Tecnológico Nacional De México Campus Chihuahua II

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Modelo de Learning Analytics para predecir rendimiento en alumnos con datos escolares

TESIS
PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA

VALERIA SARAI AVILA GRAJEOLA

CHIHUAHUA, CHIH. A JUNIO 2022

DIRECTOR DE TESIS
M.C LEONARDO NEVÁREZ CHÁVEZ

CODIRECTORA DE TESIS
Dr. MARISELA IVETTE CALDERA FRANCO

Dictamen

Chihuahua, Chih., 25 de mayo 2022

M.C. MARÍA ELENEA MARTÍNEZ CASTELLANOS
COORDINADORA DE POSGRADO E INVESTIGACIÓN
PRESENTE

Por medio de este conducto el comité tutorial revisor de la tesis para obtención de grado de Maestro en Sistemas Computacionales, que lleva por nombre "MODELO DE LEARNING ANALYTICS BASADO EN INFORMACIÓN DEL SII Y DATOS GENERALES PARA PREDECIR EL RENDIMIENTO DE ALUMNOS", que presenta el C. VALERIA SARAÍ ÁVILA GRAJEOLA, hace de su conocimiento que después de ser revisado ha dictaminado la APROBACIÓN de la misma.

Sin otro particular de momento, queda de Usted.

Atentamente
La Comisión de Revisión de Tesis.

Leonardo Nevárez Chávez
M.C. LEONARDO NEVÁREZ CHÁVEZ
Director de tesis

DRA.MARISELA IVETTE CALDERA FRANCO
Co-Directora

Gregorio Ronquillo Máynez.
DR.GREGORIO RONQUILLO MAYNEZ
Revisor

M.C. ARTURO LEGARDA SAENZ
Revisor

DEDICATORIA

Primero que todo a Dios por darme la salud y la sabiduría para cursar todas las fases de estudio que hasta ahora he cumplido.

Dedico con todo mi corazón mi tesis a mis padres Aurora y Lorenzo por siempre estar apoyándome y brindándome todo su amor y sabiduría, por darme siempre lo mejor que pueden y aguantar todas mis tonterías, a mis abuelos Alicia y Vicente por ser así de loquillos y los mejores abuelos del mundo mundial, por darme una mamá tan buena y lista, por siempre hacerme mis pitufos y recogerme de la escuela, por quererme tanto como los quiero yo y por cuidarme ahora que estan en el cielo, a mis abuelos Eleno y Beatriz por darle lo mejor que pudieron a mi padre y por quererme y a pareja Javier por estar siempre a mi lado apoyándome, cuidándome, ayudándome y diciéndome que yo puedo lograr las cosas siempre. Muchas gracias a todos ustedes porque sin ustedes a mi lado ya sea físicamente o espiritualmente no lo habría logrado.

Tuve la bendición a diario de poder tenerlos en mi vida desde físicamente hasta espiritualmente para que pudieran llevarme por el camino del bien. Por eso les dedico todo mi trabajo en ofrenda por su paciencia, amor, palabras de aliento y consejos brindados los AMO con todo mi corazón.

De igual manera quiero agradecer a la institución Conacyt la cual me brindó su apoyo durante todo este tiempo en el que estuve como su investigadora.

AGRADECIMIENTOS

Me gustaría dedicar unas cuantas líneas de este gran logro para agradecer a todas aquellas personas que me han ayudado de alguna manera a llegar hasta este punto en mi carrera académica.

En primer lugar, gracias a mi familia, en especial a mis padres, abuelos y pareja, quienes han dado todo sin dudarlo para que crezca tanto como persona como en mi formación y educación, y que me han apoyado en todo momento, especialmente en los malos momentos y cuando más lo necesitaba sin ustedes nada habría sido posible y siempre estaré en deuda con ustedes.

Gracias Dios por permitirme tener y disfrutar de esta gran familia que me diste, gracias familia por apoyarme en cada decisión y proyecto que tuvo y tengo, gracias a la vida porque cada día me demuestra lo hermosa que es la vida y lo justa que puede llegar a ser. Gracias por creer en mí y gracias a Dios por permitirme vivir y disfrutar de cada día.

A todos y cada uno de los profesores que he tenido a lo largo de mi vida académica gracias, porque he aprendido algo nuevo de cada uno de ellos. Me gustaría poner especial énfasis en mi tutor de proyecto, Leonardo, que tanta paciencia ha tenido conmigo para ayudarme a llevar este proyecto a cabo, que me ha mostrado siempre una disponibilidad absoluta y que ha sacrificado gran parte de su tiempo de manera altruista.

A mis amigos y compañeros de la universidad (algunos son ambas cosas para mí), porque he aprendido que en la facultad y en la vida no sólo se adquieren conocimientos de cálculo, programación, etc., sino que además uno puede aprender cosas que nunca se habría imaginado, cómo jugar a los Tres toques, Bromas, etc. de ellos he aprendido muchísimo en cuanto a valores humanos, y han sido un apoyo fundamental a lo largo de la carrera, no me imagino haber llegado hasta donde estoy hoy sin ellos. Gracias por las risas, las penas, alegrías, enfados, viajes y muchísimas memorias que quedarán siempre en el recuerdo como una de las mejores épocas de mi vida.

Por último, al Tecnológico de Chihuahua, en concreto al campus Chihuahua II que ha sido como un segundo hogar para mi durante todos estos años y que me ha dado la oportunidad de realizar mis estudios y me ha visto crecer como persona (y físicamente).

No ha sido sencillo el camino hasta ahora, pero gracias a todos por sus palabras, amor, su inmensa bondad y apoyo. Les agradezco y les hago presente mi gran afecto hacia ustedes. Siempre los tendré en mi corazón.

