

# **Impacto de los Hábitos en Resultados Académicos**

**Autor: Natanael Marte Abreu.**

**Curso: Big Data y Business Analytics.**

**Centro: Cei - Centro de Estudios de Innovación.**

**Fecha: Julio 2025.**

# Índice.

<b>0. Retrospectiva.....</b>	<b>2</b>
• 0.1. Uso de IAs.....	2
0.2 Opinión personal.....	3
<b>1.Introducción.....</b>	<b>4</b>
Descripción general del proyecto.....	4
Elección del tema.....	4
Objetivos del proyecto.....	5
Alcance.....	5
<b>2.Metodología.....</b>	<b>5</b>
a) Datos utilizados.....	6
b) Extracción de datos.....	6
c) Transformaciones realizadas.....	6
d) Análisis de datos.....	7
<b>3.Análisis y resultados.....</b>	<b>8</b>
• Análisis e interpretación.....	8
• Machine learning.....	10
• Hallazgos y conclusiones.....	12
<b>4.Definición del sistema BI.....</b>	<b>13</b>
a) Capa de Negocio.....	13
b) Capa de BackOffice.....	14
<b>5.Solución BI / Cuadro de mando.....</b>	<b>15</b>
• Características y funcionalidades.....	15
• Construcción.....	16
• Uso en la toma de Decisiones.....	16
<b>6.Gestión de proyecto.....</b>	<b>17</b>
• Alcance.....	19
• Tiempo.....	19
• Coste.....	20
<b>7.Conclusiones y recomendaciones.....</b>	<b>20</b>
1. Conclusiones.....	20
2. Recomendaciones.....	21
<b>8.Referencias.....</b>	<b>22</b>
<b>9.Anexos.....</b>	<b>23</b>
• Anexo 0, Fuente original.....	23
• Anexo 1, Google Colab.....	24
• Anexo 1.2, Google Colab, Modelos ML.....	25
• Anexo 2, SQLite.....	28
• Anexo 3, Power BI.....	30

## 0. Retrospectiva.

El punto cero, es el último en ser escrito, aquí quería meter varias cosas que se salen de los puntos establecidos para la estructura del plan de proyecto y necesitaba tener una visión general antes de comentar nada.

### • 0.1. Uso de IAs

He utilizado IAs a lo largo del proyecto. Cuando era adolescente, siempre que tenía dudas, preguntaba a Google. Wikipedia era una de mis páginas más visitadas. Ahora, Chat GPT cubre esas necesidades. Digo ésto porque se supone debo poner los prompts que haya hecho, pero son demasiados. Igualmente explicaré el uso que le he dado mayormente.

1. *Cuestiones técnicas.* En cuanto a scripts, algunas de ellas las he buscado en Chat GPT o en su defecto corregidas con ayuda de Gemini ( IA integrada en Google Colab ). La función `.agg()` es un ejemplo de una función que no conocía, y la aprendí preguntando: ¿ se puede en python mostrar el min y máx de varias columnas a la vez? y me respondió con esa función. La cláusula "CASE" en SQL por ejemplo la había visto, pero no practicado, sabía que existía pero no recordaba cómo ejecutarla por lo que tuve que buscarlo. En general, cuando tenía dudas técnicas, buscada en Chat GPT.
2. *Diseño.* Para el diseño, en el proyecto, por ejemplo, busqué qué tipos de letras encajan más en un trabajo como el mío, cuáles son más agradables a la hora de leer, colores que no cansen mucho la vista al leer de forma continua. La IA me daba opciones, las probaba e iba eligiendo la que más me convencía.
3. *Nuevos acercamientos.* A la hora de hacer el proyecto, tuve que volver a las herramientas las cuales llevaba más de una década sin tocar, viendo que ahora además están las versiones de Google, me refiero en concreto a Google Docs, el cual tuve que ver vídeos y además preguntar cosas puntuales a la IA sobre conceptos, literal pensaba que "pestaña" a la izquierda de la interfaz se refería a páginas. Como es mi primera vez haciendo un proyecto de este estilo y llevaba tanto tiempo sin hacer algo parecido a este ( desde el instituto ), tenía literalmente conversaciones con la IA sobre temas de estructura, presentación, cuál formato es más cómodo para ambas partes usar, etc.

Como resumen, he usado mucho la IA, pero en aprender conceptos que tenía sin adquirir. Lo que yo saco en claro es que, supe qué preguntar, supe entenderlo y aprendí a aplicarlo.

- **0.2 Opinión personal.**

La experiencia general ha sido increíble, tanto las clases como éste proyecto, donde puedo ver que aunque estén en fases iniciales, he adquirido habilidades que hace 3 meses no tenía. Aún me falta mucho, pero algo ya me suena. La vorágine de pensamientos que tuve, la calmé tomando la decisión de actuar aunque hubiesen riesgos, sabiendo que estoy apenas empezando, aún así estoy mucho mejor que antes. Gracias por la paciencia por parte de los profesores.

Respecto al proyecto, obviamente si tengo que volver a hacer algo parecido, voy volando...espero. Si es cierto que las primeras veces de algo ( supongo nos pasa a todos ), suelo tomarme mi tiempo y hasta no asimilar del todo, me cuesta aplicar. Me pasarón 2 errores que quisiera mencionar. Primero, me equivoqué con la traducción de una columna y no me di cuenta hasta ya el final... en concreto la llamada "attendace\_percentage", la cual confiando por algún motivo ciegamente en mi inglés básico, traduje como "porcentaje de atención" y es realmente "porcentaje de asistencia" 😊. Pues todo el proyecto pensado que estaba tratando con el porcentaje de atención, inclusive estaba metida la variables en algunas queries, hasta había escrito un texto donde ponía en muy mal lugar el nivel de la captura de los datos del dataset, porque me parecía imposible que el consumo de redes sociales no impacte en nada en el nivel de atención de los estudiantes...claro estaba comparando el consumo de redes sociales con el nivel de asistencia. Ahora que lo pienso, sería una buena información a añadir en una base de datos, el nivel de atención.

El segundo error vino al no recordar ( o saber ) sobre que el random forest tiene una versión distinta a la utilizada en clase ( classifier ), que es la versión Regressor. Una clasifica categorías y la otra valores numéricos y mi Label era la nota ( formato numérico ). Estuve revisando una y otra vez, dónde había fallado, hasta que acudí a mi colega Gemini que básicamente me dijo: espabila crack, que estas intentando predecir una columna con valores continuos 😞

Pero de los errores se aprende, así que guay en verdad.

## 1.Introducción.

### Descripción general del proyecto.

En este proyecto analizaremos datos de patrones reales de estilos de vida, hábitos y circunstancias personales de estudiantes, todo ésto contenido en un dataset simulado. Con dicha información, exploraremos las correlaciones que hayan entre los comportamientos de los alumnos con mejores notas y los con peores notas e investigaremos si esas diferencias marcan realmente el desempeño académico.

¿Son los alumnos con mejores calificaciones unos genios, iluminados, personas milagrosas? ¿O esa genialidad la tenemos todos y hay que activarla de alguna manera? ¿Cómo se activa? Sumerjámonos con todas estas dudas ( y más ) en éste análisis. Veamos qué acciones, si es que hace falta, se podrían replicar en pos de elevar las calificaciones generales.

### Elección del tema.

Teniendo en cuenta que me puedo interesar por casi cualquier tema en cuanto se profundiza en el, tomé el siguiente enfoque para decidirme: considerar a los profesores como mis clientes. Así que comencé a buscar y entender cuáles son los problemas con más peso para los profesores de hoy en día en relación con sus alumnos. Por lo cual me centraré en buscar y ofrecer KPIs y resultados accionables que sirvan en la toma de decisiones de los docentes respecto a cómo mejorar algunos aspectos que estén teniendo mucho consumo de energía a cambio de unos muy ineficientes resultados.

En mi búsqueda de datos, veía muchas bases de datos relacionadas con el abandono estudiantil, lógico si tomamos en cuenta que el hecho de que un alumno abandone, se siente ( y es ) una representación de un cúmulo de tiempo, paciencia, esfuerzos y energías que han terminado tristemente en nada. Por lo cual entiendo que sea un problema tratado de forma más popular que los demás ( en los cuales me quiero enfocar ), pero a la par me parece un tema complejo y que muchas veces está más allá del alcance de acción de solamente los profesores, siendo así un problema donde realmente haría falta una coordinación tanto de profesores, familiares, los mismos alumnos e inclusive en algunos casos, haría falta coordinación social.

