***Informe “Aprendizaje de máquina”***

***“Modelo de predicción de tasa de éxito o deserción de alumnos en colegio.”***

**

* Alumnos: Maximiano Ovalle, Fabian Hevia
* Docente: Billy Peralta
* NRC: 8074

Fecha: 08/06/2024

**Introducción**

Durante los años 2020 a 2022, se realizó un estudio en una institución de la educación superior (adquirido a partir de varias bases de datos inconexas) relacionado con estudiantes matriculados en diferentes carreras de grado, como agronomía, diseño, educación, enfermería, periodismo, administración, servicio social, y tecnologías. Esto se recopiló en un set de datos que también incluye información conocida en el momento de la inscripción de estos estudiantes, de las que fueron otorgadas datos como su trayectoria académica, demografía y factores socioeconómicos, junto con el rendimiento académico de los estudiantes al final del primer y segundo semestre.

El objetivo con el cual esta investigación fue realizada, es con el fin de lograr predecir en base a los factores anteriormente mencionados, si un alumno inscrito en una institución de educación superior mantiene las condiciones para que este tenga éxito en su carrera, o este termine desertando. El problema como tal se formula como una tarea de clasificación de tres categorías, en la que existe un fuerte desequilibrio hacia una de las clases según los autores del dataset.

Es por eso que a lo largo de este documento se irán aplicando metodologías conocidas en el mundo de la inteligencia artificial, las cuales cuentan con herramientas y librerías que nos permiten implementar modelos de predicción para modelado. En esta ocasión haremos uso del dataset como tal, que estará siendo cargado mediante el archivo descargado de la página oficial del [dataset](https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success).

Contaremos también con el uso de tres modelos de clasificación para llevar a cabo las predicciones:

* Regresión Logística: Este modelo predice probabilidades usando función sigmoide, además de utilizar optimización para ajustar los coeficientes y es adecuado para problemas de clasificación binaria.
* Support Vector Machine (SVM): Encuentra un hiperplano óptimo para separar las clases, además de que es eficaz en espacios de alta dimensión, utiliza kernels para datos no linealmente separables.
* Random Forest: Modelo de ensamble basado en múltiples árboles de decisión, que implementa Bootstrap sampling y selección aleatoria de características. Esta modelo promedia resultados de varios árboles para mejorar la robustez y la precisión.

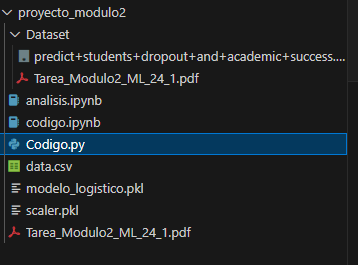
Una vez implementados realizaremos comparativas entre ellos para determinar cual entre estos modelos fue el mas preciso y el que representaría la mejor opción para este caso.

**Ambiente de trabajo para la implementación.**

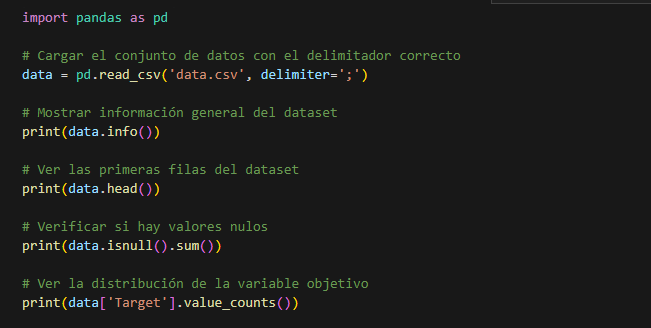
Como paso previo, utilizaremos para la implementación y generación de los modelos el ambiente de trabajo otorgado por Jupyter Notebook ya que, para la implementación de modelos de predicción, es ideal por la flexibilidad que presentan las casillas de código y el manejo de las herramientas para poder ajustar los parámetros de cada uno de los modelos de predicción. Además, para desplegar el notebook de Jupyter, se utilizará el entorno de Python con el compilador de Visual Studio Code.

