Cómputo Estadístico

November 7, 2024

Godinez Bravo Diego

Tarea 5 - Métodos de Contracción y Reducción de Dimensión

Centro de Investigación en Matemáticas

Maestría en Cómputo Estadístico

0.1 Problema 1

Utilizando el conjunto de datos `College` disponible en la libreria ISLR, predice el numero de solicitudes recibidas (Apps) utilizando las otras variables del conjunto de datos.

- Divide el conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba.
- Ajusta un modelo lineal utilizando mínimos cuadrados en el conjunto de entrenamiento y reporta el error de prueba obtenido.
- Ajusta un modelo de regresión Ridge en el conjunto de entrenamiento, con λ elegido mediante validación cruzada. Reporta el error de prueba obtenido.
- Ajusta un modelo Lasso en el conjunto de entrenamiento, con λ elegido mediante validación cruzada. Reporta el error de prueba obtenido, junto con el número de estimaciones de coeficientes diferentes de cero.
- Ajusta un modelo PCR en el conjunto de entrenamiento, con M elegido mediante validación cruzada. Reporta el error de prueba obtenido, junto con el valor de M seleccionado por validación cruzada.
- Ajusta un modelo PLS en el conjunto de entrenamiento, con M elegido mediante validación cruzada. Reporta el error de prueba obtenido, junto con el valor de M seleccionado por validación cruzada.
- Comenta sobre los resultados obtenidos. ¿Con cuánta precisión podemos predecir el número de solicitudes universitarias recibidas? ¿Hay mucha diferencia entre los errores de prueba resultantes de estos cinco enfoques?
- Propon un modelo (o un conjunto de modelos) que parezca funcionar bien en este conjunto de datos y justifica tu respuesta. Asegúrate de evaluar el rendimiento del modelo utilizando el error del conjunto de validación, la validación cruzada o alguna otra alternativa razonable, en lugar de utilizar el error de entrenamiento. ¿El modelo que elegiste incluye todas las características del conjunto de datos? ¿Por qué o por qué no?

0.2 Resultados

[7]:
```
set.seed(614)
```

[8]:
```
library(pls)
library(ISLR)
library(glmnet)
library(rsample) # loading libraries
```

Vista previa del conjunto de datos `College`, compuesto por **18 variables** y **777 observaciones**.

[10]:
```
colnames(College) # column names of College dataset
```

1. 'Private' 2. 'Apps' 3. 'Accept' 4. 'Enroll' 5. 'Top10perc' 6. 'Top25perc' 7. 'F.Undergrad' 8. 'P.Undergrad' 9. 'Outstate' 10. 'Room.Board' 11. 'Books' 12. 'Personal' 13. 'PhD' 14. 'Terminal' 15. 'S.F.Ratio' 16. 'perc.alumni' 17. 'Expend' 18. 'Grad.Rate'

[6]:
```
dim(College) # (777 rows, 18 columns)
```

1. 777 2. 18

Dividir el conjunto de datos en dos subconjuntos: uno de **entrenamiento**, que contenga el 80% de los datos, y otro de **prueba**, con el 20% restante.

[7]:
```
data_split <- initial_split(College, prop = .80)
train <- training(data_split)
test  <- testing(data_split) # split data into training (80Matrices de diseño␣
 ↪para el ajuste de modelos de regresión Ridge y Lasso.%) and testing set (20%)
```

[8]:
```
dim(train) # training set contains 621 observations
```

1. 621 2. 18

[9]:
```
dim(test) # testing set contains 156 observations
```

1. 156 2. 18

Matrices de diseño para el ajuste de modelos de regresión Ridge y Lasso.

[10]:
```
x_train <- model.matrix(Apps ~ ., data = train)[, -1] # exclude response␣
 ↪variable
y_train <- train$Apps # response variable
```

[11]:
```
x_test <- model.matrix(Apps ~ ., data = test)[, -1] # exclude response variable
y_test <- test$Apps # response variable
```

0.2.1 Ajuste del Modelo por Mínimos Cuadrados

[12]:
```
lm_model <- lm(Apps ~ ., data = train)
predictions <- predict(lm_model, test) # fit a linear regression model using␣
 ↪least squares method
```