Así que teniendo en cuenta lo anterior, quise buscar el cómo mejorar aspectos donde los docentes tuviesen mayor potestad de acción a la par que englobar ésto en un entorno donde

se entienda que el alumno no abandonará pero que tiene resultados ineficientes y se quiera solucionar éste último. Por lo que finalmente opté por tratar el tema de las calificaciones, a través de los hábitos de los estudiantes. En última instancia, es un tema que personalmente me parece importante abordar y cuyo impacto final tiene una proyección y escalabilidad pasmosa.

### **Objetivos del proyecto.**

- Analizar los factores de aquellos estudiantes con mejores calificaciones.
- Buscar si hay alguna correlación clara entre los que cumplen ciertos hábitos y sus resultados.
- Intentar predecir cuales de estos hábitos tendrían más peso a la hora de perseguir calificaciones más altas.
- Aportar y proponer acciones para contribuir a la persecución de los resultados deseados.

### **Alcance.**

Con los datos disponibles, se hará un EDA inicial. Luego un análisis que abarcará los tipos descriptivo, diagnóstico y predictivo, mediante la implementación de varios modelos de machine learning.

No se trabajará en conexión de otras fuentes de datos además de la ya mencionada, lo que provocará limitaciones a la hora de la cantidad de datos utilizados, al igual que la ausencia de un diccionario de datos nos hará más susceptibles a la hora de entender mal alguna que otra columna ( siendo que las descripciones de columnas que hay en Kaggle sobre este dataset son bastante escuetas ).

## **2. Metodología.**

La metodología u orden el cual seguí, fue principalmente el siguiente:

1. Búsqueda y elección del dataset.
2. Análisis superficial y limpieza del dataset.
3. Consultas SQL.
4. Escalado e implementación de modelos de machine learning.
5. Cuadro de mando, dashboard.

Igualmente mencionar que la elección de las herramientas utilizadas ( Google Colab, SQLite, Power BI ), tuvieron un motivo principal y es: familiaridad. Estuve tonteando con otras como Visual Studio Code, Anaconda, PostgreSQL, Looker Studio pero teniendo en ellas muchas menos horas de práctica y experiencia opté por usar las ya conocidas las cuales tampoco es que las domine y sea un experto en ellas, pero un poco más familiares si me resultan.

#### **a) Datos utilizados.**

El dataset que estaremos utilizando fue obtenido de la plataforma Kaggle. Compuesto de mil de filas de alumnos ficticios, pero con patrones de comportamientos reales. Contiene columnas muy interesantes sobre los hábitos de los estudiantes, tales como: horas de estudio, horas de sueño, horas dedicadas a redes sociales. Además de otras que aportan matices del tipo: salud mental, nivel de educación de los padres, frecuencia de ejercicio semanal. Por supuesto teniendo datos más básicos pero igual de importantes como son: la edad, género y nota final.

#### **b) Extracción de datos.**

El dataset en cuestion es un archivo .csv el cual se descargó desde la plataforma Kaggle y cargamos desde Google Colab, para mediante Python, junto a la librería Pandas, poder hacer un primer análisis inicial de los datos. Posteriormente en SQLite se creó una base de datos con los mismos campos que el dataset que estamos trabajando, para finalmente cargar los datos del .csv a la base de datos como una tabla.

#### **c) Transformaciones realizadas.**

Las transformaciones se hicieron principalmente en:

- *Google Colab*. Aquí realizamos limpieza de nulos, comprobación de duplicados. Posteriormente, a la hora de alimentar a los modelos de machine learning, tuvimos que hacer escalado de variables ( método Min-Max Scaling ), transformación de variables categóricas ( texto ) en variables binarias ( booleanos ) mediante la técnica de One-Hot Encoding y eliminación de columnas.
- *Power BI ( Power Query )*. Con esta herramienta, principalmente hicimos cambios de formatos a las columnas, asignar si eran tipo texto, numérica ( entero o decimal ) que de primeras no los reconocía automáticamente. También se tuvo que cambiar la configuración regional de las columnas con decimales, ya que estos venían con el punto "." en vez de coma "," para los

decimales. Por último se generó una columna personalizada en base a la columna ya existente de 'study\_hours\_per\_day' para crear grupos de intervalos de horas y poder cubrir así una necesidad concreta para una de las visualizaciones.

#### d) **Análisis de datos.**

Haremos un análisis lo más completo posible, intentando cubrir de la mejor manera los siguientes tres tipos de análisis:

1. Análisis Descriptivo. Básicamente en los tres entornos que utilizamos ( Google Colab, SQLite y Power BI ) hacemos éste tipo análisis, en Google Colab ( a través de Python ) de forma más inicial, viendo las columnas del dataset, sus nombres, qué representan, valores que contienen, sus valores mínimos y máximos, sus valores únicos, etc.

Para cuando llegamos a SQLite hacemos ya consultas más concretas, haciendo uso de los filtros ( cláusula WHERE ), vemos cosas anteriormente no comprobadas como: cuántos alumnos hay según género, por edad, en relación al nivel de estudios de los padres, nota media en relación a las horas de estudio.

Finalmente en Power BI, aprovechando que accedemos a una exploración más visual, repasamos un poco las queries hechas con anterioridad, pero representadas ahora en gráficas de columnas agrupadas, de dispersión, entre otras.

2. Análisis Diagnóstico. Donde más interactivo fue este análisis fue en Power BI gracias a las segmentaciones de botones ( slicers ), que nos permiten ir viendo en tiempo real cómo va afectando el cambio de variables ( por ejemplo, si trabaja o no, educación de los padres ), en datos como la nota media y tiempo medio de horas de estudio. También en SQLite, como ya mencionamos, se hicieron consultas con filtros, tales como, nota media según salud mental, horas en redes sociales, etc.
3. Análisis Predictivo. Para cubrir éste tipo de análisis se utilizó en Google Colab la librería de Scikit-learn para entrenar los tres modelos de machine learning, los cuales fueron:
  - Linear Regression.
  - RandomForest Regressor.
  - Gradient Boosting Regressor.



Los cuales se les midió el rendimiento utilizando métricas como RMSE y  $R^2$ . Teniendo como Label en todo momento las notas de los exámenes.

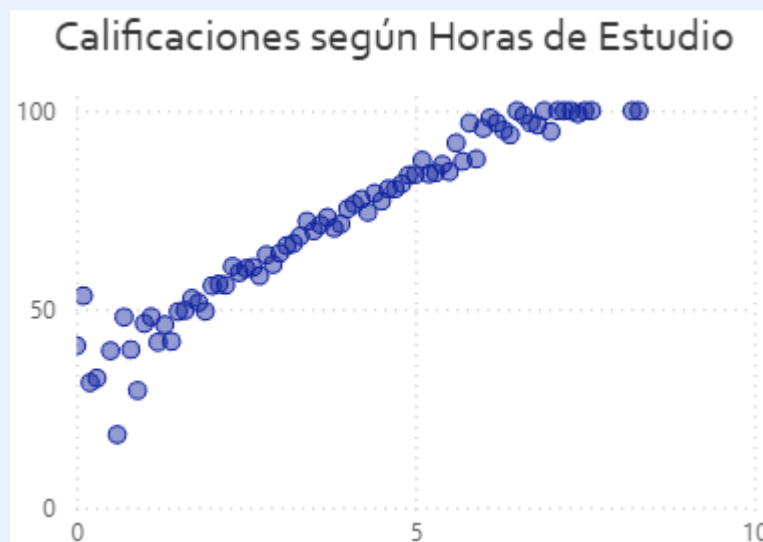
### 3. Análisis y resultados.

¿Qué hemos aprendido hasta ahora? Hemos asentado lo que se intuía de manera clásica, entonces, ¿para qué tanto análisis? Si le preguntamos a alguien que no tenga mucha idea de fútbol ( como es mi caso ), ¿cómo se ganan los partidos? Respondería seguramente: marcando goles, ¿cierto? PERO ¿cómo conseguimos ese gol?, ¿qué estrategia renta más?, ¿condición física de los jugadores?, etc etc....

He de admitir que cuando comencé este proyecto, esperaba resultados más escondidos. Si es cierto que ha sido interesante ver cuales de mis relaciones lógicas, hechas a priori, han sido corroboradas con los datos, aunque otras no lo han sido y éstas últimas me llamaron más la atención. Veámoslo.

- **Análisis e interpretación.**

Veamos los resultados gráficos que hemos obtenido e intentemos interpretar de la mejor forma posible dichos resultados.



