

Ejercicio Integrador Final

Jennifer Ortega

Tabla de Contenidos

Descripción del Problema	1
Datos	2
Preprocesamiento de los Datos	2
Particionamiento	2
Regresión Lineal con 2 Variables Explicativas	2
Creación de receta (Incluir Feature Engineering)	2
Aplicación de receta	3
Especificación del modelo	3
Entrenamiento del modelo	3
Aplicación del modelo a datos de testeo	4
Evaluación del modelo	4
Regresión Lineal con 1 Variables Explicativas	5
Creación de receta (Incluir Feature Engineering)	5
Aplicación de receta	5
Especificación del modelo	5
Entrenamiento del modelo	5
Aplicación del modelo a datos de testeo	6
Evaluación del modelo	6
Referencias Bibliográficas	7

Descripción del Problema

Se intentará predecir si un estudiante logrará aprobar el **Examen Final** de la materia de Mecánica Newtonia, de acuerdo a las notas que obtuvo a lo largo de todo el semestre en los diferentes exámenes.

El conjunto de datos pertenece al curso de Mecánica Newtoniana de primer semestre de la Escuela Politécnica Nacional en el ciclo 2022B.

Datos

Los datos pertenecen a **238** Estudiantes de primer semestre de la Escuela Politécnica Nacional. Las calificaciones corresponden al primer examen, al segundo examen y al examen final, **en el presente proyecto se intentará predecir cual será la calificación que obtendra el estudiante en el examen final.**

Nombre_Estudiante	Primer_Examen	Segundo_Examen	Examen_Final
Length:238	Min. :0.000	Min. :0.240	Min. :1.860
Class :character	1st Qu.:1.500	1st Qu.:3.090	1st Qu.:4.030
Mode :character	Median :2.250	Median :3.770	Median :4.883
	Mean :2.393	Mean :3.775	Mean :4.902
	3rd Qu.:3.000	3rd Qu.:4.440	3rd Qu.:5.815
	Max. :6.500	Max. :7.390	Max. :9.303

Preprocesamiento de los Datos

Particionamiento

Primero realizaremos el Particionamiento de los datos, para lo cual dividiremos a los datos en una proporción de 75% y 25% para los datos de entrenamiento y prueba, respectivamente.

```
set.seed(500)
conf_particionam <- initial_split(Notas, prop = 0.75)
Notas_training <- training(conf_particionam)
Notas_testing <- testing(conf_particionam)
```

Regresión Lineal con 2 Variables Explicativas

Creación de receta (Incluir Feature Engineering)

Seleccionamos las variables que nos sirvan para crear el modelo, por lo que para el análisis el valor que no aporta ninguna información relevante es el nombre del estudiantes, por lo que esta variable será descartada de nuestro conjunto de datos a analizar. Además, se seleccionará la variable que va a ser explicada con el modelo que en este caso son las calificaciones del **Examen Final**, de acuerdo a las calificaciones obtenidas en el primer y segundo examen.

Aplicación de receta

Rows: 178

Columns: 3

```
$ Primer_Examen <dbl> -1.14042120, -1.14042120, 0.05954909, 2.85947977, -0.34~  
$ Segundo_Examen <dbl> 0.61168480, -0.22876565, -0.58895870, 0.99958501, -2.68~  
$ Examen_Final <dbl> 4.9600, 2.2875, 4.8450, 5.3500, 5.2325, 9.3025, 6.2800,~
```

Rows: 60

Columns: 3

```
$ Primer_Examen <dbl> 1.45951443, -0.74043111, -0.74043111, 0.05954909, -1.54~  
$ Segundo_Examen <dbl> 0.9718778, 0.6116848, 0.9718778, -0.4504229, -0.4134800~  
$ Examen_Final <dbl> 5.5825, 6.2800, 4.0700, 2.9075, 2.9850, 6.1625, 6.0075,~
```

Especificación del modelo

Usaremos regresión lineal (Fox & Weisberg (2018)).