[13]:
```
ols_test_error <- mean((test$Apps - predictions)^2) # test error
```

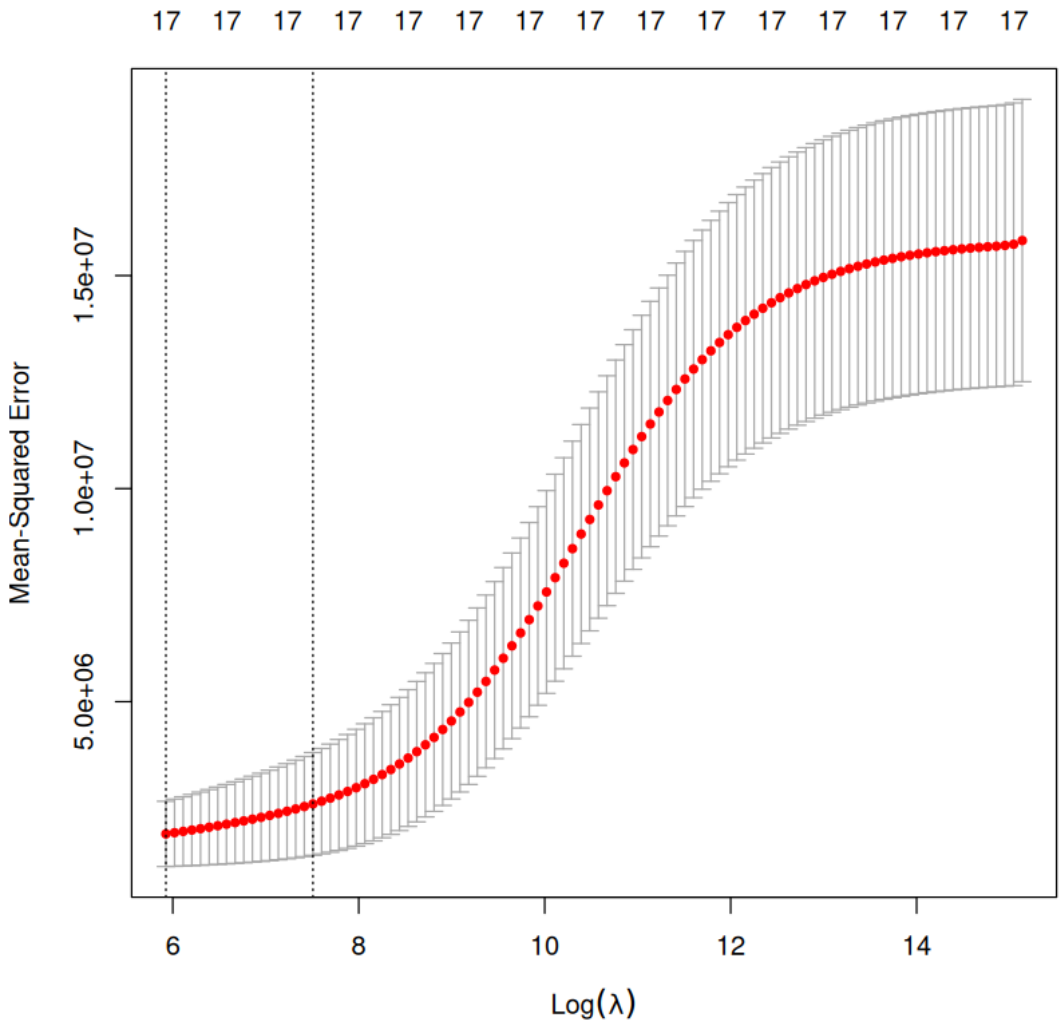
[14]:
```
cat("Error de prueba obtenido (OLS):", ols_test_error)
```

```
Error de prueba obtenido (OLS): 946141.8
```

0.2.2 Ajuste del Modelo de Regresión Ridge

[15]:
```
ridge_model <- glmnet(x_train, y_train, alpha = 0) # ridge regression model;␣
↪alpha parameter specifies ridge penalty
```

[16]:
```
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0)
plot(cv_ridge)
```



El parámetro λ controla la penalización aplicada al modelo:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=RSS+\lambda\sum_{j=1}^{p}\beta_j^2$$

donde el segundo término, $\lambda\sum_{j=1}^{p}\beta_j^2$, se conoce como **penalización por contracción**.

Cuando $\lambda = 0$, la penalización no tiene ningún efecto, por lo que la ecuación se reduce al modelo ajustado mediante mínimos cuadrados. En cambio, cuando $\lambda \to \infty$, los coeficientes se reducen a cero, lo que corresponde al modelo nulo.

[17]:
```
cat("Valor mínimo de lambda:", cv_ridge$lambda.min, log(cv_ridge$lambda.min))
```

```
Valor mínimo de lambda: 374.7353 5.92622
```

El valor mínimo de λ se alcanza aproximadamente cuando $\log(\lambda) \approx 6$, lo que resulta en una penalización mínima y un modelo cercano al ajustado mediante mínimos cuadrados. Sin embargo, dado que el objetivo es minimizar el error de predicción en el conjunto de prueba, se optó por elegir un valor de λ cercano a $\log(\lambda) \approx 8$. Este valor simplifica el modelo al reducir los coeficientes, lo que va a conducir a una mejora en la generalización en datos nuevos.

[18]:
```
cat("Valor óptimo de lambda:", cv_ridge$lambda.1se, log(cv_ridge$lambda.1se))
```

```
Valor óptimo de lambda: 1822.189 7.507794
```

[19]:
```
best_lambda <- cv_ridge$lambda.1se # choosen lambda value
```

[20]:
```
predictions <- predict(ridge_model, s = best_lambda, newx = x_test) # calculate␣
↪predictions using the choosen lambda value
```

