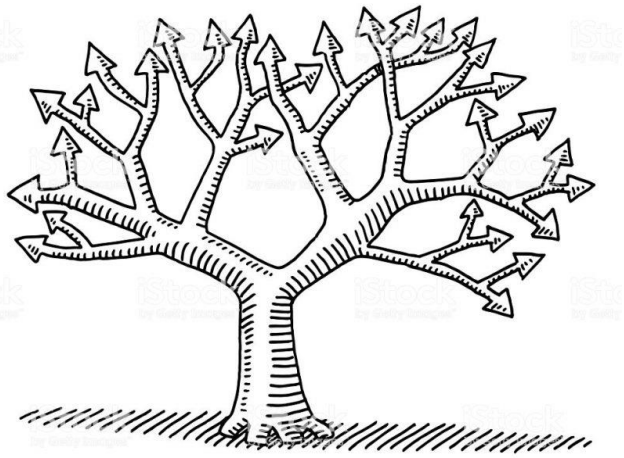
Segunda entrega – Proyecto final

Juan Manuel Muñoz (202010100010), Juan Miguel Castro (202010077010) y Juan Manuel Zapata (202010091010).

Algoritmo a implementar.

Todo el grupo se ha decantado por implementar el algoritmo CART de clasificación. No se tomó en cuenta el algoritmo CART de regresión ya que su objetivo es el de producir valores continuos o lineales.



Un ejemplo de esto es predecir el valor de una vivienda en base al número de baños, habitaciones, etc. En cambio, el algoritmo de clasificación tiene el objetivo de predecir valores de verdadero o falso, true o false, etc., conocidos también como valores discretos.

Como en nuestro caso, debemos predecir si un estudiante aprobará o no la prueba en base a sus puntajes previos y factores socio económicos, vemos que lo que buscamos predecir son valores discretos, falso o verdadero. Motivo que apoya nuestra decisión de implementar clasificación.

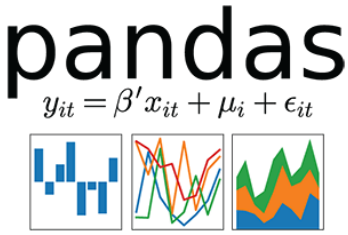
Otras de sus ventajas a la hora del análisis de datos son conocer las variables que más influyen en el árbol, su fácil compresión gracias a la representación gráfica de los árboles y su alta velocidad a la hora de ejecutarse en comparación de otros algoritmos como los bosques aleatorios.

Librerías a utilizar.

Una de las ventajas que ofrece Python frente a otros lenguajes de programación es su basta cantidad de librerías que ayudan a la lectura, análisis, escritura y gestión de grandes sets de datos, además de dar apoyo en la implementación de Machine Learning. A continuación, presentaremos las librerías que utilizaremos en nuestro proyecto.

Numpy.

Podemos definir esta librería como la versión mejorada de las listas que vienen por defecto en Python. Su mayor ventaja es su gran optimización que permite procesar más datos en menos tiempo que las listas por defectO. También implementa un sistema algebraico mucho mejor desarrollado que el de Python, el cual nos permite realizar un sinnúmero de nuevas acciones.



Pandas.

El principal motivo por el que decidimos implementar Pandas, es por su facilidad a la hora abrir, escribir y guardar archivos csv. Pandas es la líder por excelencia a la hora de tratar análisis de datos Ya que, podemos filtrar los datos, etiquetarlos y manipularlos de mil formas. Acá, las listas y matrices son conocidas por otro nombre, las listas como Series y las matrices como DataFrames.

Esta librería tiene una alta dependencia de Numpy y una alta compatibilidad con Scikit-learn, librería que veremos en instantes.

Matplotlib.

Como todos sabemos, el ser humano es un ser demasiado visual y esta librería nos ayuda con esto. Matplotlib nos permite generar de una forma muy fácil gráficas de todo tipo, circulares, lineales, de torres, etc., aparte de ser compatible tanto con Pandas como con Numpy.

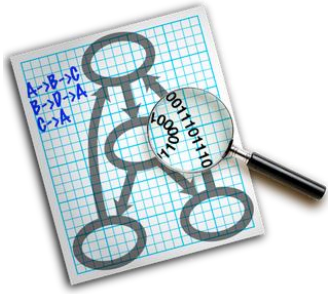
Scikit-learn.