Empezamos con un gráfico de dispersión ( scatter plot ), en este caso representa una correlación entre el desarrollo de las notas en relación a las horas estudiadas. En éste gráfico podemos apreciar fácilmente que cuantas más horas, más nota. Por lo que tanto tendencias como agrupaciones, son más cómodas de ver aquí. Por ejemplo, podemos dividir el cuerpo de la ilustración en tres

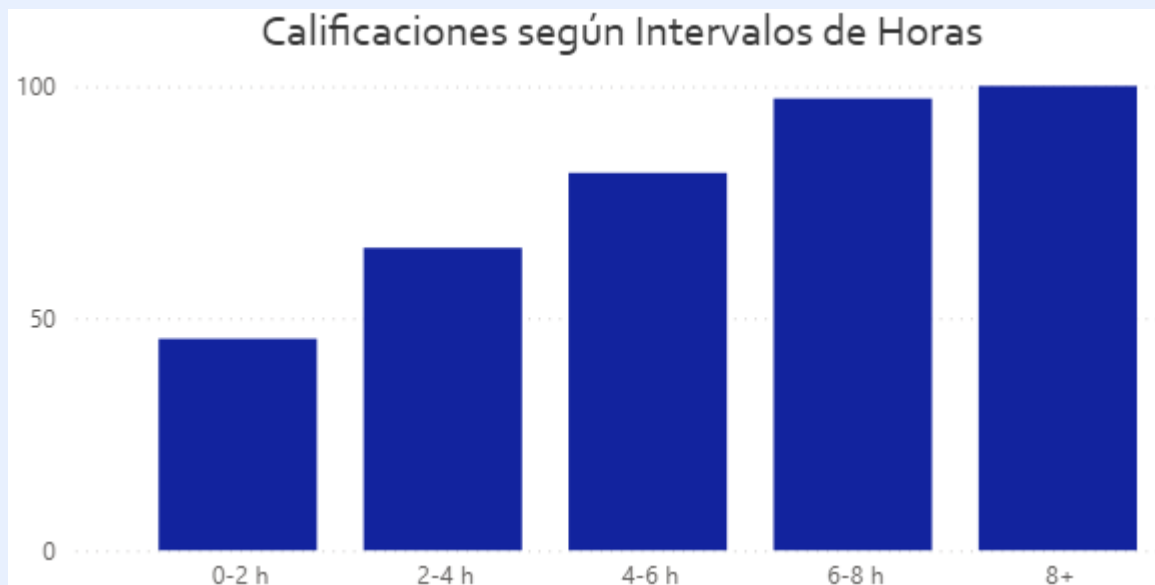
partes:

1. *La raíz.* Vemos sobre todo mucha separación en estos puntos iniciales, algunos con más tiempo de estudios sacan menos notas que otros, pero claro, hay que tener en cuenta el contexto global, de que la mayoría de esta "raíz", son suspensos, entendiendo que ahí nos movemos, primero, en un terreno que no nos debería preocupar en un principio, segundo, son resultados subóptimos los cuales

representan la forma dispersa que se obtiene al empezar a hacer algo, pero que aún le queda por desarrollar ( en éste caso, se estudia pocas horas, algo se consigue, pero no se efectúa el número de horas necesarias para conseguir el aprobado o notas mejores ).

2. *El cuerpo.* De ésta parte central del dibujo, llama la atención dos cosas, primero, la bastante homogénea agrupación y segundo, la tendencia de subida constante que se aprecia. Además, parece que ha evolucionado la dispersión inicial de inconsistencia, a unos resultados consistentes respecto a las horas estudiadas. Podríamos por lo tanto ver ya cuál sería el rango mínimo de horas para aspirar al aprobado, por ejemplo. Pero no es un objetivo, apuntamos a las mejores notas posibles, así que simplemente seguimos avanzando.
3. *La cabeza.* Arriba del todo, nos topamos con una agrupación, la cual se puede diferenciar del cuerpo y que a diferencia de la raíz, está sí que está con una agrupación más homogénea. Podemos aquí si fijarnos en algo que nos interesa, rango de horas entre las cuales se mueven las mejores notas.

Sigamos con el siguiente gráfico.



Aquí vemos un gráfico de columnas agrupadas, donde podemos visualizar mejor los valores, ahora divididos por grupos. Tenemos la misma correlación que gráfico anterior, pero ahora la variable de horas de estudios, la tenemos en forma de agrupaciones que van en intervalos de dos en dos horas, lo que facilita por ejemplo a responder las preguntas que se hicieron en el gráfico anterior, sobre cuál sería el rango de horas para aprobar o sacar las mejores notas. Los vemos fácilmente, de cero a dos horas, las probabilidades de aprobar son muy bajas; en un rango de seis a ocho horas, te aseguras estar estudiando lo suficiente para sacar notas sobresalientes.

	parental_education_level	nota_media
	Filtro	Filtro
1	Bachelor	70.27
2	High School	69.64
3	Master	68.09

A continuación veremos una de las primeras relaciones lógicas que pensé y cuyos resultados fueron distintos a lo esperado.

En éste resultado de una consulta en SQLite, tenemos la nota media de los alumnos, en relación al nivel de estudios de los padres ( traducidos a los homólogos españoles serían tal que: High School = La ESO + Bachillerato /

Bachelor = Universidad ). Y he aquí nuestra primera corrección, por parte de los datos a nuestra intuición ( mola ). Lo normal sería pensar que estudiantes con padres que hayan sido universitarios tendrían mejores resultados, que estudiantes con padres que hayan hecho bachillerato, ya que padres que han tenido más tiempo, por lo tanto, más experiencia de estudios y suponemos que transmiten las herramientas conseguidas por el camino a sus hijos, dando con ello a hijos con mejores notas, ¿no? La diferencia de nota media de un escalón a otro es de tan solo de 1 punto. Curioso.

	mental_health_rating	nota_media
	Filtro	Filtro
1	1	62.37
2	10	77.95

En la respuesta a otra consulta, obtuvimos ésto. Quería saber, comparando los dos extremos, cuánto afectaría el nivel de salud mental al promedio de la nota. Encontrándonos con una significativa diferencia de quince puntos. Era algo que se suponía, pero que en éste caso los datos corroboran.

Parece ser que solo tienen importancia las horas de estudio y que los demás hábitos o situaciones tienen poco o cero impacto en el resultado final de las calificaciones. Vamos por ello a hacer una visita a unos amigos que controlan bastante sobre problemas de regresión.

- **Machine learning.**

Hechas ya algunas queries, quería saber el peso ( en %) de todas las columnas, es decir, el Feature Importance. Para ello entrenamos tres modelos: RandomForest ( rf ), Linear Regression ( lr ) y Gradient Boosting Regressor ( gbr ).

Los tres modelos fueron entrenados con la columna de las notas ( 'exam\_score' ) como el Label, siendo el resto de columnas los Features. Los resultados fueron bastante similares, por lo que, para evitar repetirnos, vamos a ver únicamente los resultados de dos de los tres modelos, teniendo éstos dos cosas a comentar y considerar.

	index	0
1	71.935195	study_hours_per_day
7	10.871284	mental_health_rating
2	3.721252	social_media_hours
6	3.260780	exercise_frequency
3	3.244784	netflix_hours
5	2.897234	sleep_hours
4	1.598136	attendance_percentage
0	0.785765	age
11	0.282178	diet_quality_Good
15	0.233005	internet_quality_Good
8	0.209089	gender_Male
12	0.195962	diet_quality_Poor
10	0.185702	part_time_job_Yes
13	0.185499	parental_education_level_High School
17	0.163355	extracurricular_participation_Yes
16	0.125008	internet_quality_Poor
14	0.097697	parental_education_level_Master
9	0.008074	gender_Other

Empezamos con los resultados del rf.

He de admitir que la primera vez que vi esta tabla, el resultado me impactó. Un aplastante 71.93% por parte de las horas de estudio, le saca básicamente 61 puntos a la segunda posición....

Ahora bien, además del indiscutible ganador y la segunda posición ( salud mental con un 10% ), las demás categorías tienen muy poco peso. También podemos volver a observar , en un formato diferente, los resultados a la consulta de la correlación entre el nivel de estudios de los padres y la nota de los hijos.

