Modelo Predictivo de hábitos estudiantiles y rendimiento académico

Modelo Predictivo de hábitos estudiantiles y rendimiento académico

Descripción general del dataset

El dataset contiene **1000 registros de estudiantes**, cada uno con información sobre hábitos diarios, condiciones personales y rendimiento académico. En total, hay **16 variables**, tanto numéricas como categóricas. Estas variables incluyen aspectos como edad, género, horas de estudio, uso de redes sociales, calidad del sueño, ejercicio, salud mental y nota obtenida en el examen final.

Objetivos

General

Desarrollar un modelo predictivo capaz de estimar el rendimiento académico de los estudiantes, medido a través de la nota del examen final, en función de sus hábitos de estudio, estilo de vida y condiciones personales.

Especificos

- Explorar y analizar los datos disponibles sobre hábitos estudiantiles, bienestar y rendimiento académico.
- Identificar las variables que tienen mayor correlación con el puntaje en el examen, como horas de estudio, sueño, uso de redes sociales, entre otras.
- Preprocesar los datos para su uso en modelos predictivos (tratamiento de valores nulos, codificación de variables categóricas, normalización, etc.).

- Interpretar los resultados del modelo para comprender la importancia relativa de cada hábito o variable en la predicción del rendimiento académico.
- ¿Que función propongo? Una función que tome un dataframe, realice una imputacion, saque valores extremos, codificar a numeros ("encoder") y dejar en la clase que requiere el ML.
- Pensar para el examen la función que da la predección final.

Importar dataset

```
rendimiento_estudiantes_inmutable=read.csv("student_habits_performance.csv")
```

Cargar librerias

```
#Librerias
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
library(tidymodels)
```

```
-- Attaching packages --------------------------------------- tidymodels 1.3.0 --
v broom        1.0.8     v rsample      1.3.0
v dials        1.4.0     v tune         1.3.0
v infer        1.0.8     v workflows    1.2.0
v modeldata    1.4.0     v workflowsets 1.1.0
v parsnip      1.3.1     v yardstick    1.3.2
v recipes      1.3.1
```

```
-- Conflicts --------------------------------------- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()      masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()   masks stats::step()
```

Visualización del dataset

Primeras 6 filas del dataset

```
head(rendimiento_estudiantes_inmutable,5)
```

```
  student_id age gender study_hours_per_day social_media_hours netflix_hours
1      S1000  23 Female                 0.0                1.2           1.1
2      S1001  20 Female                 6.9                2.8           2.3
3      S1002  21   Male                 1.4                3.1           1.3
4      S1003  23 Female                 1.0                3.9           1.0
5      S1004  19 Female                 5.0                4.4           0.5
  part_time_job attendance_percentage sleep_hours diet_quality
1            No                 85.0         8.0         Fair
2            No                 97.3         4.6         Good
3            No                 94.8         8.0         Poor
4            No                 71.0         9.2         Poor
5            No                 90.9         4.9         Fair
  exercise_frequency parental_education_level internet_quality
1                  6                  Master          Average
2                  6             High School          Average
3                  1             High School             Poor
4                  4                  Master             Good
5                  3                  Master             Good
  mental_health_rating extracurricular_participation exam_score
1                    8                           Yes       56.2
2                    8                            No      100.0
3                    1                            No       34.3
4                    1                           Yes       26.8
5                    1                            No       66.4
```

Variables del dataset

Variables numéricas continuas:

- age – Edad del estudiante
- study_hours_per_day – Horas de estudio por día
- social_media_hours – Horas en redes sociales por día
- netflix_hours – Horas viendo Netflix por día
- attendance_percentage – Porcentaje de asistencia a clases
- sleep_hours – Horas de sueño por día
- exercise_frequency – Frecuencia de ejercicio (veces por semana)
- mental_health_rating – Valoración del bienestar mental (escala 1 a 10)
- exam_score – Puntaje en el examen final (0 a 100)

Variables categóricas:

- student_id – ID único del estudiante (no se analiza, sirve para identificación)
- gender – Género (Male, Female, Other)
- part_time_job – Tiene trabajo de medio tiempo (Yes/No)
- diet_quality – Calidad de la dieta (Poor, Fair, Good)
- parental_education_level – Nivel educativo de los padres (por ejemplo: High School, Bachelor, etc.)
- internet_quality – Calidad del internet (Poor, Average, Good)
- extracurricular_participation – Participación en actividades extracurriculares (Yes/No)