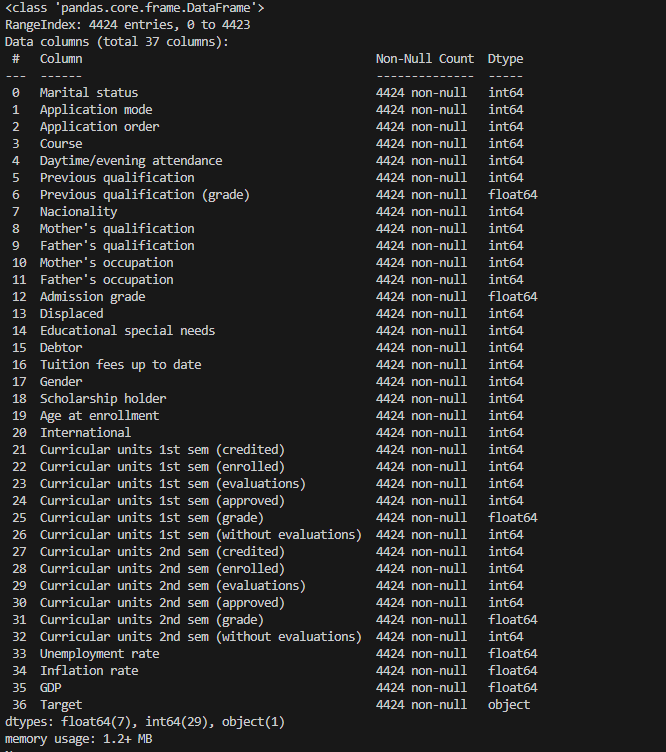
**Formateo e implementación de los datos**

En primera instancia debemos ser capaces de precargar los datos al ambiente de trabajo. Utilizaremos un ambiente de área local en el que los datos estarán disponibles para que el ambiente de Python sea capaz de relacionarlos al momento de precargarlos en el código.



Antes de realizar el formateo, haremos un análisis breve del estado natural de los datos, es decir cómo llegó descargada la información obtenida de internet.

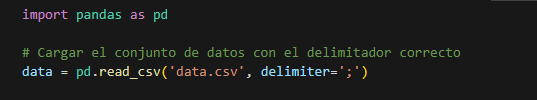
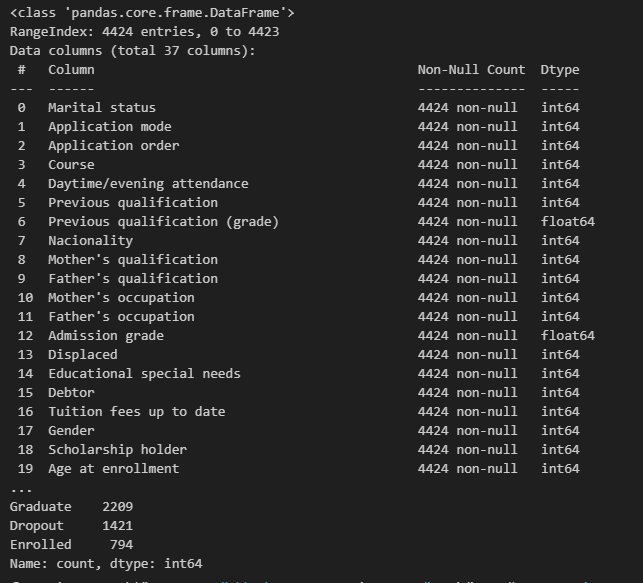
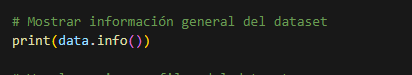




Al chequear podemos notar que tenemos 37 columnas dentro del dataset, de las que 36 son de tipo numérico y 1 de tipo object.