En el Linear Regression avanzaremos un poco más en la comprensión de éstos datos.

	feature	coefficient
1	study_hours_per_day	0.956068
2	social_media_hours	-0.222934
7	mental_health_rating	0.213358
5	sleep_hours	0.170191
3	netflix_hours	-0.147386
6	exercise_frequency	0.102474
4	attendance_percentage	0.071847
10	part_time_job_Yes	0.009107
15	internet_quality_Good	-0.007615
14	parental_education_level_Master	-0.006017
0	age	0.005352
16	internet_quality_Poor	-0.005149
12	diet_quality_Poor	-0.004887
9	gender_Other	0.004804
11	diet_quality_Good	-0.004183
17	extracurricular_participation_Yes	0.003555
8	gender_Male	0.003006
13	parental_education_level_High School	0.000734

De primeras, los resultados del lr no los tenemos en porcentajes, pero podemos seguir entendiéndolo con el mismo significado de “peso” que en el rf. En este caso, hace referencia a que por cada unidad de la categoría X, se suma la cantidad del coeficiente indicado o se resta... y he ahí el punto que me llamó la atención de este modelo, lo cual con el rf no pensé, y es el

peso negativo.

Porque entendemos que una categoría pueda aportar poco a conseguir mejores resultados, pero ¿cuánto influye en conseguir peores resultados?. Por ejemplo, podemos ver en comparación al rf que aquí tenemos en segunda posición, con un coeficiente negativo, el de las horas de redes sociales. Por lo tanto cuantas más horas en redes sociales peores notas, teniendo todo el sentido ya que cuantas más horas de rrss, menos horas de estudio, nuestro campeón invicto.

- **Hallazgos y conclusiones.**

Antes que nada, decir que cuando empecé este proyecto, un dataset de mil filas me parecía suficiente... Ahora ya experimentado, no puedo parar de pensar que me faltan datos, filas( además de otras categorías ), mil se me ha quedado muy corto y sesga demasiado, para mi gusto, los resultados. También mencionar que al no disponer de un diccionario de datos, hay mucha incertidumbre en cuanto a ciertos datos, por ejemplo, la salud mental, la escala que se usa se interpreta que 1 es malo, pero ¿qué tan malo, qué implica? no se sabe.

Dicho esto, se viene el café para los muy cafeteros. Vamos a omitir el orden del título e iremos mencionando hallazgos y conclusiones de forma conjunta.

Las horas de estudio es lo que da mejores resultados. Actividades que quiten tiempo a las horas de estudio pesan negativamente debido a la pérdida en costo de oportunidad de estudiar en ese tiempo. Vemos que categorías como “tener un trabajo a tiempo parcial” no tienen un valor negativo, que dormir más de ocho horas tampoco tiene un peso negativo, pero sí redes sociales y ver netflix. Entendemos que es por la distribución del día, es decir,

las redes sociales y netflix, compiten en el mismo intervalo de horas que entraría el estudiar; dormir, trabajar o hacer deporte en un principio no. También se sobreentiende que las actividades que tienen un coeficiente positivo, pasadas de cierto valor ( en donde ya choquen en conflicto con las horas de estudio ), pasarían a tener un coeficiente negativo.

Además de otra apreciación, la sinergia. Actividades como dormir, comer bien, hacer deporte, entre otras, aportan valor y mejoran el rendimiento de las horas de estudio. Son categorías que por sí solas tienen poco peso, pero que fortalecen la categoría principal.

Por ejemplo, tener un buen descanso no te hará sacar mejores notas per se, pero estudiar habiendo descansado adecuadamente te ayudará a sacarle más provecho a esas horas de estudio.

Lo que me lleva a lo siguiente: Potenciar las horas de estudio. Comer bien, hacer ejercicio, dormir las horas correspondientes, todas esas y otras actividades preparan y ponen a punto la herramienta que usamos para estudiar: el cerebro. El próximo paso a dar, sería, la forma de estudiar, es decir, las técnicas de estudio. La forma en la que se emplean esas horas de estudio, importan. Enfocarte mucho tiempo en técnicas ineficientes, te dará resultados, pero como podemos suponer, si ese mismo tiempo lo aplicas en técnicas eficientes, los resultados serán mucho mejores y consumiendo los mismos recursos.

Técnicas de estudio probadas y testeadas en cuanto a eficacia tenemos varias, entre las cuales tenemos opciones interesantes tales como el "Active Recall" y el "Spaced Repetition". Siendo las técnicas de estudio una de las categorías que me encantaría tener dentro de los datos a analizar. Porque vale, los que estudian de seis a ocho horas tienen mejores resultados, pero, ¿cómo emplean esas ocho horas?, ¿toman descansos? ¿de cuánto tiempo? y así muchas más preguntas surgen.

Llegados a este punto, empezaron a fluir diferentes corrientes, lo aprendido en el proyecto, lo aprendido con anterioridad de forma colateral sobre este tema empezó a emanar desde la memoria y finalmente lo aprendido por la experiencia propia. Entre todo eso me vino una cita de Richard Feynman, premio nobel de física del año 1965 al cual le preguntaron: ¿Si una persona normal estudiara mucho, podría imaginar las cosas que usted imagina? A lo cual él respondió: "...por supuesto, yo era una persona normal que estudiaba mucho..."

📺 Feynman-"I was an ordinary person who studied hard" ( Fragmento de 46 segundos de la cita ).

En la respuesta completa se da en un término que considero clave: 'Interés'. Pensemos en esa persona que no destaca de manera académica pero que luego cuando te habla se su serie o deporte favorito...te sabe decir todos los nombres de personajes, sus tramas, trayectorias, fechas, premios, etc...con una precisión muchas veces pasmosa.

Habrían más conclusiones y preguntas, pero sin llegar a aplicar lo ya mencionado, sería divagar sin más. Por ahora, nos quedamos con éstas conclusiones.

## 4. Definición del sistema BI.

### a) Capa de Negocio.

- Objetivos: El principal objetivo del sistema BI es analizar los hábitos y situaciones personales de los estudiantes buscando correlaciones entre esos factores y los resultados académicos obtenidos. Respondiendo por el camino, a cuántas horas de estudio corresponden a las mejores calificaciones, a si el tiempo de sueño, la calidad de la dieta o el uso durante horas de redes sociales tienen algún impacto en la nota, etc.
- Información: Los datos utilizados contienen información de los estudiantes divididas en categorías tales como: nota final, horas de estudio, salud mental, nivel de asistencia, entre otras. Con dichas categorías se pudieron crear filtros para explorar de forma más precisa. Entrenar modelos de machine learning, para poder despejar las dudas sobre cómo conseguir mejores calificaciones a través de las predicciones obtenidas.
- Acciones: Las acciones a seguir, teniendo en cuenta las conclusiones que hemos obtenido, serían las siguientes:
  - 1) Hacer cuestionarios a los alumnos para saber si utilizan o no alguna técnica de estudio, en caso de que si, ¿cuál?
  - 2) Proporcionar información introductoria sobre una o dos técnicas de estudio a los alumnos que no tengan ningún sistema a la hora de estudiar.
  - 3) Analizar los datos obtenidos del paso uno y dos.

### b) Capa de BackOffice.

- Identificación de los datos: Los datos proceden de un dataset simulado, de la plataforma Kaggle, creado con patrones reales de estudiantes, que explora cómo los estilos de vida de los estudiantes afecta a sus resultados académicos.
- ETL:
  - 1) *Extracción* -> Dataset descargado desde Kaggle, archivo en forma .csv.
  - 2) *Transformación* -> Con Python se limpio el dataset, eliminando nulos, duplicados, etc. En Power BI se realizaron cambios de formatos de algunas columnas ( asignar si era tipo texto, decimal, entero ), además de la agregación de una columna personalizada para cubrir una necesidad concreta en relación a una visualización.

- 3) *Carga* -> El archivo .csv ya limpio, tuvo dos entornos de carga, en una base de datos de SQLite que hicimos con los mismos campos que tiene el dataset para que coincidieran los datos al cargarlos y así poder hacer las respectivas consultas. Finalmente los cargamos también en Power BI como su destino final para su visualización y análisis de forma más interactiva.
- Almacenamiento: Tuve dos entornos de almacenamiento:
    - 1) *Local*. En mi ordenador se guardó tanto la versión raw como la limpia del archivo .csv, además de la base de datos en formato .db, las queries guardadas en .sql, el dashboard de Power BI en .pbix.
    - 2) *Google Drive ( nube )*. Todos los archivos .ipynb relacionados con los modelos de machine learning están guardados aquí, además de copias de los .csv.
  - Procesamiento: Se dió mayormente en Power BI, haciendo promedios, visualizaciones dinámicas, segmentaciones con slicers. Además de los entrenamientos de los modelos de machine learning.

## 5.Solución BI / Cuadro de mando.

Vale, por fin la culminación de tanto jaleo previo, el dashboard en Power BI. Mi proyecto es de la modalidad Data Analyst, por lo que el dashboard se entiende debe ser uno de mis puntos fuertes. Aplicando criterios aprendidos durante el módulo, he preferido poner poca cosa, pero que lo que esté puesto tenga peso y utilidad. También noté sobre todo aquí en Power BI, con los slicers, la molestia de tener pocos datos; cuando iba a poner un valor en un slicer y veía que ese valor sólo representaba a dos o tres personas ... me ponía triste, la verdad.

