

Graduation Rate Drivers – PCR vs PLS Modelling (R)

Pan Phyu Phyu Hmwe

Executive Summary

This project evaluates two modelling approaches, Principal Component Regression (PCR) and Partial Least Squares (PLS), to predict and understand graduation rate performance when many institutional variables are correlated. Using cross-validation and an out-of-sample test set, PLS achieved lower prediction error than PCR (lower test MSE), making it the preferred approach for performance monitoring and forecasting in high-dimensional settings.

Business Context

Graduation rate is commonly used as an outcome indicator for education performance monitoring and planning. When decision-makers have many correlated metrics (admissions, spending, staff ratios, student mix, etc.), traditional regression can become unstable due to multicollinearity. **PCR and PLS** address this by reducing dimensionality while retaining predictive signal, supporting more stable reporting, monitoring, and scenario exploration.

Objectives

- Build a reliable model to predict graduation rate with correlated predictors.
- Select an appropriate number of components to balance accuracy and simplicity.
- Compare PCR vs PLS and summarise implications for performance monitoring and decision-making.

Data

- **Dataset:** ISLR2 College dataset (777 institutions, 17 predictors)
- **Target:** Grad.Rate (graduation rate)
- **Pre-processing:** Convert Private to numeric (0/1) and scale predictors.

```
library(ISLR2)
```

```
Warning: package 'ISLR2' was built under R version 4.2.3
```

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```
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(tidyr)

Warning: package 'tidyr' was built under R version 4.2.3

library(pls)

Attaching package: 'pls'

The following object is masked from 'package:stats':

loadings

data("College")

college.df <- College

Convert 'Private' to numeric (Yes=1, No=0)

college.df\$Private <- as.character(college.df\$Private)

college.df\$Private <- ifelse(college.df\$Private == "Yes", 1, 0)

head(college.df)

	Private	Apps	Accept	Enroll	Top10perc	Top25perc
Abilene Christian University	1	1660	1232	721	23	52
Adelphi University	1	2186	1924	512	16	29
Adrian College	1	1428	1097	336	22	50
Agnes Scott College	1	417	349	137	60	89
Alaska Pacific University	1	193	146	55	16	44
Albertson College	1	587	479	158	38	62

	F.Undergrad	P.Undergrad	Outstate	Room.Board
--	-------------	-------------	----------	------------

Books

Abilene Christian University	2885	537	7440	3300
450				
Adelphi University	2683	1227	12280	6450
750				
Adrian College	1036	99	11250	3750
400				
Agnes Scott College	510	63	12960	5450
450				
Alaska Pacific University	249	869	7560	4120
800				
Albertson College	678	41	13500	3335
500				

	Personal	PhD	Terminal	S.F.Ratio	perc.alumni
Expend					
Abilene Christian University 7041	2200	70	78	18.1	12
Adelphi University 10527	1500	29	30	12.2	16
Adrian College 8735	1165	53	66	12.9	30
Agnes Scott College 19016	875	92	97	7.7	37
Alaska Pacific University 10922	1500	76	72	11.9	2
Albertson College 9727	675	67	73	9.4	11
	Grad.Rate				
Abilene Christian University	60				
Adelphi University	56				
Adrian College	54				
Agnes Scott College	59				
Alaska Pacific University	15				
Albertson College	55				

Method Overview

Validation design

- **Holdout evaluation:** 50/50 train-test split for final performance estimate.
- **Model selection:** Cross-validation (CV) on training data to select number of components.

```
set.seed(1)

# 50/50 split
train <- college.df %>% sample_frac(0.5)
test  <- college.df %>% setdiff(train)

# matrices for consistent prediction
x_test <- model.matrix(Grad.Rate ~ ., test)[, -1]
y_test <- test$Grad.Rate
```

Model 1: Principal Component Regression (PCR)

Cross-validation to select components

```
set.seed(1)
pcr_model <- pcr(Grad.Rate ~ ., data = college.df, scale = TRUE, validation = "CV")
summary(pcr_model)

Data:  X dimension: 777 17
      Y dimension: 777 1
Fit method: svdpc
```

Number of components considered: 17

VALIDATION: RMSEP

Cross-validated using 10 random segments.