[21]:
```
ridge_test_error <- mean((y_test - predictions)^2) # test error
```

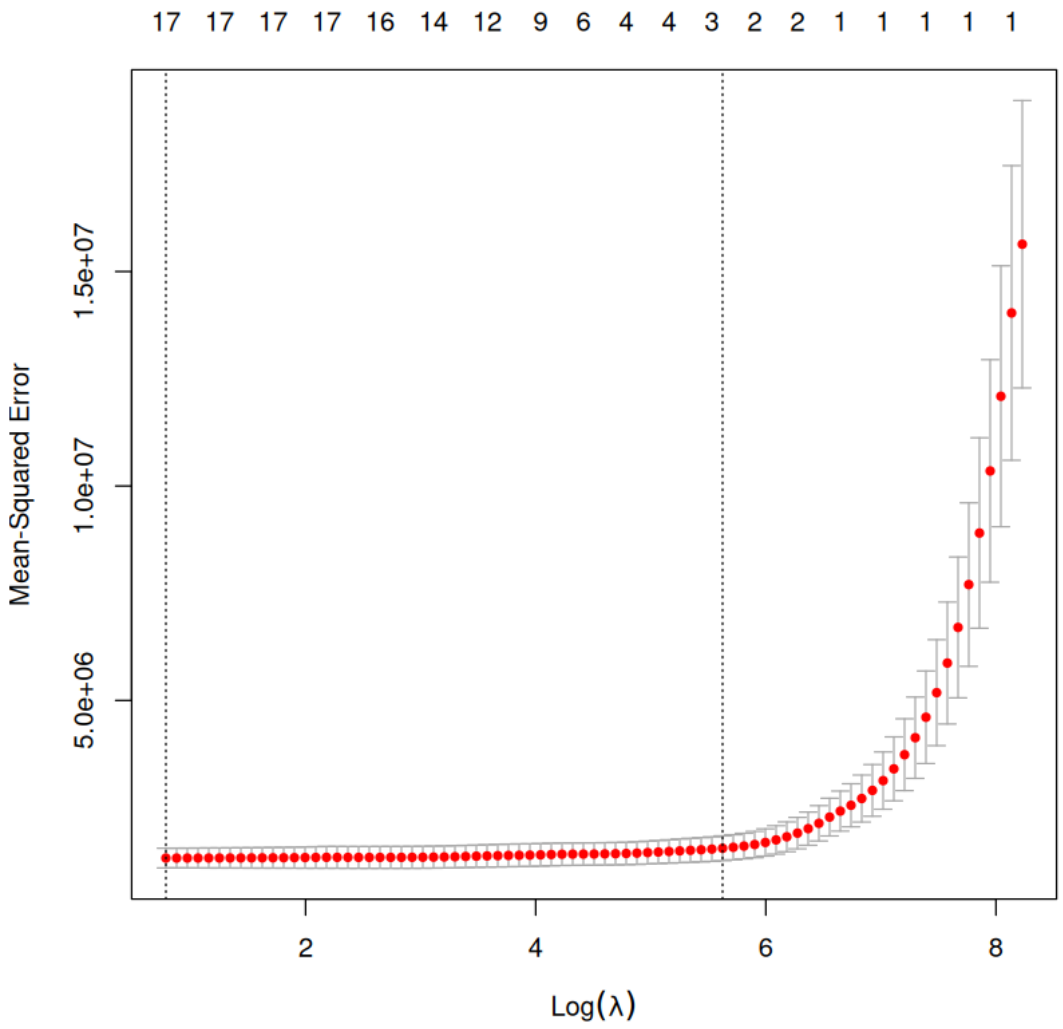
[22]:
```
cat("Error de prueba obtenido (Regresión Ridge):", ridge_test_error)
```

```
Error de prueba obtenido (Regresión Ridge): 1106708
```

0.2.3 Ajuste del Modelo de Regresión Lasso

[23]:
```
lasso_model <- glmnet(x_train, y_train, alpha = 1) # lasso regression model;␣
↪alpha parameter specifies lasso penalty
```

[24]:
```
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1)
plot(cv_lasso)
```



Al igual que en la regresión Ridge, el parámetro λ controla la intensidad de la penalización. En este caso, la función que se minimiza en la regresión Lasso es:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

donde en el término de penalización, se reemplaza β_j^2 por $|\beta_j|$.

De manera similar, cuando $\lambda = 0$, el modelo se ajusta simplemente mediante mínimos cuadrados, y cuando $\lambda \to \infty$, el modelo se reduce al modelo nulo, donde todos los coeficientes son iguales a cero.

[25]:
```
cat("Valor mínimo de lambda:", cv_lasso$lambda.min, log(cv_lasso$lambda.min))
```

```
Valor mínimo de lambda: 2.194833 0.7861058
```

[26]:
```
lasso_coef <- predict(lasso_model, type = "coefficients", s = cv_lasso$lambda.
↪min)[1:17,]
```

[27]:
```
cat("Coeficientes/Estimaciones diferentes de 0:", length(lasso_coef[lasso_coef !
↪= 0]))
```

```
Coeficientes/Estimaciones diferentes de 0: 17
```

[28]:
```
cat("Valor óptimo de lambda:", cv_lasso$lambda.1se, log(cv_lasso$lambda.1se))
```

```
Valor óptimo de lambda: 276.9565 5.62386
```

[29]:
```
lasso_coef <- predict(lasso_model, type = "coefficients", s = cv_lasso$lambda.
↪1se)[1:17,]
```

[30]:
```
cat("Coeficientes/Estimaciones diferentes de 0:", length(lasso_coef[lasso_coef !
↪= 0]))
```

```
Coeficientes/Estimaciones diferentes de 0: 4
```

Se optó por seleccionar un valor de λ cercano a $\log(\lambda) \approx 6$. Aunque este valor no minimiza el MSE, reduce significativamente el número de predictores, lo que mejora la interpretabilidad del modelo y su capacidad de generalización a datos no vistos. Además, el aumento en el MSE es mínimo en comparación con el error obtenido con el valor de λ mínimo.