- **Características y funcionalidades.**

El dashboard está constituido de los siguientes elementos:

- 1) *Un Gráfico de dispersión ( scatter plot )*. Con el que podemos visualizar con claridad la correlación entre la variable nota media y horas estudiadas, viendo como ésta primera sube positivamente en relación a la segunda. También se identifican fácilmente las agrupaciones ( clusters ) y tendencias en ésta gráfica.
- 2) *Un Gráfico de columnas agrupadas*. Aquí vemos las mismas variables que en el gráfico de dispersión, pero ahora la variable de horas de estudio, están en distintos



grupos de intervalos de horas. Así vemos por ejemplo, dentro de qué intervalo de horas están metidos la mayoría de suspensos, si hay algún salto de escalón drástico.

- 3) *Dos Tarjetas*. Muestran los datos más relevantes, una tiene la nota media y la otra el promedio de horas estudiadas, de tal manera que según vas interactuando en el dashboard puedes ir apreciando los cambios en estas dos tarjetas.
- 4) *Cuatro Slicers*. Puse cuatro segmentaciones de botones, tres de ellos con tres filtros diferentes y uno con dos filtros. Aquí tuve que hacer un poco de criba, ya que varias categorías que a priori parecían interesantes, al probarlas no daban ningún cambio relevante. Por lo que opté por poner categorías que al filtrar a través de ella, se vea un cambio notorio en nuestras dos tarjetas ( y por lo tanto también en las gráficas ). Los valores concretos ( botones ) dentro de cada slicer, fueron elegidos teniendo en cuenta la representación de ese valor ( es decir, los valores con más representación tenían prioridad ), e intentando dar un orden de menos a más.
- 5) *Cuadro de texto*. Un pequeño texto, que deja claro en cuales métricas debemos fijarnos más, pero que también éstas métricas pueden ser transformadas por otros parámetros, animando un poco a probar esos filtros ( Slicers ).

- **Construcción.**

El dashboard se hizo en la aplicación Power BI Desktop usando el dataset que previamente limpiamos con Python. Al cargarlo en Power BI, tuvimos que hacer algunas transformaciones, primero cambiar la configuración regional a las columnas que contuviesen valores decimales, para que la aplicación tradujera y leyera, los puntos decimales por comas decimales. Por último creamos una columna personalizada a partir de la columna ya existente de 'study\_hours\_per\_day', para tener una columna con esos valores pero agrupados en intervalos de dos horas.

En cuanto al diseño, busqué paletas de colores las cuales fueran cómodas de ver, que no fuera chillón, por ello me decanté por azul oscuro de forma predominante en las visualizaciones ( en sustitución del azul celeste que viene de forma predeterminada ), cambié el fondo blanco puro, por un tono más apagado. A los elementos que hay en el dashboard, les puse tanto bordes como sombreado para que resaltaran en contraste al fondo. A la par que también redondeé todas las esquinas de las ilustraciones, a mi parecer da una imagen más limpia y agradable. A los gráficos, les modifiqué la información que traen de forma predeterminada ( texto de la variable que representa cada eje ), les añadí un título a cada uno. En concreto, al gráfico de dispersión, le hice un cambio que a mi entender hace que tenga mejor aspecto y no sesga en nada los resultados; básicamente en el eje Y le aumenté el máximo de 100 a 105 y en el eje X de 8 a 10, al tenerlo de ésta forma, los puntos que salen sobre todo en la cabeza del dibujo, se aprecian mejor y sin ser cortados ( porque ahora hay más espacio más allá del límite ). Además a éste mismo gráfico, le cambié la

opacidad de los puntos, para con ello ver más claramente cada uno de forma individual aunque esté dentro o rodeado por otros.

- **Uso en la toma de Decisiones.**

El cuadro de mando lo quise hacer de tal forma que los clientes finales interactuaran con el, viendo qué pasa si filtro aquí, o si filtro allá.

Aún así, como algo que ya comenté al inicio, hay factores que están fuera del alcance de los profesores. Por lo que en última instancia, resalto los KPIs más relevantes ( las tarjetas ), y accesibles para la intervención de los profesores. Por ello esos valores son a los que hay que intentar dedicarles más tiempo y trabajo, hacer hincapié en cómo esas horas de estudio pueden ser más eficaces a través del método o técnica que se use. Centrando los esfuerzos en desarrollar estrategias de mejores rutas de aprendizaje.

## **6.Gestión de proyecto.**

La gestión, siendo sincero, al ser algo individual, la hice bastante flexible e iterativo, por lo que dentro de las metodologías ya existentes, con la cual mayor similitud tuve, fue la Agile.

La planificación previa fue bastante escasa, por no decir nula (aparte de ya tener pensando las herramientas que iba a utilizar), por lo que el plan de acción era básicamente ir avanzando y decidiendo sobre la marcha. Teníamos las fechas, teníamos los requisitos, sólo faltaba ponerse e ir viendo.

Dicho ésto, si que podemos distinguir tres etapas:

1. Etapas de elección ( o inicial ). Antes de empezar a enfocarme totalmente en el proyecto, ya iba pensando y anotando ideas con las cuales podría trabajar. La que más me convenció y la cual iba a ser la elección final, era analizar la capacidad de Madrid de construir y aumentar la oferta de viviendas de manera vertical, es decir, capacidad de construir bloques de viviendas más altos, allí donde hubieran edificios de pocas plantas. El tema del acceso a la vivienda me parecía de mucha relevancia, tiene impacto a niveles más profundos de los que a primera vista parece. El no acceder a una vivienda propia en ciertos momentos de la vida de toda persona, dificulta o retrasa elecciones vitales importantes tales como si casarse o tener hijos, además de que la estadía en casa de los padres o algún familiar, también afecta a la

falta de experiencia en cuanto a independencia y con ello a desarrollo de madurez de toda persona al emanciparse.

Tenía ganas, tenía referencias, información de personas que saben mucho más que yo, y veía una aportación valiosa en dar gráficos y datos de forma lo más sintetizados posibles, para un consumo ligero. Tenía puntos importantes, pero después de escuchar unas palabras de tutoría sobre mejor hacer un proyecto simple pero terminado a uno complejo pero con muchos cabos sueltos. Decidí aparcas dicho tema, ya que soy totalmente consciente de la complejidad a la cual me iba a enfrentar, y de mi escasa experiencia y habilidad la cual aún está en desarrollo.

En éste punto volví a estar en la línea de partida. Estuve pensando otros temas, a la par que también iba ojeando datasets de distintas páginas ( [datos.gob.es](https://datos.gob.es), [kaggle.com](https://kaggle.com), entre otras ), pasé por datos de accidentes en Madrid, datos sobre productos de Mercadona, pero me parecía que estaba buscando, casi inventando, problemas que resolver en vez de ir a problemas reales y tratarlos. Hasta que finalmente, teniendo ésto último en mente, pensé, para tener una experiencia más real en cuanto a lo que quiero hacer con todo éste conocimiento adquirido, voy a tomar a los evaluadores como mis clientes, de tal forma que fue mucho más fácil elegir tema, ya que no tenía que elegir temas al azar que me parecieran importantes, tenía que elegir lo problemas más relevantes a los que mis clientes se enfrentaban y ver cómo yo podía solucionar o lidiar con ellos. Finalmente terminé eligiendo el tema de cómo los hábitos de los estudiantes afectan a sus calificaciones.

2. Etapas técnicas ( o intermedia ). A partir de la elección del dataset, lo demás fue bastante fluido, los aspectos de analizar la composición del dataset, la limpieza de nulos, al ser algo que se había practicado repetidas veces durante el módulo, resultó en algo rápido de hacer y entender.

Una vez limpio el dataset, de seguido aproveché para dejarlo escalado así cuando vaya a hacer el paso de machine learning, tenerlo ya listo. Primero quería hacer algunas consultas que en python, la verdad no sabía cómo hacerlas e igualmente me parecía más cómodo hacerlas desde SQLite, así que fue el siguiente paso.

Hechas las consultas iniciales en SQLite, pasé directamente a los modelos de machine learning, quería saber ya los pesos de cada atributo. Teniendo los resultados, surgieron algunas consultas extras debido a esta nueva información.

Por último, el dashboard, sinceramente, quise llegar aquí teniendo claro lo que quería mostrar, es decir, algo de análisis hice, bastante cómodo y visual ciertamente, pero ya tenía claro las variables y KPIs que quería representar.