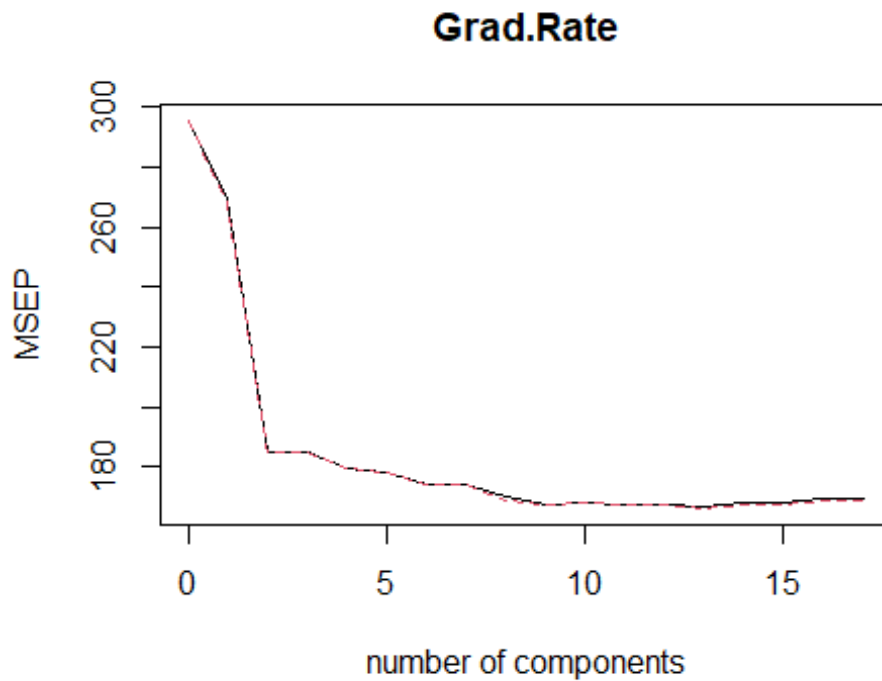
	(Intercept)	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps
CV	17.19	16.41	13.6	13.59	13.41	13.35	13.19
adjCV	17.19	16.36	13.6	13.59	13.40	13.35	13.19
	7 comps	8 comps	9 comps	10 comps	11 comps	12 comps	13 comps
CV	13.2	13.05	12.95	12.97	12.95	12.95	12.91
adjCV	13.2	12.99	12.94	12.96	12.94	12.94	12.89
	14 comps	15 comps	16 comps	17 comps			
CV	12.96	12.97	13.02	13.01			
adjCV	12.94	12.95	12.99	12.98			

TRAINING: % variance explained

	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps	7 comps
X	30.61	59.36	66.26	72.24	77.67	82.50	85.96
Grad.Rate	10.58	37.51	38.17	40.03	40.68	42.21	42.23
	8 comps	9 comps	10 comps	11 comps	12 comps	13 comps	14 comps
X	89.13	92.25	94.36	96.19	97.29	98.29	99.14
Grad.Rate	44.31	44.88	44.91	45.15	45.51	45.80	45.91
	15 comps	16 comps	17 comps				
X	99.65	99.86	100.00				
Grad.Rate	45.91	45.95	46.15				