Analisis exploratorio de datos

```
summary(rendimiento_estudiantes_inmutable)
```

```
 student_id             age           gender          study_hours_per_day
Length:1000        Min.   :17.00   Length:1000        Min.   :0.00
Class :character   1st Qu.:18.75   Class :character   1st Qu.:2.60
Mode  :character   Median :20.00   Mode  :character   Median :3.50
                   Mean   :20.50                      Mean   :3.55
                   3rd Qu.:23.00                      3rd Qu.:4.50
                   Max.   :24.00                      Max.   :8.30
social_media_hours netflix_hours   part_time_job      attendance_percentage
Min.   :0.000      Min.   :0.000   Length:1000        Min.   : 56.00
1st Qu.:1.700      1st Qu.:1.000   Class :character   1st Qu.: 78.00
Median :2.500      Median :1.800   Mode  :character   Median : 84.40
Mean   :2.506      Mean   :1.820                      Mean   : 84.13
3rd Qu.:3.300      3rd Qu.:2.525                      3rd Qu.: 91.03
Max.   :7.200      Max.   :5.400                      Max.   :100.00
 sleep_hours    diet_quality       exercise_frequency parental_education_level
Min.   : 3.20   Length:1000        Min.   :0.000      Length:1000
1st Qu.: 5.60   Class :character   1st Qu.:1.000      Class :character
Median : 6.50   Mode  :character   Median :3.000      Mode  :character
Mean   : 6.47                      Mean   :3.042
3rd Qu.: 7.30                      3rd Qu.:5.000
Max.   :10.00                      Max.   :6.000
internet_quality   mental_health_rating extracurricular_participation
Length:1000        Min.   : 1.000      Length:1000
Class :character   1st Qu.: 3.000      Class :character
Mode  :character   Median : 5.000      Mode  :character
                   Mean   : 5.438
                   3rd Qu.: 8.000
                   Max.   :10.000
  exam_score
Min.   : 18.40
1st Qu.: 58.48
Median : 70.50
Mean   : 69.60
3rd Qu.: 81.33
Max.   :100.00
```

```r
str(rendimiento_estudiantes_inmutable)
```

```
'data.frame':   1000 obs. of  16 variables:
 $ student_id                   : chr  "S1000" "S1001" "S1002" "S1003" ...
 $ age                          : int  23 20 21 23 19 24 21 21 23 18 ...
 $ gender                       : chr  "Female" "Female" "Male" "Female" ...
 $ study_hours_per_day          : num  0 6.9 1.4 1 5 7.2 5.6 4.3 4.4 4.8 ...
 $ social_media_hours           : num  1.2 2.8 3.1 3.9 4.4 1.3 1.5 1 2.2 3.1 ...
 $ netflix_hours                : num  1.1 2.3 1.3 1 0.5 0 1.4 2 1.7 1.3 ...
 $ part_time_job                : chr  "No" "No" "No" "No" ...
 $ attendance_percentage        : num  85 97.3 94.8 71 90.9 82.9 85.8 77.7 100 95.4 ...
 $ sleep_hours                  : num  8 4.6 8 9.2 4.9 7.4 6.5 4.6 7.1 7.5 ...
 $ diet_quality                 : chr  "Fair" "Good" "Poor" "Poor" ...
 $ exercise_frequency           : int  6 6 1 4 3 1 2 0 3 5 ...
 $ parental_education_level     : chr  "Master" "High School" "High School" "Master" ...
 $ internet_quality             : chr  "Average" "Average" "Poor" "Good" ...
 $ mental_health_rating         : int  8 8 1 1 1 4 4 8 1 10 ...
 $ extracurricular_participation: chr  "Yes" "No" "No" "Yes" ...
 $ exam_score                   : num  56.2 100 34.3 26.8 66.4 100 89.8 72.6 78.9 100 ...
```

```r
# 1. Preparación de Datos (funcional e inmutables)
#Funcion para preparar datos para el modelo

preparacion_data <- function(data){
    data %>%
    select(-student_id) %>%
    mutate(gender = factor(gender),
           part_time_job = factor(part_time_job),
           diet_quality = factor(diet_quality),
           parental_education_level = factor(parental_education_level),
           internet_quality = factor(internet_quality),
           extracurricular_participation = factor(extracurricular_participation),
           gender = as.numeric(gender),
           part_time_job = as.numeric(part_time_job),
           diet_quality = as.numeric(diet_quality),
           parental_education_level = as.numeric(parental_education_level),