La librería para implementar Machine Learning por excelencia. Esta enorme librería cuenta con una gran cantidad de algoritmos de Machine Learning, tanto supervisados como no supervisados, desde arboles de decisión de regresión y clasificación, bosques aleatorios por clasificación y regresión, regresión lineal, regresión lineal múltiple, regresión polinomial, hasta k-vecinos más cercanos.

Da la posibilidad entrenar el algoritmo de una forma muy fácil con los valores de entrenamiento que se creen, y mostrar su efectividad evaluando los valores de prueba que se entren. Permite igualmente crear matrices de confusión, mostrar que variables influenciaron más al algoritmo y exportar los modelos creados en base a los datos.

Graphviz.

Debido a que Matplotlib no cuenta con la opción de representar los árboles de decisión, se utilizará esta librería que permite exportar y representar diagramas en base al lenguaje descriptivo DOT.



Carga de datos.

La carga de datos ha sufrido cambios considerables respecto a la primera entrega. Ahora toda la carga está implementada con las librerías Pandas y Numpy. Se han creado tres métodos, los cuales leen distintos tipos de datos en específico. Hay un método que únicamente lee los puntajes y el éxito del estudiante, otro método que únicamente lee la información socio económica y el éxito del estudiante y un método que lee tanto los puntajes como la información socioeconómica, y obviamente el éxito del estudiante.

La carga se encuentra ya montada en Github, y se llama “Data_Processing.py”.

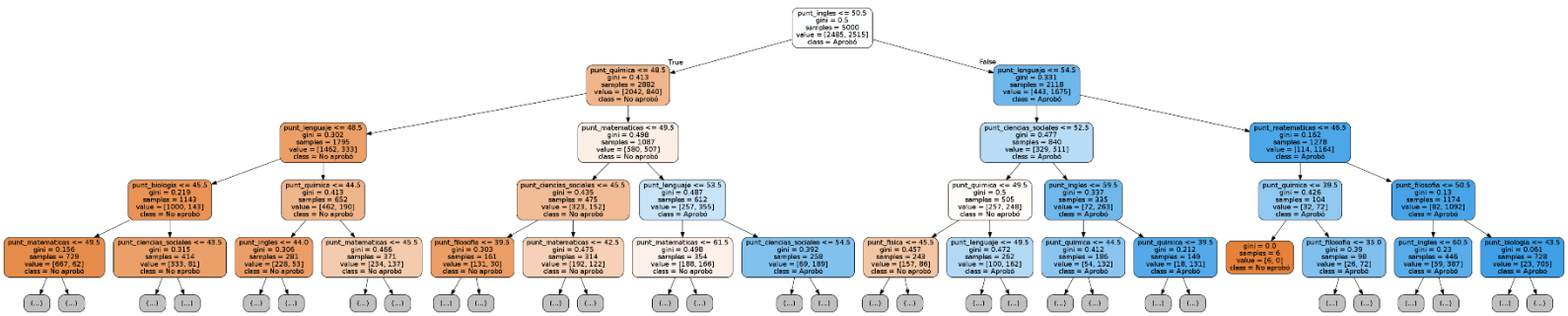
Métodos auxiliares.

Actualmente están escritos los códigos que permiten crear los árboles de decisión y crear los bosques aleatorios y, aparte de crearlos, entrenarlos con los datos de entrenamiento. Se encuentran en Github y se llaman “Decision_Tree.py” y “Random_Forest.py” respectivamente. En cada archivo hay tres métodos que cumplen funciones similares pero aplicadas a su respectivo algoritmo de Machine Learning, hay un método que crea y entrena árboles o bosques en base a los puntajes y éxito del estudiante, otro que crea y entrena árboles o bosques en base a la información socio económica y éxito del estudiante, y un método que crea y entrena árboles o bosques en base a los puntajes, información socio económica y éxito del estudiante.