Linear Regression Model Specification (regression)

Computational engine: lm

Model fit template:

```
stats::lm(formula = missing_arg(), data = missing_arg(), weights = missing_arg())
```

Entrenamiento del modelo

Call:

```
stats::lm(formula = Examen_Final ~ ., data = data)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.6386	-0.7937	0.0478	0.7427	3.2775

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.03001	0.09495	52.976	< 2e-16 ***
Primer_Examen	0.04794	0.10044	0.477	0.63373
Segundo_Examen	0.30887	0.10044	3.075	0.00244 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.267 on 175 degrees of freedom
Multiple R-squared: 0.06325, Adjusted R-squared: 0.05255
F-statistic: 5.908 on 2 and 175 DF, p-value: 0.003288

Aplicación del modelo a datos de testeo

Vamos a ver la predicción según el modelo.

```
# A tibble: 60 x 5
  Primer_Examen Segundo_Examen Examen_Final .pred .resid
      <dbl>         <dbl>         <dbl> <dbl> <dbl>
1      1.46         0.972         5.58  5.40  0.182
2     -0.740        0.612         6.28  5.18  1.10
3     -0.740        0.972         4.07  5.29 -1.22
4      0.0595     -0.450         2.91  4.89 -1.99
5     -1.54       -0.413         2.98  4.83 -1.84
6     -0.340     -0.192         6.16  4.95  1.21
7     -0.340     -0.275         6.01  4.93  1.08
8      0.860        0.612         4.54  5.26 -0.725
9      0.0595        0.215         4.77  5.10 -0.332
10     0.460        1.06         4.18  5.38 -1.20
# i 50 more rows
```

Evaluación del modelo

```
# A tibble: 3 x 3
  .metric .estimator .estimate
    <chr>   <chr>         <dbl>
1 rmse    standard      1.31
2 rsq     standard      0.0970
3 mae     standard      1.11
```

Según los resultados encontrados, podemos observar que las variables no son explicativas.

Regresión Lineal con 1 Variables Explicativas

Creación de receta (Incluir Feature Engineering)

Aplicación de receta

```
Rows: 178
Columns: 2
$ Segundo_Examen <dbl> 0.61168480, -0.22876565, -0.58895870, 0.99958501, -2.68~
$ Examen_Final   <dbl> 4.9600, 2.2875, 4.8450, 5.3500, 5.2325, 9.3025, 6.2800,~
```

```
Rows: 60
Columns: 2
$ Segundo_Examen <dbl> 0.9718778, 0.6116848, 0.9718778, -0.4504229, -0.4134800~
$ Examen_Final   <dbl> 5.5825, 6.2800, 4.0700, 2.9075, 2.9850, 6.1625, 6.0075,~
```

Especificación del modelo

Usaremos regresión lineal (Fox & Weisberg (2018)).

Linear Regression Model Specification (regression)

Computational engine: lm

Model fit template:

```
stats::lm(formula = missing_arg(), data = missing_arg(), weights = missing_arg())
```

Entrenamiento del modelo

Call:

```
stats::lm(formula = Examen_Final ~ ., data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.6684	-0.7941	0.0387	0.7638	3.1911

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.03001	0.09474	53.093	< 2e-16 ***
Segundo_Examen	0.32414	0.09501	3.412	0.000801 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.264 on 176 degrees of freedom

Multiple R-squared: 0.06203, Adjusted R-squared: 0.0567

F-statistic: 11.64 on 1 and 176 DF, p-value: 0.0008008

Aplicación del modelo a datos de testeo

Vamos a ver la predicción según el modelo.

```
# A tibble: 60 x 4
  Segundo_Examen Examen_Final .pred .resid
      <dbl>         <dbl> <dbl> <dbl>
1      0.972         5.58  5.35  0.237
2      0.612         6.28  5.23  1.05
3      0.972         4.07  5.35 -1.28
4     -0.450         2.91  4.88 -1.98
5     -0.413         2.98  4.90 -1.91
6     -0.192         6.16  4.97  1.19
7     -0.275         6.01  4.94  1.07
8      0.612         4.54  5.23 -0.693
9      0.215         4.77  5.10 -0.332
10     1.06          4.18  5.37 -1.19
# i 50 more rows
```

Evaluación del modelo

```
# A tibble: 3 x 3
  .metric .estimator .estimate
    <chr>    <chr>         <dbl>
1 rmse    standard      1.32
2 rsq     standard      0.0969
3 mae     standard      1.11
```

Según los resultados encontrados, podemos observar que las variables no son explicativas.

Referencias Bibliográficas

Fox, J., & Weisberg, S. (2018). *An r companion to applied regression*. Sage publications.
Search parsnip models. (n.d.). In *Explore tidymodels*. <https://www.tidymodels.org/find/parsnip/>