3. Etapas plan de proyecto ( o final ). Aquí ya solo me quedaba recopilar, y explicar todo aquello que había hecho, parte que me resultó bastante tediosa y que he de admitir, sentía no lo había tenido muy en cuenta mientras desarrollaba todo, es decir, para hacer la parte de documentación, tuve que ir volviendo muchas veces a cosas ya hechas. Ya a posteriori, veo que hubiese sido mucho más práctico el ir documentando a la par que se iba haciendo ( obviamente de forma suelta y ya más adelante unir esas piezas sueltas de “puntos de guardado” )

Vamos a traducir ahora toda ésta “flexibilidad” ( por no llamarlo caos ), en las siguientes tres áreas de conocimiento:

- **Alcance.**

El alcance se expandió desde la obtención, ETL, EDA, de los datos hasta la culminación de un cuadro de mando en Power BI, pasando por un componente predictivo mediante modelos de machine learning. El objetivo era evaluar cómo distintas variables personales y sociales influyen en el rendimiento académico y ver si era posible su replicación para aplicar acciones que aumentaran las notas académicas en general.

- **Tiempo.**

No hubo una agenda temporal como tal... utilizando fechas, podría quedar de la siguiente manera. Por motivos de disponibilidad, la etapa uno ( inicial ) descrita con anterioridad, fue en días alternos y con temporalidad de horas diversas, pero para ubicarnos, a finales de la semana del 7 al 13 de julio terminé ésta etapa. Por lo que la dos siguientes etapas ( intermedia y final ) me llevaron desde el 14 al 26 de julio, todos los días, pero igual que antes, con diversidad de horarios.

Juntando un poco las dos últimas etapas, he de decir que las partes técnicas me fueron más rápidas que las partes lógicas, porque si algo no me acordaba o tenía duda, tenía los apuntes o en su defecto buscaba respuesta en Chat GPT. Por otro lado, la parte lógica, me tomaba mucho más tiempo de reflexión, no miento si digo que tardé más en hacer algunas queries que en hacer el escalado para los modelos de machine learning. Me levantaba del ordenador, daba vueltas por la casa o me tiraba en la cama a pensar sobre las posibles relaciones o que parte importantes tendría que abordar ( y ésto podía ser cinco minutos o media hora ).

Luego la parte que más tiempo me llevó sin dudas, fue documentarlo, por diversos motivos. Primero la falta de experiencia haciendo éste tipo de proyectos, no saber de primeras cómo hacerlo, cuál herramienta usar para hacer el plan de proyecto... de tal forma que tuve que dedicar tiempo de investigación y aprendizaje sobre cómo utilizar Google Docs, cómo estructurar el proyecto, cómo guardar ciertas acciones para demostrar que no es plagio y demás.

- **Coste.**

Gracias a que se utilizaron herramientas y plataformas gratuitas ( SQLite, Google Colab, Power BI Desktop, Google Docs, YouTube, Chat GPT ) no hubo un coste monetario en cuanto a ellas. No hubo coste más allá del que haya hecho de forma personal en el gasto de energía que ha consumido el ordenador en todo el tiempo del proyecto ( cuyas cifras no tengo ).

## **7.Conclusiones y recomendaciones.**

A lo largo del proyecto, se han visto ya conclusiones y puntos débiles donde se podría mejorar y escalar, vamos a hacer un resumen y agrupación de todo eso, aquí.

### **1. Conclusiones.**

Las conclusiones han sido, visto de alguna manera, parciales. Me explico, en cuanto hemos dado con KPIs importantes, nos pusimos a ahondar en ellos, de tal forma que dejamos de lado la opción de probar más enfoques distintos en cuanto a análisis y dejamos por confirmar si no hay realmente ninguna otra variable que afecte de alguna forma que no estemos percibiendo.

Por ello digo parcial, en cuanto a que mientras no llevemos a la práctica dichas soluciones, no veremos realmente si hemos dado con conclusiones válidas o no, pero bueno, lo que vamos a utilizar es lo que nos tiene más pinta de respuesta correcta con los datos que hemos tratado. Serían las siguientes:

- Tanto las consultas en sql, como la visualización de gráficos en Power BI y el refuerzo y apoyo de los resultados de los modelos de machine learning, dejan en claro una idea principal, el peso de las horas de estudio es lo que más importa. Más concretamente en el intervalo de seis a ocho horas, los resultados son de los mejores.

- Ligado a lo anterior, la segunda conclusión con más peso a la que se ha llegado, es la forma en la cual se emplean esas seis u ocho horas. No es simplemente, si estás suspendiendo, pues estudia ocho horas y ya está. Por lo que aquí entraría la idea del uso de técnicas de estudio. Se tendría que preguntar a los nuevos alumnos ( al inicio de cada convocatoria, por ejemplo ), si usan algún tipo de técnica de estudio, en caso afirmativo, preguntar cuál o cuáles. A los que den una respuesta negativa, ofrecerles opciones introductorias de ejemplos de técnicas las cuales puedan empezar a utilizar y probar si hay o no mejoría en los resultados.
- Vimos también que otras variables, con no mucho peso, igualmente tienen el rol de potenciar positivamente la efectividad de las horas de estudios, tales hábitos como dormir bien, buena alimentación, actividad física, básicamente cuidado y mantenimiento del cerebro, que en última instancia es la herramienta que se usa a la hora de estudiar.

## 2. Recomendaciones.

La verdad es que se me iban viniendo muchas formas de mejorar o escalar un proyecto con estos objetivos, aunque algunas de éstas ideas, al igual que las conclusiones, habría que testearlas, pero repito, con los datos que tengo estas son las ideas que me parecen más relevantes y acertadas. Tendríamos los siguientes puntos:

- Ampliación de categorías. Ligado a las ya dichas conclusiones, si veo interesante añadir categorías que aporten información relacionada a las preguntas del cuestionario que se haría a los alumnos, usos de técnicas de estudios, en caso de que sí, cuál es. Poder profundizar más en las horas de estudio como por ejemplo, número de descansos y duración de los mismos.
- Con todas éstas nuevas categorías vendría por supuesto, un reentrenamiento de modelos, viendo ahora si hay alguna relación entre cuáles de las técnicas pueda afectar más de forma positiva a los resultados. Aunque cosas como los descansos en periodos largos de estudio, sea de primeras algo subjetivo ( cada uno puede tener una recuperación más lenta o rápida ) sí que puede valer para poner unas marcas orientativas. Porque esto al final es una habilidad que con la práctica va mejorando, por lo que si se está empezando, que hayan referencias de a lo que se debe llegar, sin que sean tampoco límites o metas obligatorias.
- Dentro de las recomendaciones se tenía que añadir también escalabilidad, teniendo en cuenta cómo es el proyecto, veo perfectamente encajable, el uso de Hadoop como arquitectura base. Sabiendo que el ritmo de la obtención de los datos será periódica ( cada convocatoria, por ejemplo ), se trataría de una

carga de datos por lote ( batch ), de tal forma que no haría falta un procesamiento en tiempo real.

Si su uso da buenos resultados, en los centros de forma individual, y se quisiera conectar por lo tanto todas las sedes de un mismo centro de estudios, se tendría que pasar a usar Hadoop en la nube, donde habría que contratar un proveedor para ello.

## 8. Referencias.

### Google Colab

Google. (n.d.). *Google Colaboratory*. <https://colab.research.google.com/>

### SQLite

SQLite. (n.d.). *SQLite Documentation*. <https://www.sqlite.org/docs.html>

### Power BI Desktop

Microsoft. (n.d.). *Power BI Documentation*. <https://powerbi.microsoft.com/>

### Google Docs

Google. (n.d.). *Google Docs Editors Help*. <https://docs.google.com/>

### Google Drive

Google. (n.d.). *Google Drive Help*. <https://drive.google.com/>

### ChatGPT (IA de OpenAI)

OpenAI. (2024). *ChatGPT (July 2024 version)* [Large language model].

<https://chat.openai.com/>

### Gemini (IA en Google Colab)

Google. (2024). *Gemini in Google Colab* [AI assistant]. <https://colab.research.google.com/>

### Kaggle

Kaggle. (2024). *Student Habits vs Academic Performance*. Kaggle.

<https://www.kaggle.com/datasets/jayaantanaath/student-habits-vs-academic-performance/data>

Las siguientes referencias, no fueron mencionadas en el documento, pero han aportado utilidad a la hora de la creación del proyecto, son videos tutoriales los cuales vi antes y durante el transcurso del proyecto.