[31]:
```
best_lambda <- cv_lasso$lambda.1se # choosen lambda value
```

[32]:
```
predictions <- predict(lasso_model, s = best_lambda, newx = x_test) # calculate␣
↪predictions using the choosen lambda value
```

[33]:
```
lasso_test_error <- mean((y_test - predictions)^2) # test error
```

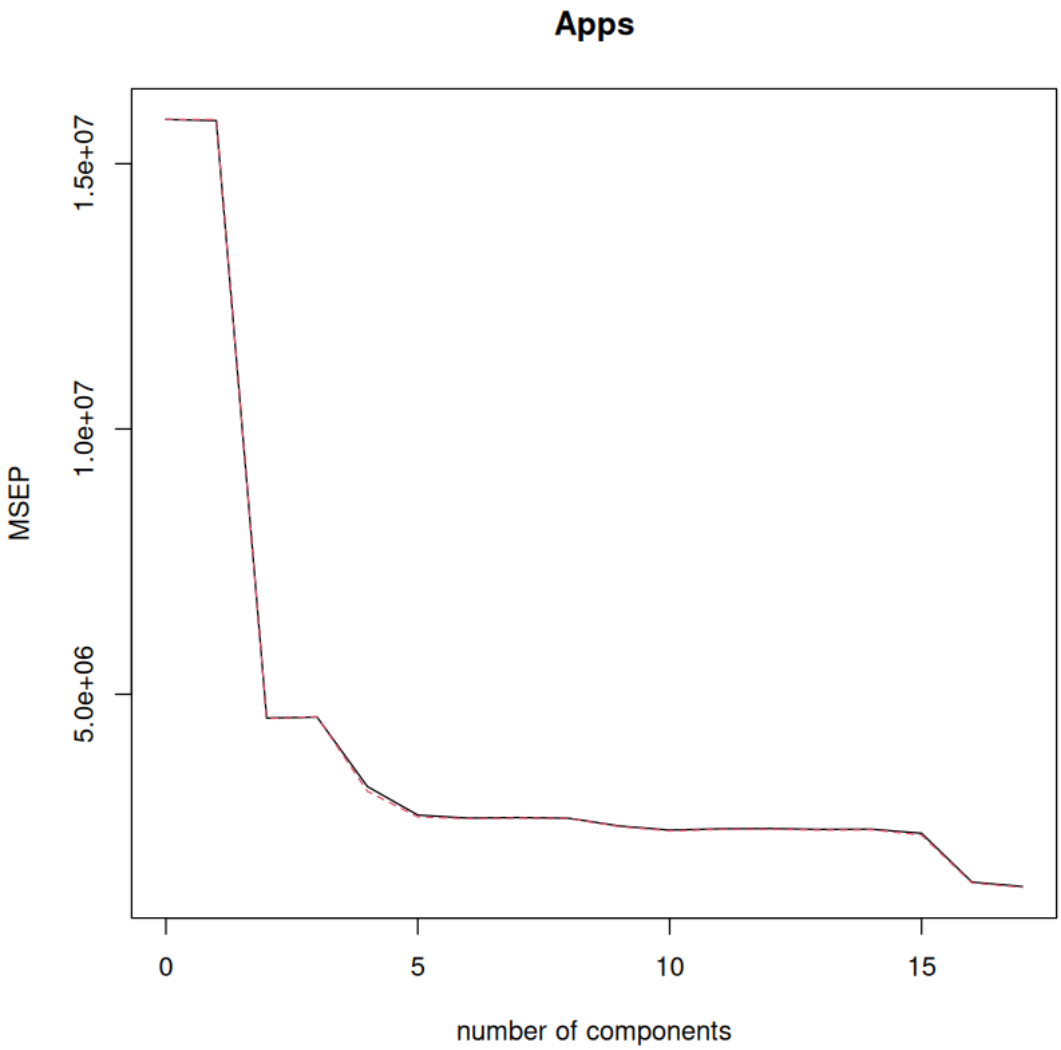
[34]:
```
cat("Error de prueba obtenido (Regresión Lasso):", lasso_test_error)
```

```
Error de prueba obtenido (Regresión Lasso): 854761.4
```

0.2.4 Ajuste del Modelo de Regresión por Componentes Principales

[35]:
```
pcr_model <- pcr(Apps ~ ., data = train, scale = TRUE, validation = 'CV') #␣
↪define a principal components regression model
```

[36]:
```
validationplot(pcr_model, val.type = 'MSEP') # plot the mean squared error of␣
↪prediction (MSEP)
```



Observamos que, a partir de 5 componentes, el modelo no muestra una mejora significativa en su capacidad de predicción. Incluir más de 5 componentes no sería recomendable, ya que solo aumentaría la complejidad de forma innecesaria. Un modelo con aproximadamente $3 - 5$ componentes resulta óptimo para este caso.