CV error vs components

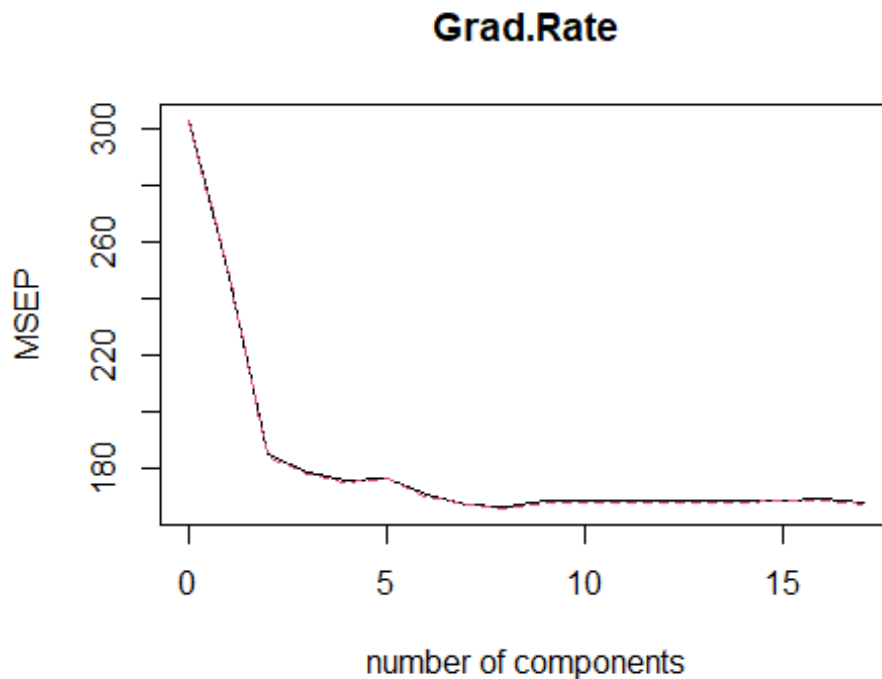
```
validationplot(pcr_model, val.type = "MSEP")
```



Interpretation (high level): CV error drops quickly with early components then levels off, suggesting diminishing returns after a small number of components.

Test performance (PCR)

```
# Fit PCR on training set
pcr_fit2 <- pcr(Grad.Rate ~ ., data = train, scale = TRUE, validation = "CV")
validationplot(pcr_fit2, val.type = "MSEP")
```



```
# Use 5 components (based on CV)
pcr_pred <- predict(pcr_fit2, x_test, ncomp = 5)
pcr_mse <- mean((pcr_pred - y_test)^2)
pcr_mse

[1] 181.3046
```

Model 2: Partial Least Squares (PLS)

Cross-validation to select components

```
pls_fit <- pls(Grad.Rate ~ ., data = train, scale = TRUE, validation = "CV")
summary(pls_fit)
```

```
Data:   X dimension: 388 17
       Y dimension: 388 1
Fit method: kernelpls
Number of components considered: 17
```

VALIDATION: RMSEP

Cross-validated using 10 random segments.

	(Intercept)	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps
CV	17.41	13.33	13.05	12.93	12.99	13.03	13.09
adjCV	17.41	13.33	13.05	12.92	12.97	13.00	13.05

	7 comps	8 comps	9 comps	10 comps	11 comps	12 comps	13 comps
CV	13.09	13.13	13.17	13.12	13.10	13.06	13.06
adjCV	13.06	13.09	13.12	13.08	13.06	13.03	13.03

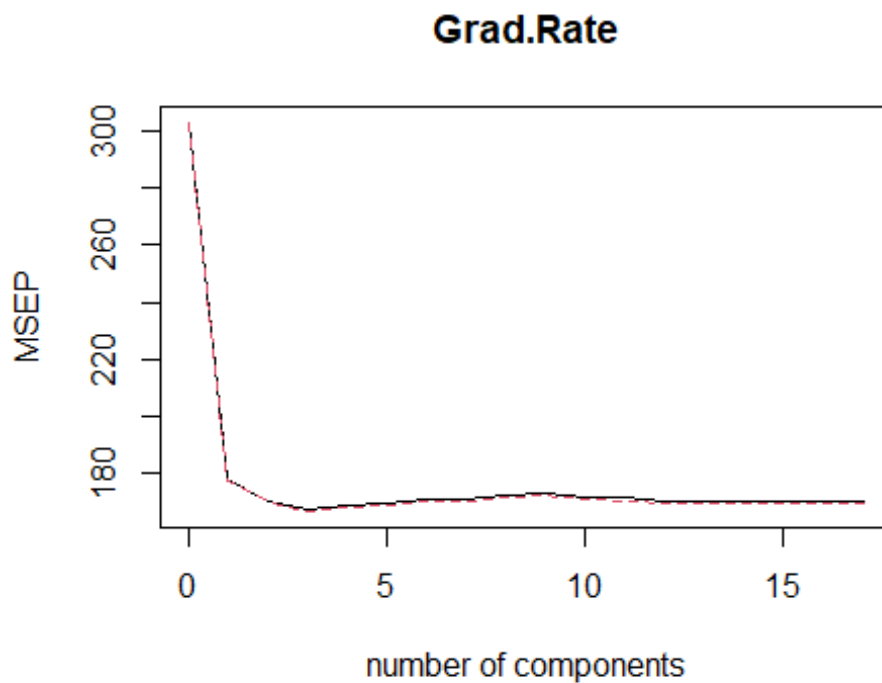
	14 comps	15 comps	16 comps	17 comps
CV	13.06	13.06	13.06	13.06
adjCV	13.03	13.03	13.03	13.03