           internet_quality = as.numeric(internet_quality),
           extracurricular_participation = as.numeric(extracurricular_participation),
           )
}
```

```r
# Funcion para calcular los coeficientes de correlacions de la variable
# dependiente con respecto a las variables independientes

coeficientes_correlacion <- function(data, metodo = "pearson"){
  if(!metodo %in% c("kendall", "pearson", "spearman") ){
    cat("Metodo no es el correcto, debe escoger entre 'kendall', 'spearman' o 'pearson' ")

  }else{
        i <- 1
        nombre_columnas <- names(data)
        columnas_qty <- ncol(data)
        Puntaje_examen <- data[,columnas_qty]
        columnas_p_value <- character(0)
        value_p <- c()
        correlacion <- c()
        while (i<columnas_qty){
          columna_sel <- data[,i]
          c1 <- cor.test(columna_sel,Puntaje_examen)
          corre_value <- cor(columna_sel,Puntaje_examen,method = metodo)
          p <-c1$p.value

            if(p<0.05){
              columnas_p_value <- append(columnas_p_value,nombre_columnas[i])
              value_p <- append(value_p, p)
             correlacion <- append(correlacion, corre_value)
              }
          i <- i+1
          }


      datanew <- data %>%
        select(columnas_p_value,nombre_columnas[[columnas_qty]])

      cor_y_p_value <- list(
          columnas_p_value_signi = columnas_p_value,
          valor_p = value_p,
          correlación = correlacion,
          metodo = metodo,
          col_qty = length(columnas_p_value),
          data_modelo = datanew
        )
        class(cor_y_p_value) <- "coeficientes"
```

```
        return(cor_y_p_value)
      }
  }
```

```
#Con esta función se da formato al imprimir el resultado de la clase coeficientes

print.coeficientes <- function(x, ...) {
  i <- 1
  cat("--- Resultado del calculo de las correlaciones con el Metodo:", x$metodo ," --- \n \n"
   while (i <= x$col_qty) {
     cat("La correlación entre",x$columnas_p_value_signi[[i]],"y exam_score es:", round((x$co
     i <- i+1
   }
  cat("\n--- Todas con un p-value < 0.05 ---")


}
```

```
# Se preparan los datos utilizando la funcion 'preparacion_data' y se guardan en 'rendimiento
rendimiento_estudiantes_preparado <- preparacion_data(rendimiento_estudiantes_inmutable)
```

```
# Se utilizan los datos preparados en la funcion 'coeficientes_correlacion' y se imprimen los
corpearson <- coeficientes_correlacion(rendimiento_estudiantes_preparado,metodo = "pearson")
```

```
Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
i Please use `all_of()` or `any_of()` instead.
  # Was:
  data %>% select(columnas_p_value)

  # Now:
  data %>% select(all_of(columnas_p_value))

See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
print(corpearson)
```

```
--- Resultado del calculo de las correlaciones con el Metodo: pearson  ---

La correlación entre study_hours_per_day y exam_score es: 82.54 %
La correlación entre social_media_hours y exam_score es: -16.67 %
```

```
La correlación entre netflix_hours y exam_score es: -17.18 %
La correlación entre attendance_percentage y exam_score es: 8.98 %
La correlación entre sleep_hours y exam_score es: 12.17 %
La correlación entre exercise_frequency y exam_score es: 16.01 %
La correlación entre mental_health_rating y exam_score es: 32.15 %

--- Todas con un p-value < 0.05 ---
```

```
data_modelo <- corpearson$data_modelo
head(data_modelo)
```

```
  study_hours_per_day social_media_hours netflix_hours attendance_percentage
1                 0.0                1.2           1.1                  85.0
2                 6.9                2.8           2.3                  97.3
3                 1.4                3.1           1.3                  94.8
4                 1.0                3.9           1.0                  71.0
5                 5.0                4.4           0.5                  90.9
6                 7.2                1.3           0.0                  82.9