[37]:
```
predictions <- predict(pcr_model, test, ncomp = 5) # calculate predictions␣
↪using M = 5
```

[38]:
```
pcr_test_error <- mean((test$Apps - predictions)^2) # test error
```

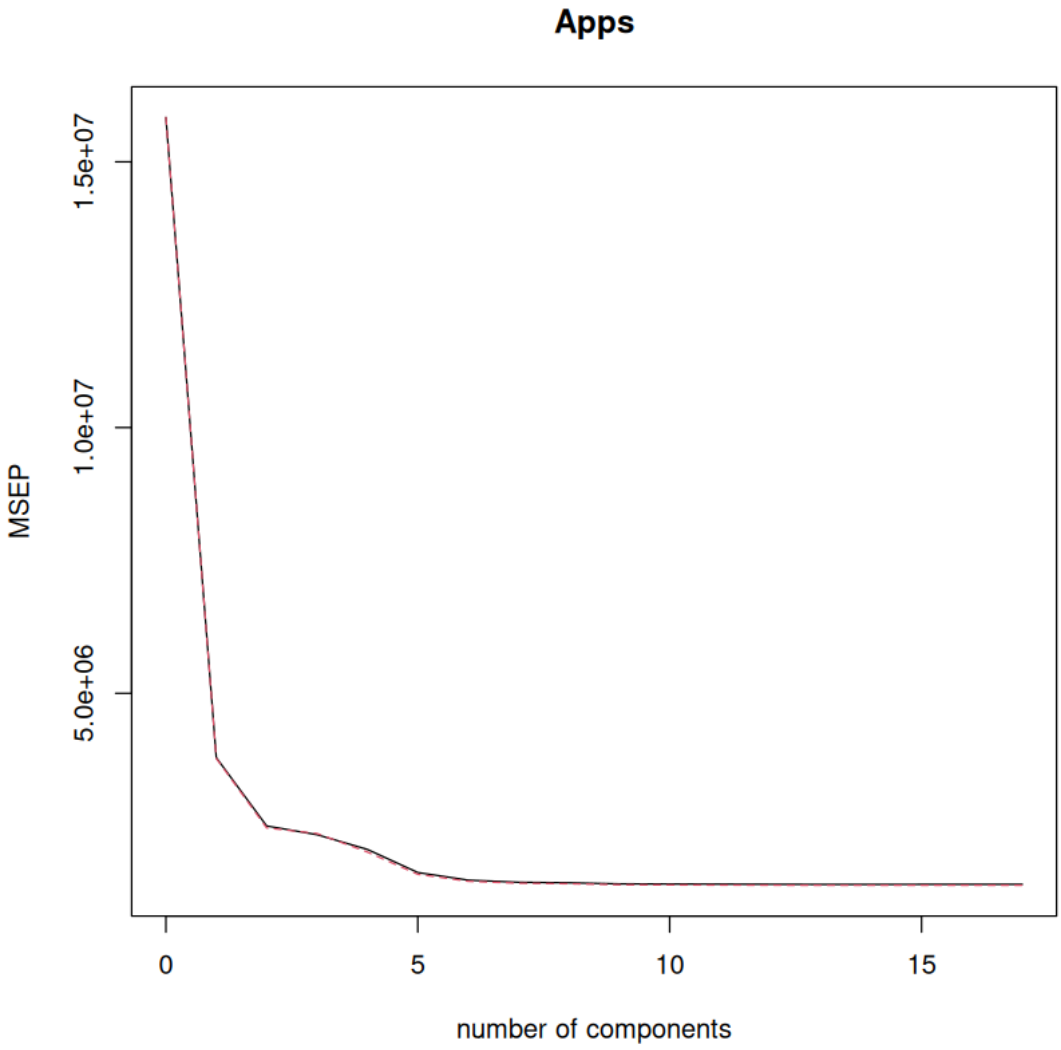
[39]:
```
cat("Error de prueba obtenido (PCR):", pcr_test_error)
```

```
Error de prueba obtenido (PCR): 1889150
```

0.2.5 Ajuste del Modelo de Regresión por Mínimos Cuadrados Parciales

[40]:
```
pls_model <- plsr(Apps ~ ., data = train, scale = TRUE, validation = 'CV') #␣
↪define a partial least squares regression model
```

[41]:
```
validationplot(pls_model, val.type = 'MSEP') # plot the mean squared error of␣
↪prediction (MSEP)
```



Inicialmente, al aumentar el número de componentes, mejora la capacidad de predicción del modelo y se reduce el error de prueba. Sin embargo, entre 5 y 7 componentes, el error de prueba se estabiliza y no se observa una disminución significativa adicional. Por lo tanto, en este modelo PLS, seleccionar $M = 6$ componentes sería óptimo, evitando así un aumento innecesario en la complejidad del modelo.