CV	13.06	13.06	13.05	13.05
adjCV	13.02	13.02	13.02	13.02

TRAINING: % variance explained

	1 comps	2 comps	3 comps	4 comps	5 comps	6 comps	7 comps
X	29.78	49.71	65.49	70.22	74.28	77.61	81.56
Grad.Rate	41.86	45.26	47.33	47.84	47.99	48.10	48.17
	8 comps	9 comps	10 comps	11 comps	12 comps	13 comps	14 comps
X	83.35	86.05	89.27	92.17	93.34	95.36	96.69
Grad.Rate	48.34	48.46	48.52	48.53	48.53	48.53	48.53
	15 comps	16 comps	17 comps				
X	97.72	99.01	100.00				
Grad.Rate	48.53	48.53	48.53				

```
validationplot(pls_fit, val.type = "MSEP")
```



Test performance (PLS)

```
# Use 3 components (based on CV)
pls_pred <- predict(pls_fit, x_test, ncomp = 3)
pls_mse <- mean((pls_pred - y_test)^2)
pls_mse

[1] 170.0129
```

Key Results

- **PCR (5 components) test MSE:** 181.3046
- **PLS (3 components) test MSE:** 170.0129

Conclusion: PLS achieved a lower test MSE than PCR, suggesting better predictive performance, likely because PLS uses the response variable when deriving components.

What this means for decision-making

- **Better forecasting with fewer components:** PLS provides improved accuracy with a simpler representation, making it suitable for performance monitoring and reporting.
- **More stable modelling under multicollinearity:** Component-based regression helps reduce noise from redundant predictors and produces more reliable estimates.
- **Practical use:** These models can support monitoring dashboards (predicted vs actual), early-warning flags, and scenario exploration alongside other evidence.

Model performance and selection

Partial Least Squares (PLS) achieved a lower test Mean Squared Error than Principal Component Regression (PCR) while using fewer components, indicating stronger predictive performance with a simpler model structure. This demonstrates that incorporating the response variable when deriving components improves forecasting accuracy in high-dimensional datasets with correlated predictors.

Dimensionality reduction with retained analytical value

The final PLS model captured a substantial proportion of the variance in both the predictors and the graduation rate outcome using only three components. This shows that a small number of derived components can efficiently summarise complex institutional data while preserving the key information required for reliable prediction and analysis.

Stability in multicollinear environments

By transforming the original variables into orthogonal components, the model reduced the impact of multicollinearity and produced more stable and interpretable predictions. This approach is particularly suitable for performance monitoring scenarios where multiple related indicators are analysed simultaneously.

Implications for performance monitoring and planning

In a real-world context, the selected PLS model provides a practical and scalable method for forecasting graduation rate outcomes and tracking performance over time. The reduced model complexity supports clearer reporting, easier maintenance, and more consistent results, enabling data-driven planning and more effective identification of institutions that are over- or under-performing.

Recommendations for Next Steps

- Create stakeholder-ready outputs: a short insight summary, key visuals (error vs components), and a clear narrative of findings.
- If used for monitoring: retrain on updated data periodically, validate performance each cycle, and track data/model drift.
- Improve explainability: review component loadings and summarise which types of factors drive predictions.
- Optional: publish a simple dashboard (Tableau/Power BI) showing predicted vs actual graduation rate and summary KPIs.

Tools

R / RStudio (ISLR2, dplyr, tidyr, pls)