  sleep_hours exercise_frequency mental_health_rating exam_score
1         8.0                  6                    8       56.2
2         4.6                  6                    8      100.0
3         8.0                  1                    1       34.3
4         9.2                  4                    1       26.8
5         4.9                  3                    1       66.4
6         7.4                  1                    4      100.0
```

```
str(data_modelo)
```

```
'data.frame':   1000 obs. of  8 variables:
 $ study_hours_per_day  : num  0 6.9 1.4 1 5 7.2 5.6 4.3 4.4 4.8 ...
 $ social_media_hours   : num  1.2 2.8 3.1 3.9 4.4 1.3 1.5 1 2.2 3.1 ...
 $ netflix_hours        : num  1.1 2.3 1.3 1 0.5 0 1.4 2 1.7 1.3 ...
 $ attendance_percentage: num  85 97.3 94.8 71 90.9 82.9 85.8 77.7 100 95.4 ...
 $ sleep_hours          : num  8 4.6 8 9.2 4.9 7.4 6.5 4.6 7.1 7.5 ...
 $ exercise_frequency   : int  6 6 1 4 3 1 2 0 3 5 ...
 $ mental_health_rating : int  8 8 1 1 1 4 4 8 1 10 ...
 $ exam_score           : num  56.2 100 34.3 26.8 66.4 100 89.8 72.6 78.9 100 ...
```

```
# 2. Especificacion del modelo (Funcional)
especificacion_modelo <- linear_reg() %>%
  set_engine("lm")
```

```r
# 3. Ajuste del Modelo
ajuste_modelo <- especificacion_modelo %>%
  fit(exam_score ~ study_hours_per_day + social_media_hours + netflix_hours  + attendance_per
```

```r
#Se visualiza el resumen del modelo generado
resumen_modelo <- ajuste_modelo %>%
  pluck("fit") %>%
  anova() %>%
  tidy()
print(resumen_modelo)
```

```
# A tibble: 8 x 6
  term                    df   sumsq   meansq statistic    p.value
  <chr>                <int>   <dbl>    <dbl>     <dbl>      <dbl>
1 study_hours_per_day      1 194133. 194133.     6832.   0
2 social_media_hours       1   9596.   9596.      338.   3.89e- 65
3 netflix_hours            1   5902.   5902.      208.   6.81e- 43
4 attendance_percentage    1   1617.   1617.       56.9  1.04e- 13
5 sleep_hours              1   6142.   6142.      216.   2.07e- 44
6 exercise_frequency       1   8609.   8609.      303.   2.03e- 59
7 mental_health_rating     1  30752.  30752.     1082.   4.53e-161
8 Residuals              992  28189.     28.4      NA    NA
```

```r
# se realizan test de la funcion creada con el metodo spearman
corspearman <- coeficientes_correlacion(rendimiento_estudiantes_preparado,metodo = "spearman"
print(corspearman)
```

```
--- Resultado del calculo de las correlaciones con el Metodo: spearman  ---

La correlación entre study_hours_per_day y exam_score es: 81.21 %
La correlación entre social_media_hours y exam_score es: -16.63 %
La correlación entre netflix_hours y exam_score es: -16.52 %
La correlación entre attendance_percentage y exam_score es: 9.39 %
La correlación entre sleep_hours y exam_score es: 12.34 %
La correlación entre exercise_frequency y exam_score es: 15.02 %
La correlación entre mental_health_rating y exam_score es: 32.34 %

--- Todas con un p-value < 0.05 ---
```

```r
# se realizan test de la funcion creada con el metodo kendall
corkendall <- coeficientes_correlacion(rendimiento_estudiantes_preparado,metodo = "kendall")
print(corkendall)
```

```
--- Resultado del calculo de las correlaciones con el Metodo: kendall  ---

La correlación entre study_hours_per_day y exam_score es: 62.55 %
La correlación entre social_media_hours y exam_score es: -11.26 %
La correlación entre netflix_hours y exam_score es: -11.31 %
La correlación entre attendance_percentage y exam_score es: 6.17 %
La correlación entre sleep_hours y exam_score es: 8.46 %
La correlación entre exercise_frequency y exam_score es: 10.73 %
La correlación entre mental_health_rating y exam_score es: 22.79 %

--- Todas con un p-value < 0.05 ---
```

```r
# se realizan test de la funcion creada con el metodo lala
cor1 <- coeficientes_correlacion(rendimiento_estudiantes_preparado,metodo = "lala")
```

```
Metodo no es el correcto, debe escoger entre 'kendall', 'spearman' o 'pearson'
```