- Relacionado con Power BI:

A2 Capacitación: Excel. (2025, febrero 10). *Power BI sin complicaciones* 🚀 Curso GRATIS para empezar YA [Video]. YouTube. [https://www.youtube.com/watch?v=mlO\\_OoGM5-I](https://www.youtube.com/watch?v=mlO_OoGM5-I)

A2 Capacitación: Excel. (2023, octubre 10). *Dax para Principiantes* [Video]. YouTube. <https://www.youtube.com/watch?v=rKreQw9JGvo>

- Relacionado con SQL:

Eze Talamona. (2024, julio 29). *Tutorial SQL Básico | Instalación PostgreSQL + Creación de Base de Datos y Tablas | 1/5* [Video]. YouTube. [https://www.youtube.com/watch?v=M3MG\\_3cRPJo](https://www.youtube.com/watch?v=M3MG_3cRPJo)

Eze Talamona. (2024, agosto 5). *Tutorial SQL Básico | Sentencia SELECT y Funciones de agregado | 2/5* [Video]. YouTube. <https://www.youtube.com/watch?v=mPWCRGHqfto>

Eze Talamona. (2024, agosto 12). *Tutorial SQL Básico | Cláusula WHERE | 3/5* [Video]. YouTube. <https://www.youtube.com/watch?v=kcuTsfOd9MI>


Eze Talamona. (2024, agosto 19). *Tutorial SQL Básico | Cláusula GROUP BY | 4/5* [Video]. YouTube. <https://www.youtube.com/watch?v=7NEBe49TqQI>

Eze Talamona. (2024, agosto 26). *Tutorial SQL Básico | Cláusula ORDER BY y LIMIT | 5/5* [Video]. YouTube. <https://www.youtube.com/watch?v=1Kt5HDyL8SY>

- Relacionado con Google Docs:

El Tío Tech. (2022, febrero 18). *Cómo usar Google Docs - Editor de Documentos de Google 2022* [Video]. YouTube. <https://www.youtube.com/watch?v=xApZhkNcJyY>

El Tío Tech. (2022, febrero 23). *10 Trucos de Google Docs que mejoran tu productividad | Guía 2022* [Video]. YouTube. <https://www.youtube.com/watch?v=S62iabuj9Aw>

El Tío Tech. (2022, julio 27).  *Cómo hacer un índice automático en Google Docs 2022* [Video]. YouTube. <https://www.youtube.com/watch?v=sAT0ExbuyFk>

## 9. Anexos.

Vamos a establecer una hoja de ruta para poder guiarnos mejor, dividiendo este punto en cuatro sub-anexos, del cero al tres, siendo el cero la extracción del archivo raw sobre el cual se construye el proyecto y del uno al tres dividiremos según los entornos utilizados.



- **Anexo 0, Fuente original.**

Empezamos por descargar el archivo .csv, el cual será nuestro dataset de la plataforma Kaggle, con nombre predeterminado de "student\_habits\_performance".

Una vez descargado, se procede a cargar en la nube, en la cuenta de Google Drive, desde donde se accederá al archivo posteriormente con Google Colab.

- **Anexo 1, Google Colab.**

Carga del archivo .csv desde Pandas.

```
import pandas as pd
df =
pd.read_csv('/content/drive/MyDrive/Searching/student_habits_performance.csv')
```

Muestra aleatoria del df.

```
df.sample(10)
```

Características del df.

```
df.info()
```

Comprobación de duplicados.

```
df.duplicated('student_id').sum()
```

Verificación de valores nulos.

```
df.isnull().sum()
```

Buscando los valores categóricos de una columna.

```
df['parental_education_level'].unique()
```

Imputación de la moda para la eliminación de valores nulos.

```
pel_mode = df['parental_education_level'].mode()[0]
df['parental_education_level'] =
df['parental_education_level'].fillna(pel_mode)
```

Agregación de varias columnas, comparando sus valores mínimos y máximos.

```
df[['age', 'study_hours_per_day', 'social_media_hours',  
'netflix_hours', 'sleep_hours', 'exercise_frequency',  
'attendance_percentage', 'exam_score']].agg(['min', 'max'])
```

Guardado del archivo limpio en Drive.

```
df.to_csv('/content/drive/MyDrive/Searching/student_habits_performance_  
limpio.csv', index=False)
```

Eliminación de una columna con sólo valores únicos, la cual no nos interesa para entrenar a los modelos.

```
df = df.drop(columns=['student_id'])
```

Aplicación del One-Hot Encoding.

```
df = pd.get_dummies(data=df, columns=['gender', 'part_time_job',  
'diet_quality', 'parental_education_level', 'internet_quality',  
'extracurricular_participation'], drop_first=True, dtype='bool')
```

Variable con la lista de columnas a escalar.

```
cols_a_escalar = ['age', 'study_hours_per_day', 'social_media_hours',  
'netflix_hours', 'attendance_percentage', 'sleep_hours',  
'exercise_frequency', 'mental_health_rating', 'exam_score']
```

Función personalizada para el escalado.

```
def escalador(col):  
    return (col - col.min()) / (col.max() - col.min())
```

Aplicación de la función a la lista de columnas.

```
df[cols_a_escalar] = df[cols_a_escalar].apply(escalador)
```

Guardamos el archivo escalado con el cual se alimentarán los modelos.

```
df.to_csv('/content/drive/MyDrive/Searching/student_habits_performance_  
procesado.csv', index=False)
```

- **Anexo 1.2, Google Colab, Modelos ML.**

Librerías utilizadas para el Random Forest.

```
import pandas as pd  
  
from sklearn.model_selection import train_test_split, GridSearchCV  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import mean_absolute_error, mean_squared_error,  
r2_score  
from sklearn.inspection import PartialDependenceDisplay
```

Separación del Label y Features.

```
df_X, df_y = df.drop("exam_score", axis=1), df['exam_score']
```

Separación en entrenamiento y test.

```
X_train, X_test, y_train, y_test = train_test_split(df_X, df_y,  
test_size=.2)
```

Definimos el modelo a utilizar.

```
random_forest = RandomForestRegressor()
```

Definimos los hiperparámetros.

```
param_grid = {  
    'n_estimators': [100, 200, 300],  
    'max_depth': [None, 10, 20],  
    'min_samples_split': [2, 5, 10],  
    'min_samples_leaf': [1, 2, 4]  
}  
  
rf_grid = GridSearchCV(estimator=random_forest, param_grid=param_grid,  
cv=5)
```

Entrenamiento del modelo.

```
rf_grid.fit(X_train, y_train)
```

Mejor configuración de hiperparámetros.

```
rf_best = rf_grid.best_estimator_
```

Vemos cuáles son concretamente los mejores.

```
rf_grid.best_params_
```

Ponemos a prueba al modelo con las predicciones.

```
y_pred = rf_best.predict(X_test)
```

Métricas utilizadas para evaluar el rendimiento del modelo.

```
mae = mean_absolute_error(y_test, y_pred)  
mse = mean_squared_error(y_test, y_pred)  
rmse = mean_squared_error(y_test, y_pred)**0.5  
r2 = r2_score(y_test, y_pred)
```

Copiamos el código del profe hecho en la clase del Titanic, para visualizar en una tabla los resultados, el feature importances.

```
resultados = pd.DataFrame(X_train.columns,
rf_best.feature_importances_).reset_index().sort_values(by = 'index',
ascending=False)
resultados['index'] = resultados['index'] * 100
```

Librerías usadas para el Linear Regression.

```
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
```

Variables con mayor peso.

```
coef_df = pd.DataFrame({
    'feature': X_train.columns,
    'coefficient': linear_model.coef_
})
```

Lo vemos en formato tabla.

```
coef_df['abs_coef'] = coef_df['coefficient'].abs()
coef_df = coef_df.sort_values(by='abs_coef',
ascending=False).drop(columns='abs_coef')

print(coef_df)
```

Librerías utilizadas para el Gradient Boosting Regressor

```
import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
```

Variables con los resultados de las métricas de los tres modelos.

```
lr_metricas = {'MAE': 0.06, 'MSE': 0.00, 'RMSE': 0.07, 'R2': 0.90}
rf_metricas = {'MAE': 0.06, 'MSE': 0.01, 'RMSE': 0.07, 'R2': 0.87}
gbr_metricas = {'MAE': 0.06, 'MSE': 0.01, 'RMSE': 0.07, 'R2': 0.88}
```

Creación de un data frame para una visualización y comparación más cómoda.

```
df_comparacion = pd.DataFrame({
    'Linear Regression': lr_metricas,
    'Random Forest': rf_metricas,
    'Gradient Boosting': gbr_metricas
})

print(df_comparacion)
```

- **Anexo 2, SQLite.**

En DB Browser for SQLite, se creó una base de datos ( .db ), con la respectiva creación de una tabla la cual imputamos las mismas columnas ( y sus formatos pertinentes ) que nuestro dataset para poder cargar los datos del archivo .csv a nuestra base de datos.