[42]:
```
predictions <- predict(pls_model, test, ncomp = 6) # calculate predictions␣
↪using M = 6
```

[43]:
```
pls_test_error <- mean((test$Apps - predictions)^2) # test error
```

[44]:
```
cat("Error de prueba obtenido (PLS):", pls_test_error)
```

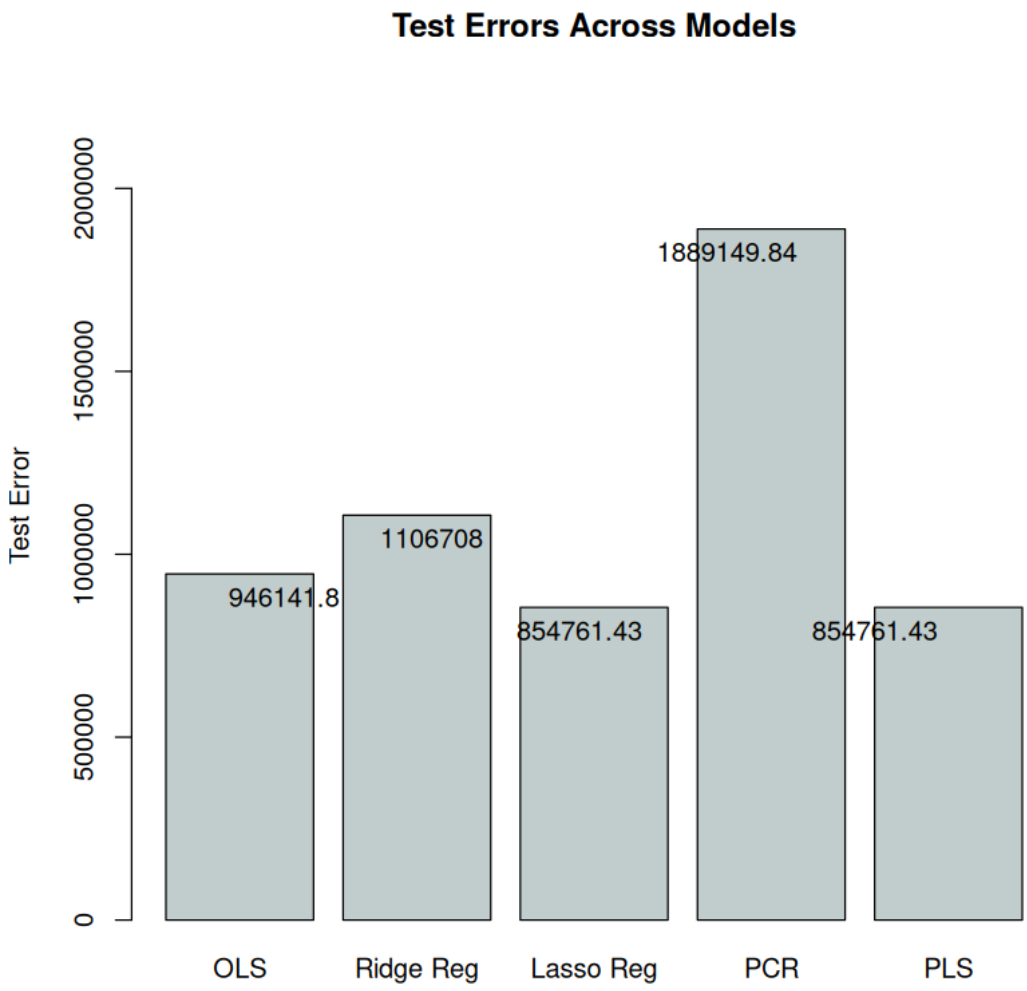
```
Error de prueba obtenido (PLS): 870684.1
```

0.2.6 Comparación del Error de Prueba entre Modelos

[45]:
```
test_errors <- c(ols_test_error, ridge_test_error, lasso_test_error,␣
↪pcr_test_error, lasso_test_error) # test error obtained
models <- c('OLS', 'Ridge Reg', 'Lasso Reg', 'PCR', 'PLS') # models fitted
```

[65]:
```
barplot(test_errors, names.arg = models, main = 'Test Errors Across Models',
        ylab = 'Test Error', col = 'Azure 3', ylim = c(0, max(test_errors) * 1.
↪2))

text(x = seq_along(test_errors), y = test_errors, label = round(test_errors,␣
↪2), pos = 1) # show test error for each bar
```



El modelo de **regresión Lasso** presenta el menor error de prueba, seguido por el modelo PLS y el ajustado por mínimos cuadrados. Esto sugiere que, en este caso específico, el uso de estos modelos mejora la capacidad de predicción en comparación con el modelo ajustado por mínimos cuadrados. Por otro lado, los modelos de regresión Ridge y PCR tienen un error de prueba mayor al del modelo OLS, por lo que no se recomendaría su uso en este caso particular.

Cabe destacar que, en cada uno de los casos, se evitó utilizar el mayor número de componentes o el valor mínimo de λ, ya que esto resultaría en modelos muy similares al ajustado mediante OLS, lo que haría que la comparación fuera irrelevante.

0.2.7 Modelo Propuesto

Con base en los resultados previos, recomendaría utilizar un modelo de `Regresión Lasso` con un valor de $\lambda \approx 276.9$, ya que presenta un error de prueba menor que el modelo ajustado por mínimos

cuadrados. Este método de contracción reduce significativamente la varianza de los coeficientes y disminuye el número de variables, lo cual aporta a la interpretabilidad del modelo. En contraste con los modelos de regresión Ridge y PCR, que incluyen todos los predictores en sus componentes, lo que dificulta su interpretación.

[71]:
```
cat("Error de prueba obtenido mediante el modelo de regresión Lasso:",␣
↪lasso_test_error)
```

```
Error de prueba obtenido mediante el modelo de regresión Lasso: 854761.4
```

En este caso, el modelo propuesto incluye únicamente 3 predictores, además del intercepto (estimaciones distintas de cero). Los predictores considerados son: **Accept** (número de aplicaciones aceptadas), **Top10perc** (porcentaje de nuevos estudiantes dentro del 10% superior de su clase) y **Expend** (gasto educativo por estudiante).