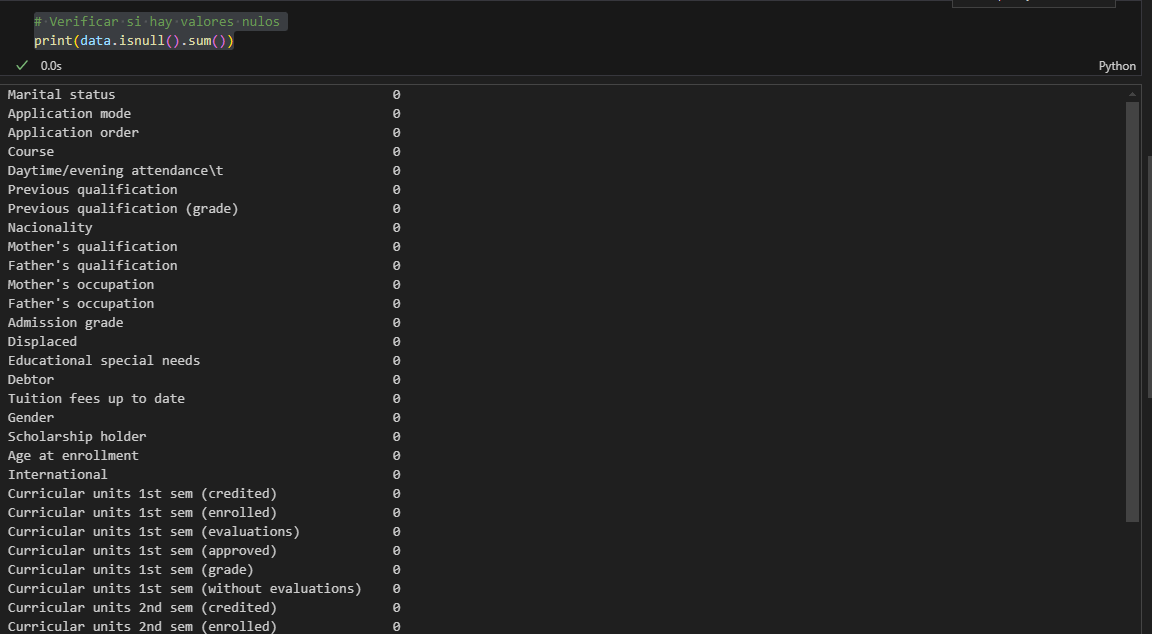
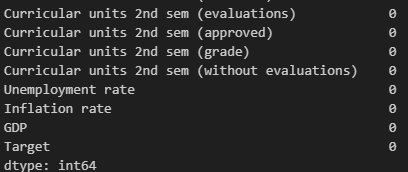
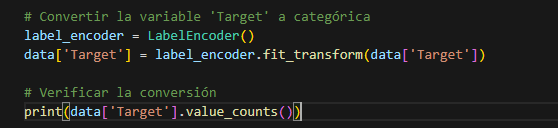
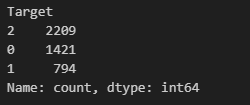
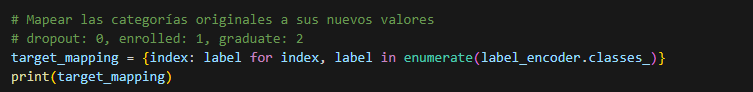
**Verificación Inicial**

Para iniciar nuestra verificación realizaremos la precarga del archivo entregado por los autores del dataset con el que contamos. Para ello lo haremos precargando los datos con un delimitador ya que es un archivo csv.

* Carga de datos
* Verificación de la estructura del dataframe

En cuanto a la verificación de datos, podemos percatarnos que todas las variables tienen relación con el tipo de dato que tienen asignado, siendo este int64 y float64 representando datos numéricos. Estos datos están correctamente asignados ya que todas las variables contienen una calificación numérica asociada a un número, es decir que cada numero en cada columna tiene una respuesta detrás de un simple número. 

**Figura 1**  
 - Tabla de clasificación a cada número.

* Existencia de datos nulos: Anteriormente en la tabla generada en la salida de nuestro algoritmo, pudimos ver un análisis reusmido de la categorización y cantidad de datos de cada variable. Si bien a simple vista no se perciben datos nulos, llevaremos a cabo un conteo de datos nulos en cada variable.
* 
* 
* Codificación de variables categóricas: En este caso nuestro dataset contempla una variable como tal categórica, siendo esta Target, ya que es la variable que determina si un alumno es o no es propenso a desertar de sus estudios. Para llevar a cabo la transformación se aplica lo siguiente:
* 
* 
* Con este código estamos transformando la categorización de los datos, finalmente se hace un mapeo a las categorías originales a sus nuevos valores.
* 
* 

Una vez realizado lo anteriormente señalado, nuestro set de datos estaría limpio y verificado para pasar a la etapa de división del conjunto de datos.

**Division del conjunto de datos**

Para dividir el conjunto de datos en conjuntos de entrenamiento y prueba, utilizaremos train\_test\_split de ‘sklearn’