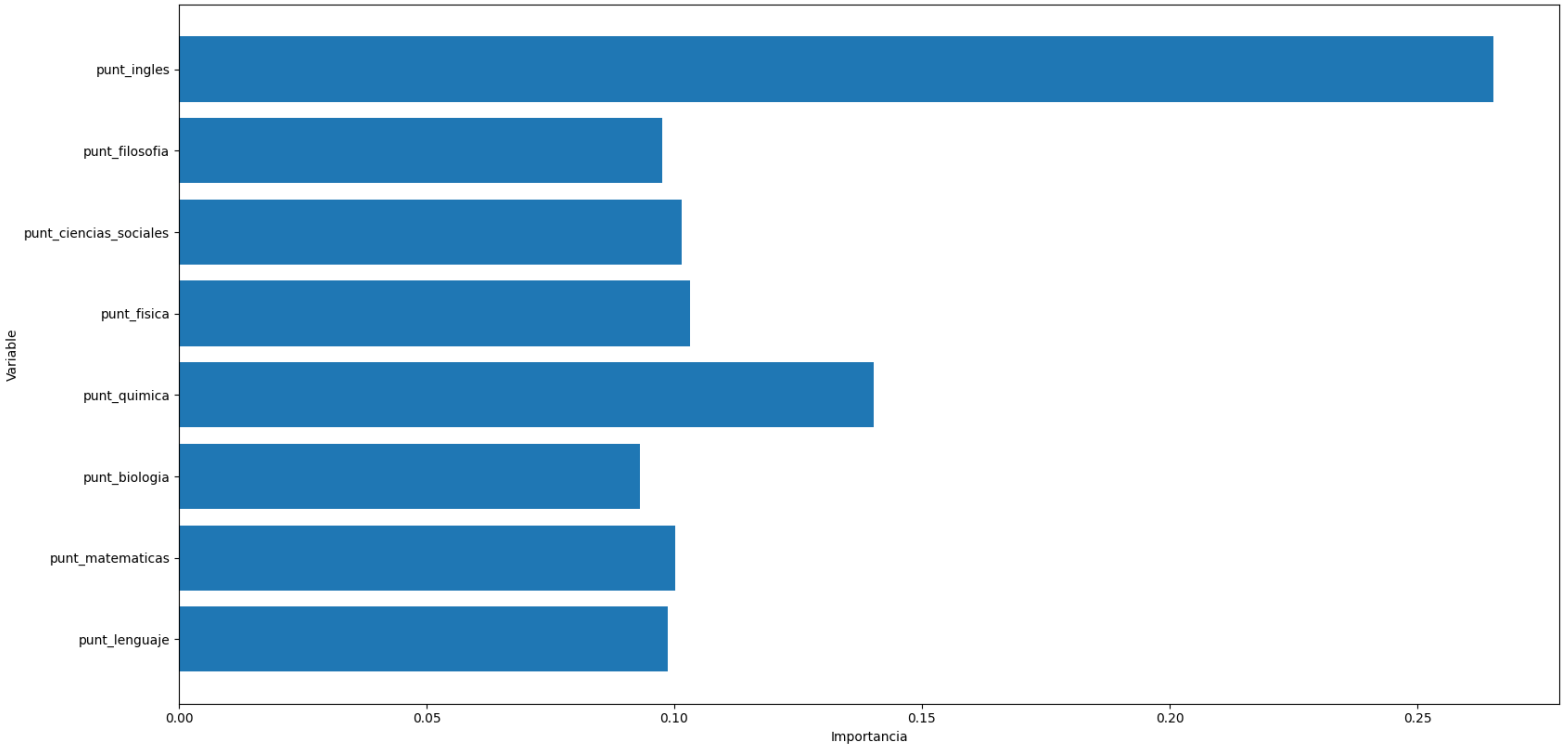
Hay también un archivo llamado “Prueba.py” en donde se están probando los árboles e imprimiendo su certeza.

Primer árbol de decisión generado.

Obviamente se han hecho pruebas respecto a los árboles de decisión y se han exportado gráficas. Este árbol ha sido generado únicamente en base a los puntajes del alumno, este modelo posee una precisión del 87% aproximadamente. Hay que tener discreción con este primer árbol ya que puede variar con el paso del desarrollo, es solo una primera vista.



Si no puede apreciar bien, también se dejará insertado en el repositorio de Github. Pero no solamente se han creado árboles, también se han graficado las variables que más influyen en el algoritmo. Recuerden, estos resultados pueden variar a lo largo del desarrollo, pero resulta conveniente insertarlo también. Para graficar se ha utilizado la librería Matplotlib.



Se puede ver que el factor más influyente es la prueba de inglés, siguiéndole la prueba de química y la prueba de física.

Enlace del proyecto a Github: Proyecto final Machine Learning