[70]:
```
cat("No. de predictores incluidos en el modelo:", length(lasso_coef[lasso_coef !
↪= 0]))
```

```
No. de predictores incluidos en el modelo: 4
```

[74]:
```
lasso_coef[lasso_coef != 0] # coefficient estimates and their respective␣
↪predictors
```

(Intercept) -399.162472751937 **Accept** 1.37026604373971 **Top10perc** 22.892130402479 **Expend** 0.00441847156889482

Al analizar los predictores de este modelo, podemos intuir que estas variables capturan aspectos relevantes de los datos. Un alto número de aplicaciones aceptadas (**Accept**) podría hacer que la institución sea más atractiva para futuros aspirantes, incrementando el número de solicitudes recibidas. Además, un alto porcentaje de alumnos de alto rendimiento (**Top10perc**) podría reflejar la calidad educativa de la institución, atrayendo a más aplicantes interesados. Por último, un mayor gasto por estudiante (**Expend**) podría asociarse con más recursos y mejores instalaciones, aumentando el atractivo de la institución para los aspirantes.

0.3 Problema 2

Es bien sabido que la regresión Ridge tiende a dar valores similares de coeficientes a las variables correlacionadas, mientras que Lasso puede dar valores de coeficientes totalmente diferentes a las variables correlacionadas. Ahora exploraremos esta propiedad en un entorno sencillo.

Supongamos que $n = 2, p = 2, x_{11} = x_{12}, x_{21} = x_{22}$. Además, supongamos que $y_1 + y_2 = 0$ y $x_{11} + x_{21} = 0$ y $x_{12} + x_{22} = 0$, de modo que el estimado para la intersección en un modelo de mínimos cuadrados, regresión de crestas o lasso es cero: $\hat{\gamma} = 0$.

- Plantea el problema de optimización con la regresión Ridge bajo estas suposiciones.
- Argumente que bajo estas suposiciones, las estimaciones de los coeficientes de Ridge satisfacen $\hat{\beta}_1 = \hat{\beta}_2$.
- Plantea el problema de optimización con la regresión Lasso bajo estas suposiciones.
- Argumenta que en este contexto, los coeficientes de lasso $\hat{\beta}_1$ y $\hat{\beta}_2$ no son únicos; es decir, hay muchas soluciones posibles al problema de optimización en (c). Describa estas soluciones.

0.4 Solución

0.4.1 Problema de Optimización Regresión Ridge

Los estimadores de los coeficientes en la regresión Ridge son aquellos valores que buscan minimizar:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

donde el segundo término, $\lambda\sum_{j=1}^{p}\beta_j^2$, se conoce como `penalización por contracción`.

Considerando $n = p = 2$.

$$\sum_{i=1}^{2}\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{2}\hat{\beta}_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{2}\hat{\beta}_j^2$$

$$(y_1 - \hat{\beta}_1 x_{11} - \hat{\beta}_2 x_{12})^2 + (y_2 - \hat{\beta}_1 x_{21} - \hat{\beta}_2 x_{22})^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)$$

Dado que $x_{11} = x_{12}, x_{21} = x_{22}$.

$$(y_1 - (\hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{11}))^2 + (y_2 - (\hat{\beta}_1 x_{22} + \hat{\beta}_2 x_{22}))^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)$$

Dado que el problema de optimización consiste en minimizar la ecuación:

$$min\left[(y_1 - (\hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{11}))^2 + (y_2 - (\hat{\beta}_1 x_{22} + \hat{\beta}_2 x_{22}))^2 + \lambda(\hat{\beta}_1^2 + \hat{\beta}_2^2)\right],$$

se calculan las derivadas parciales con respecto a $\hat{\beta}_1$ y $\hat{\beta}_2$, y se igualan a 0 para encontrar los puntos críticos.

Expandimos los términos al cuadrado:

$$(y_1 - (\hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{11}))^2 = y_1^2 - 2y_1(\hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{11}) + \hat{\beta}_1^2 x_{11}^2 + 2\hat{\beta}_1\hat{\beta}_2 x_{11}^2 + \hat{\beta}_2^2 x_{11}^2$$

$$(y_2 - (\hat{\beta}_1 x_{22} + \hat{\beta}_2 x_{22}))^2 = y_2^2 - 2y_2(\hat{\beta}_1 x_{22} + \hat{\beta}_2 x_{22}) + \hat{\beta}_1^2 x_{22}^2 + 2\hat{\beta}_1\hat{\beta}_2 x_{22}^2 + \hat{\beta}_2^2 x_{22}^2$$

Calculamos la derivada parcial con respecto a $\hat{\beta}_1$.

$$-2y_1 x_{11} - 2y_2 x_{22} + 2\hat{\beta}_1 x_{11}^2 + 2\hat{\beta}_1 x_{22}^2 + 2\hat{\beta}_2 x_{11}^2 + 2\hat{\beta}_2 x_{22}^2 + 2\hat{\beta}_1\lambda$$

$$-2y_1 x_{11} - 2y_2 x_{22} + 2\hat{\beta}_1(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_2(x_{11}^2 + x_{22}^2) = 0$$

