Principal Component Regression

Contents

- 1. Multiple Linear Regression-Quick Recap
- 2. The Problem of Multicollinearity
- 3. Principal Component Analysis General Approach
- 4. Principal Component Regression (PCR)
 - i. Introduction
 - ii. Statistical Model
- 5. PCR in R

Multiple Linear Regression: Statistical Model

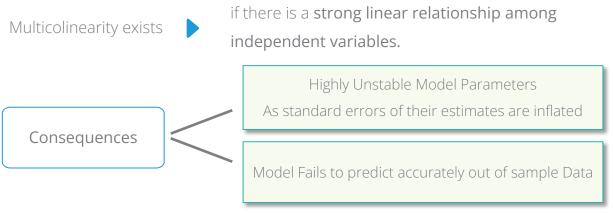
$$Y = b_0 + b_1 X_1 + b_2 X_2 + ... + b_p X_p + e$$

```
Where,
Y : Dependent Variable

X<sub>1</sub>, X<sub>2</sub>,..., X<sub>p</sub> : Independent Variables
b<sub>0</sub>, b<sub>1</sub>,..., b<sub>p</sub> : Parameters of Model
e : Random Error Component
```

Independent variables can either be Continuous or Categorical

Problem of Multicolinearity



Multicolinearity is detected using Variance Inflation Factor, VIF

Tolerance = $1 - R_i^2$

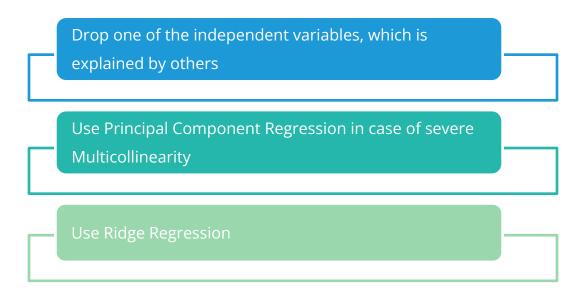
VIF = 1/Tolerance

where R_i^2 (R Squared) is obtained using regression of Xi on other independent variables

Any VIF > 5, indicates presence of multicollinearity

Multicollinearity – Remedial Measures

The problem of Multicollinearity can be solved by different approaches:



Principal Component Regression

In Principal Component Regression,

First k principal components are used as independent variables instead of original X variables

- Each PC is a linear combination of all X variables
- Final model is expressed in terms of original independent variables for ease of interpretation

Principal Component Regression

Transformation into PCs

The original **p** variables are transformed into a new set of orthogonal or uncorrelated variables called "Principal Components"



Regression Analysis



In the second step, after elimination of the least important principal components, a multiple regression analysis of the response variable against the reduced set of principal components is performed using the OLS estimation

Back Transformation In the third step, model equation is back transformed in terms of original variables.

PCR-Statistical Model

Model in terms of original X variables:

$$Y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_p x_p + e$$

Model in terms of Principal Components:

$$Y = a_0 + a_1 PC_1 + a_2 PC_2 + \dots + a_k PC_k + e'$$

Case Study

Background

 A company periodically records data for sales and expenses. The company wishes to model the relationship between its sales and sales related expenses and obtain predictions

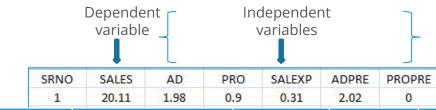
Objective

 To predict incremental sales based on planned sales related expenses

Available Information

- Data available for 143 micro business zones
- Sales is the Dependent Variable
- Expenditure towards advertisements and promotions in the current and previous months are Predictors

Data Snapshot



	Columns	Description	Туре	Measuremen t	Possible values
Ob. 2.1.1.1.1.1.	SRNO	Serial Number	-	-	Intergers
	SALES	Incremental Sales	Numerical	Euros Million	positive value
	AD	Current Advertising Expenses	Numerical	Euros Million	positive value
	PRO	Current Promotional Expenses	Numerical	Euros Million	positive value
	SALEXP	Misc. Sales Expenses	Numerical	Euros Million	positive value
	ADPRE	Previous Period's Advertising Expenses	Numerical	Euros Million	positive values
	PROPRE	Previous Period's Promotional Expenses	Numerical	Euros Million	Positive value

Output of summary

```
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.8147 6.5314 -1.656 0.10005
           4.6762 1.4100 3.316 0.00117 **
AD
   7.7886 1.2628 6.168 7.3e-09 ***
PRO
SALEXP 22.4089 0.7704 29.089 < 2e-16 ***
      3.1856 1.2442 2.560 0.01154 *
ADPRE
PROPRE
       3.4970
                     1.3697 2.553 0.01177 *
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '1
Residual standard error: 1.201 on 137 degrees of freedom
Multiple R-squared: 0.9089, Adjusted R-squared: 0.9055
```

F-statistic: 273.2 on 5 and 137 DF, p-value: < 2.2e-16

Interpretation:

Multiple
R-Squared is
0.9089, showing
model to be a
good fit.