Nombre	Tipo	Esquema
Tablas (1)		
student_habits_performance_limpio		
student_id	TEXT	"student_id" TEXT
age	INTEGER	"age" INTEGER
gender	TEXT	"gender" TEXT
study_hours_per_day	REAL	"study_hours_per_day" REAL
social_media_hours	REAL	"social_media_hours" REAL
netflix_hours	REAL	"netflix_hours" REAL
part_time_job	TEXT	"part_time_job" TEXT
attendance_percentage	REAL	"attendance_percentage" REAL
sleep_hours	REAL	"sleep_hours" REAL
diet_quality	TEXT	"diet_quality" TEXT
exercise_frequency	INTEGER	"exercise_frequency" INTEGER
parental_education_level	TEXT	"parental_education_level" TEXT
internet_quality	TEXT	"internet_quality" TEXT
mental_health_rating	INTEGER	"mental_health_rating" INTEGER
extracurricular_participation	TEXT	"extracurricular_participation" TEXT
exam_score	REAL	"exam_score" REAL

Filtro por género.

```
SELECT
    gender,
    COUNT(student_id) AS estudiantes
FROM student_habits_performance_limpio
GROUP BY 1
```

Grupos de alumnos según edad.

```
SELECT age, COUNT (student_id) AS número_de_alumnos
FROM student_habits_performance_limpio
GROUP BY age
```

Número de suspensos.

```
SELECT COUNT(student_id) AS suspensos
FROM student_habits_performance_limpio
WHERE exam_score < 49.9
```

Suspensos agrupados por edad.

```
SELECT COUNT(student_id) AS suspensos, age
FROM student_habits_performance_limpio
WHERE exam_score < 49.9
GROUP BY age
ORDER BY age
```

Salud mental.

```
SELECT
    mental_health_rating,
    ROUND (AVG (exam_score),2) AS nota_media
FROM student_habits_performance_limpio
WHERE mental_health_rating IN (1,10)
GROUP BY 1
```

Asistencia según edad.

```
SELECT age, round(AVG(attendance_percentage),2) AS asistencia
FROM student_habits_performance_limpio
WHERE exam_score < 49.9
GROUP BY age
ORDER BY age
```

Efecto en la nota del nivel de estudios de los padres.

```
SELECT
    parental_education_level,
    ROUND (AVG (exam_score),2) AS nota_media
FROM student_habits_performance_limpio
GROUP BY 1
```

Relación entre la asistencia y el uso de redes sociales.

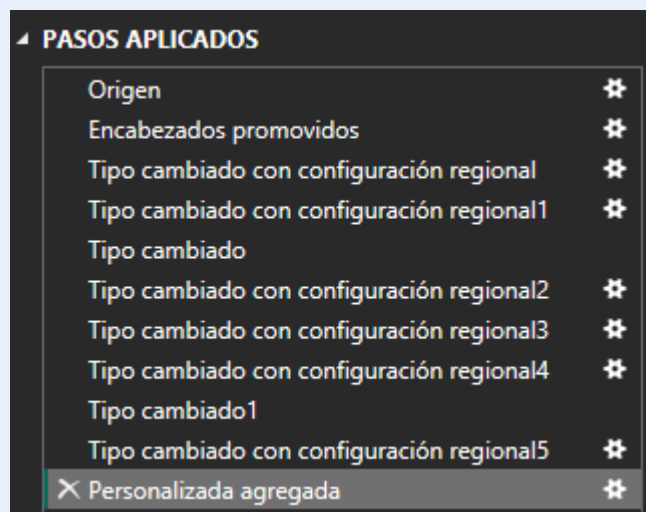
```
SELECT
  CASE
    WHEN social_media_hours < 3.5 THEN 'mitad_inferior_rrss'
    ELSE 'mitad_superior_rrss'
  END AS mitades_rrss,
  ROUND(AVG(attendance_percentage), 2) AS asistencia
FROM student_habits_performance_limpio
GROUP BY mitades_rrss
```

Nota media agrupada en intervalos de horas de estudio.

```
SELECT
  CASE
    WHEN study_hours_per_day >= 0 AND study_hours_per_day < 2 THEN '0_2 hrs'
    WHEN study_hours_per_day >= 2 AND study_hours_per_day < 4 THEN '2_4 hrs'
    WHEN study_hours_per_day >= 4 AND study_hours_per_day < 6 THEN '4_6 hrs'
    WHEN study_hours_per_day >= 6 AND study_hours_per_day <= 8 THEN '6_8 hrs'
    ELSE '8+ hrs'
  END AS intervalos_horas,
  ROUND(AVG(exam_score), 2) AS nota_media,
  COUNT(student_id) AS estudiantes
FROM student_habits_performance_limpio
GROUP BY intervalos_horas
ORDER BY intervalos_horas
```

- **Anexo 3, Power BI.**

En Power BI, vamos a: informe en blanco -> obtener datos de otro origen -> opción de importar datos a partir de un archivo de texto o CSV -> elegimos el archivo “student\_habits\_performance\_limpio” desde una ruta local y en vez de “Cargar”, damos a “Transformar datos”. Aquí hacemos los cambios pertinentes de formatos y cambios en la configuración regional.



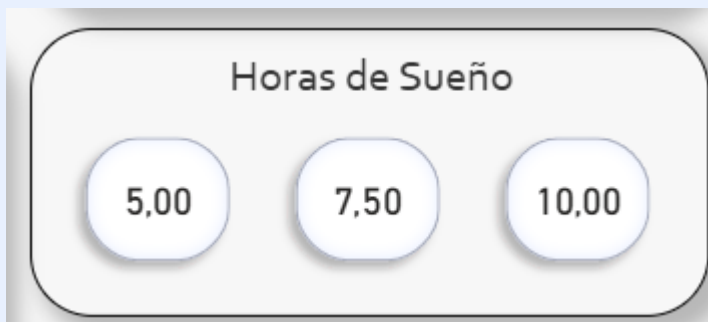
Después de haber ya cargado los datos, en cierto punto, tuvimos que hacernos una columna personalizada, con la siguiente script:

```
if [study_hours_per_day] >= 0 and [study_hours_per_day] < 2 then "0-2 h"  
else if [study_hours_per_day] >= 2 and [study_hours_per_day] < 4 then "2-4 h"  
else if [study_hours_per_day] >= 4 and [study_hours_per_day] < 6 then "4-6 h"  
else if [study_hours_per_day] >= 6 and [study_hours_per_day] < 8 then "6-8 h"  
else "8+"
```

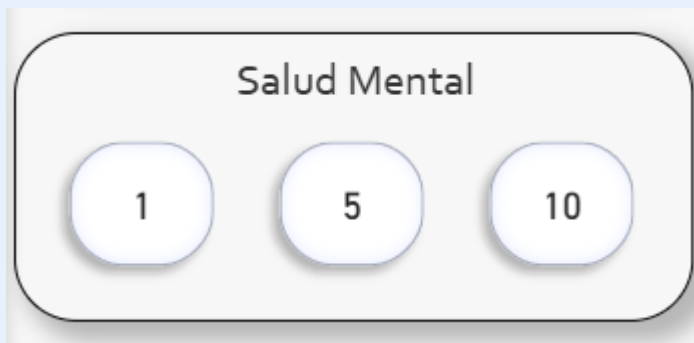
Slicer que filtra según horas de consumo de redes sociales.



Para horas de sueño.

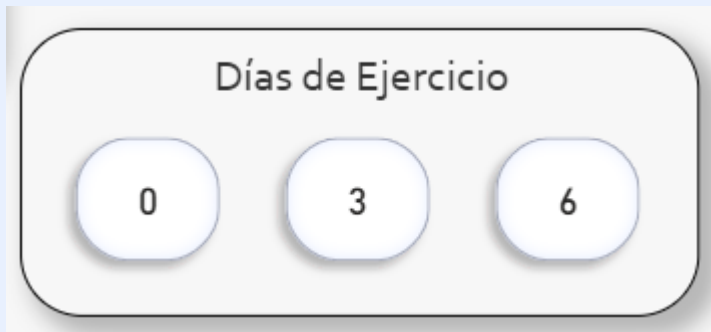


Salud Mental.





Y días de ejercicios.



Dos tarjetas que ofrecen las variables más relevantes.