Calculamos la derivada parcial con respecto a $\hat{\beta}_2$.

$$-2y_1 x_{11} - 2y_2 x_{22} + 2\hat{\beta}_2 x_{11}^2 + 2\hat{\beta}_2 x_{22}^2 + 2\hat{\beta}_1 x_{11}^2 + 2\hat{\beta}_1 x_{22}^2 + 2\hat{\beta}_2\lambda$$

$$-2y_1 x_{11} - 2y_2 x_{22} + 2\hat{\beta}_2(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_1(x_{11}^2 + x_{22}^2) = 0$$

Dado que:

$$2y_1 x_{11} + 2y_2 x_{22} = 2\hat{\beta}_1(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_2(x_{11}^2 + x_{22}^2)$$

y

$$2y_1 x_{11} + 2y_2 x_{22} = 2\hat{\beta}_2(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_1(x_{11}^2 + x_{22}^2)$$

Entonces:

$$2\hat{\beta}_1(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_2(x_{11}^2 + x_{22}^2) = 2\hat{\beta}_2(x_{11}^2 + x_{22}^2 + \lambda) + 2\hat{\beta}_1(x_{11}^2 + x_{22}^2)$$

De la eq. anterior se observa que:

$$2\hat{\beta}_1\lambda = 2\hat{\beta}_2\lambda$$

$$\hat{\beta}_1 = \hat{\beta}_2$$

0.4.2 Problema de Optimización Regresión Lasso

Los estimadores de los coeficientes en la regresión Lasso buscan minimizar la ecuación:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j|=RSS+\lambda\sum_{j=1}^{p}|\beta_j|$$

donde en el término de penalización, se reemplaza β_j^2 por $|\beta_j|$.

Considerando $n=p=2$.

$$\sum_{i=1}^{2}\left(y_i-\hat{\beta}_0-\sum_{j=1}^{2}\hat{\beta}_j x_{ij}\right)^2+\lambda\sum_{j=1}^{2}|\hat{\beta}_j|$$

$$(y_1-\hat{\beta}_1x_{11}-\hat{\beta}_2x_{12})^2+(y_2-\hat{\beta}_1x_{21}-\hat{\beta}_2x_{22})^2+\lambda(|\hat{\beta}_1|+|\hat{\beta}_2|)$$

Dado que $x_{11}=x_{12}, x_{21}=x_{22}$.

$$(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11}))^2+(y_2-(\hat{\beta}_1x_{22}+\hat{\beta}_2x_{22}))^2+\lambda(|\hat{\beta}_1|+|\hat{\beta}_2|)$$

El problema de optimización consiste en minimizar la ecuación:

$$min\left[(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11}))^2+(y_2-(\hat{\beta}_1x_{22}+\hat{\beta}_2x_{22}))^2+\lambda(|\hat{\beta}_1|+|\hat{\beta}_2|)\right]$$

Considerando la siguiente formulación para la regresión Lassso:

$$min\left\{(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11}))^2+(y_2-(\hat{\beta}_1x_{22}+\hat{\beta}_2x_{22}))^2\right\}\text{ sujeto a }|\hat{\beta}_1|+|\hat{\beta}_2|\leq s$$

Geométricamente, la restricción Lasso tiene la forma de un rombo centrado en el origen del plano, que intersecta los ejes a una distancia s del origen.

Considerando que $y_2=-y_1$ y $x_{22}=-x_{11}$, reescribimos la ecuación como sigue:

$$(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})^2+(-y_1-\hat{\beta}_1(-x_{11})-\hat{\beta}_2(-x_{11}))^2$$

$$(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})^2+(y_1-(\hat{\beta}_1+\hat{\beta}_2)x_{11})^2$$

$$(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})^2+(y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})^2=2[y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})]^2$$

Debemos minimizar la expresión:

$$2[y_1-(\hat{\beta}_1x_{11}+\hat{\beta}_2x_{11})]^2\geq 0$$

Dada la expresión obtenida, podemos ver que esta siempre es mayor o igual a 0. Por lo tanto, la solucion optima se encontrará cuando

$$2[y_1 - (\hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{11})]^2 = 0$$

Expresando la ecuación de la siguiente manera:

$$2[y_1 - (\hat{\beta}_1 + \hat{\beta}_2) x_{11}]^2 = 0$$

Podemos ver que la solución se da cuando $\hat{\beta}_1 + \hat{\beta}_2 = \frac{y_1}{x_{11}}$.

Esta solución $\hat{\beta}_1 + \hat{\beta}_2 = \frac{y_1}{x_1}$ sugiere que existen múltiples soluciones para los coeficientes $\hat{\beta}_1$ y $\hat{\beta}_2$. Ya que la ecuación solo impone una restricción sobre la suma de $\hat{\beta}_1$ y $\hat{\beta}_2$, no establece valores individuales para cada uno de los coeficientes. Es decir, la relación que describe la solución es una ecuación lineal entre $\hat{\beta}_1$ y $\hat{\beta}_2$, y cualquier par de valores de $\hat{\beta}_1$ y $\hat{\beta}_2$ que sumen $\frac{y_1}{x_1}$ puede ser una solución válida.