Output of VIF

```
AD PRO SALEXP ADPRE PROPRE 36.159771 31.846727 1.076284 24.781948 42.346468
```

Interpretation:

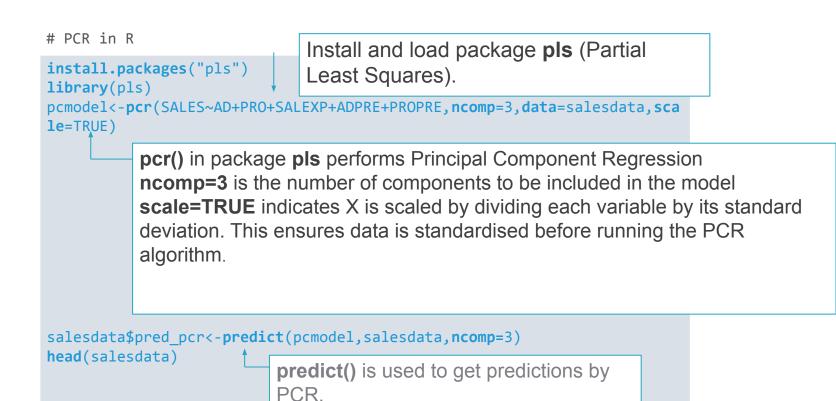
 VIF values are very high (>5, except for SALEEXP) indicating severe multicolinearity problem.

```
# PCA in R
# Subsetting data for getting Principal components and performing PCA
salesdatapca<-subset(salesdata, select=c(-SRNO, -SALES))</pre>
pc<-princomp(formula=~.,data=salesdatapca, cor=T)</pre>
summary(pc)
                      princomp()from base R performs PCA on the
                       given numeric data matrix
                      formula= contains the numeric variables. ~.
                       ensures all variables are taken
                      cor=T indicates that calculations should be
                    done
                       using the Correlation Matrix.
                      summary() generates the summary of PCA
```

Output

Interpretation:

 The first three principal components explain 82% of the variation in the data. Therefore, we can use 3 PC's in Regression Model



Output

```
SRNO SALES AD PRO SALEXP ADPRE PROPRE pred_pcr
  1 20.11 1.98 0.9
                   0.31
                        2.02
                                0.0 21.29053
  2 15.10 1.94 0.0
                   0.30 1.99 1.0 18.16976
  3 18.68 2.20 0.8
                   0.35 1.93
                               0.0 21.27149
                               0.8 17.62114
 4 16.05 2.00 0.0
                   0.35 2.20
  5 21.30 1.69 1.3
                   0.30 2.00 0.0 22.97930
  6 17.85 1.74 0.3
                   0.32 1.69
                                1.3 20.57217
```

pred_pcr column gives predicted values of SALES using PCR.

Comparing Linear Regression Model and PCR model on Test data

```
# Importing Test Data
salesdata_test<-read.csv("pcrdata_test.csv",header=TRUE)</pre>
# Getting RMSE of linear regression model
salesdata test$lmpredict<-predict(predsales,salesdata test)←
salesdata test$1mres<-(salesdata test$SALES-salesdata test$1mpredict)</pre>
RMSE lm<-sqrt(mean(salesdata test$lmres)**2)</pre>
                              predict () will give the predicted value for the model.
# Getting RMSE of PCR model
salesdata test$pcrpredict<-predict(pcmodel,salesdata test,ncomp=3}-</pre>
salesdata test$pcrres<-(salesdata test$SALES-salesdata test$pcrpredict</pre>
RMSE pcr<-sqrt(mean(salesdata test$pcrres)**2)</pre>
```

Comparing Linear Regression Model and PCR model on Test data

Viewing data after adding predicted & residual variables

```
head(salesdata test)
 # Output
           AD PRO SALEXP ADPRE PROPRE Impredict
                                                   1mres pcrpredict
SRNO SALES
                                                                      pcrres
  1 28.93 2.75 1.00
                    0.72 1.97
                                0.02 32.31368
                                               -3.3836776
                                                          23. 23291
                                                                    5.6970943
  2 25.96 1.73 1.06
                   0.89 2.77 0.02 34.36925
                                              -8.4092464 22.26693
                                                                    3,6930660
  3 31.25 2.19 1.26 0.79 1.22 0.42 32.29821 -1.0482117 27.61578 3.6342207
  4 25.05 1.82 1.45 0.83 2.23 0.15 35.21751 -10.1675083
                                                          25.21307 -0.1630736
  5 27.32 2.38 1.01 0.74 1.01 0.07 28.22616 -0.9061594 27.05439 0.2656139
  6 23.23 2.97 0.46 0.96 2.36 0.12 36.10681 -12.8768143
                                                          20.92296 2.3070370
```

```
RMSE_lm
[1] 7.949631
RMSE_pcr
[1] 0.1121959
```

Interpretation:

RMSE using PCR is less than RMSE using linear regression, we may conclude that PCR model predicts SALES better than linear regression model when multicolinearity exists.

Quick Recap

Multiple Linear Regression and Multicollinearity

• Highly correlated predictor variables is a very frequent phenomenon in real world analytics.

Principal Component Regression • PCR is a three way process where the variables are first transformed to principal components, regression is run by considering these components as regressors and finally, they are transformed back to their original forms.

PCR in R

• pcr() function in package pls performs PCR.