDown5   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CPI         : num  211 211 211 211 211 ...
##  $ Unemployment: num  8.11 8.11 8.11 8.11 8.11 ...
##  $ Type        : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Size        : int  151315 151315 151315 151315 151315 151315 151315 151315 151315 151315 ...
```

Since I already removed the only column with NA's I only need to check and Remove the 0's that are not contributing to the dataset

```r
print(sapply(data, function(x) sum(length(which(x==0)))))
```

```
##        Store    IsHoliday         Dept Weekly_Sales  Temperature   Fuel_Price
##            0       391909            0           73            0            0
##    MarkDown1    MarkDown2    MarkDown3    MarkDown4    MarkDown5          CPI
##       270889       310529       284546       286603       270138            0
## Unemployment         Type         Size
##            0            0            0
```

```r
data <- data[!(data$MarkDown1==0),]
data <- data[!(data$MarkDown2==0),]
data <- data[!(data$MarkDown3==0),]
data <- data[!(data$MarkDown4==0),]
data <- data[!(data$MarkDown5==0),]
print(nrow(data))
```

```
## [1] 96782
```

#Split data into Train/Test

```r
i <- sample(1:nrow(data), nrow(data)*.8, replace=FALSE)
train <- data[i,]
test <- data[-i,]
print(nrow(train))
```

```
## [1] 77425
```

#PCA PCA

```r
pca_out <- preProcess(train[,1:15], method=c("center", "scale", "pca"))
pca_out
```

```
## Created from 77425 samples and 15 variables
##
## Pre-processing:
##   - centered (14)
##   - ignored (1)
##   - principal component signal extraction (14)
##   - scaled (14)
```

```
##
## PCA needed 12 components to capture 95 percent of the variance
```

```r
pca_train <- predict(pca_out, train[,1:15])
pca_test <- predict(pca_out, test[,])
```

Regression on the original dataset

```r
library(class)
pred <- knn(train=train[,2:15], test=test[,2:15], cl=train[,2], k=3)
acc <- mean(pred==test$IsHoliday)
print(paste("Normal kNN Accuracy: ",acc))
```

```
## [1] "Normal kNN Accuracy:  0.990132768507517"
```

Regression on reduced dataset

```r
library(class)
pred <- knn(train=pca_train[,2:12], test=pca_test[,2:12], cl=pca_train[,1], k=3)
acc <- mean(pred==test$IsHoliday)
print(paste("PCA kNN Accuracy: ",acc))
```

```
## [1] "PCA kNN Accuracy:  0.993232422379501"
```

Accuracy Comparison: We get a higher accuracy on the reduced dataset with it's accuracy being .9932324. This is because we reduced the observations that were less correlated.
#LDA LDA. This shows us the means of all of the observations when it is and isn't a holiday.

```r
lda1 <- lda(train$IsHoliday~., data=train)
lda1$means
```

```
##      Store     Dept Weekly_Sales Temperature Fuel_Price MarkDown1 MarkDown2
## 0 20.25965 44.25840     17781.30    58.24422   3.631164  9058.710  2522.687
## 1 20.02316 44.62802     19085.06    49.00938   3.504659  7080.171 14163.837
##    MarkDown3 MarkDown4 MarkDown5      CPI Unemployment     Type     Size
## 0   245.5332  4067.581  5462.903 174.7823     7.401109 2.590654 155769.6
## 1 15291.2304  3653.116  3856.692 174.5878     7.522574 2.555867 151921.3
```

LDA Predictions

```r
pred <- predict(lda1, newdata=test, type="class")
acc <- mean(pred$class==test$IsHoliday)
print(paste("LDA Accuracy: ", acc))
```

```
## [1] "LDA Accuracy:  0.931084362246216"
```

#Overall Analysis Overall, the PCA accuracy increased the accuracy of the dataset. This is because PCA is great for reducing datasets with many dimensions, such as this one, and in turn the noise is quieted. PCA also reduces noise by removing redundant and low-correlated observations and variables. The PCA kNN had an accuracy of .9932324 which was higher than the normal kNN's accuracy of .990132. My PCA reduced the datasets 15 variables to just 12 while keeping a 95% variance. My LDA accuracy was .931084 which is not bad at all, just not as good as how PCA was. Overall, both dataset reduction methods proved to be viable on this dataset.