RESUMEN

La presente tesis describe la realización de diferentes modelos de Learning Analytics como lo son: modelos de regresión simple, regresión múltiple, modelos de agrupamiento con el método de K-Means, redes neuronales y estadísticas básicas para la visualización de características de los datos logrando de ese modo la predicción de calificaciones y agrupamiento de alumnos haciendo uso solo de datos escolares como: edad, situación sentimental, apoyo por parte de los padres, consumo de alcohol, selección de escuela, entre otros atributos. Todo esto con el fin de obtener dicha predicción y de ese modo detectar alumnos en posible situación de riesgo académico como lo es la deserción escolar.

El desarrollo de los modelos es mediante el lenguaje de programación Python los cuales utilizan diferentes librerías como lo son *Numpy*, *Pandas*, *Matplotlib*, las cuales ayudan a la utilización de grandes cantidades de datos, usos de *datasets* y creación de gráficos; los cuales podrán ser utilizados y visualizados dentro una aplicación web en donde se visualizaran sus resultados obtenidos como: gráficas de agrupación, graficas de predicción, tablas de descripción de datos, recursos de ayuda para profesores y alumnos, etc. Y en donde dicha aplicación web podrá ser utilizada tanto por profesores como por alumnos.

Los resultados obtenidos dentro de este proyecto son buenos puesto que los modelos realizan una buena predicción y en donde el modelo más preciso es el modelo de redes neuronales el cual muestra un 0.89%, además se muestra el desarrollo de la aplicación web la cual se encuentra en funcionamiento óptimo.

ABSTRACT

This thesis describes the realization of different Learning Analytics models such as: simple regression models, multiple regression, grouping models with the K-Means method, neuronal networks and basic statistics for the visualization of data characteristics This mode the prediction of qualifications and grouping of students making use only school data such as: age, sentimental situation, support by parents, alcohol consumption, school selection, among other attributes. All this in order to obtain such prediction and thus detect students in possible academic risk situation such as school dropout.

The development of the models is through the Python programming language which use different libraries such as Numpy, Pandas, Matpletlib, which help the use of large amounts of data, uses of datasets and creation of graphics; which can be used and visualized within a web application where its results obtained such as: group graphs, prediction graphs, data description tables, help resources for teachers and students, etc. will be visualized, etc. And where this web application can be used by both teachers and students.

The results obtained within this project are good since the models make a good prediction and where the most precise model is the neural networks model which shows 0.89%, in addition the development of the web application is shown which is found in optimal operation.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I. INTRODUCCIÓN

1.1 Introducción

"Vivimos rodeados de datos, a diario consumimos una cantidad de información difícil de cuantificar, hablamos de *zettabytes, petabytes, terabytes*…, pero no somos meros consumidores de datos, ya que los últimos cálculos nos dicen que generamos aproximadamente 2,5 *exabytes* de los datos al día y que el volumen de datos crece exponencialmente igual que la velocidad de dichos datos" (ALogos E-learning, 2021).

Claro que dicho volumen y variedad de datos, así como la velocidad a la que se generan han dado paso a una nueva revolución tecnológica conocida como Big Data.

El conjunto de tecnologías asociadas a Big Data han sido aplicadas con éxito en múltiples ámbitos de la vida, uno de ellos es el ámbito educativo, donde las herramientas representan un poderoso aliado, por ejemplo, Learning Analytics (LA) o analítica del aprendizaje, el cual permite generar modelos analíticos dirigidos a personalizar el aprendizaje, comprender y predecir los procesos implícitos, así como optimizar los entornos en los que dicho aprendizaje se produce; y en donde la recopilación, análisis, implementación e incluso la intervención de datos ya no es solo dominio exclusivo del investigador especializado, sino que se trata de trasladar la infraestructura educativa hacia el entorno socio técnico, para de ese modo los educadores y alumnos logren obtener información de manera oportuna que podría mejorar los resultados finales (Santiago, R., 2017).

Como menciona Santiago, R. (2017), "LA reúne una serie de metodologías y herramientas tecnológicas las cuales permiten la obtención de mucha información sobre el aprendizaje de los alumnos esto a partir de la gran riqueza que proporcionan los datos recolectados de entornos virtuales de aprendizaje (EVA)". Es por ese motivo que la analítica del aprendizaje es considerada una ciencia en sus inicios con una jungla de jerga técnico-terminología compleja y exuberante la cual puede aplicarse a cualquier dinámica de aprendizaje y formación. Pero nos hace tener las siguientes incertidumbres:

- ¿Por qué es necesario el aprendizaje?
- ¿Qué información podemos extraer de los patrones de interacción de los alumnos?
- ¿Es posible mejorar los procesos de aprendizaje a partir de datos disponibles?

LA permite que los profesores puedan realizar un seguimiento individualizado de las acciones de los alumnos en los entornos de aprendizaje y de esa manera puedan detectar anticipadamente qué alumnos necesitan intervención o ayuda especializada.

LA se apoya de la minería de datos ya que es el proceso de hallar anomalías, patrones y correlaciones en grandes conjuntos de datos para redecir resultados y representa una de las disciplinas que más ha crecido en los últimos años; lo que ha provocado que las organizaciones hayan comprendido que las grandes cantidades de datos que residen en sus sistemas, pueden ser analizados y explotados para así obtener nuevos conocimientos a partir de los mismos.

En este proyecto en específico se estará utilizando la metodología CRISP-DM la cual es detallada más adelante y se justificará el porqué de esta selección.

1.2. Planteamiento del problema

La Educación es uno de los factores que más influye en el progreso de las personas y sociedades además de proveer conocimientos, enriquece la cultura, el espíritu, los valores y todo aquello que nos caracteriza como seres humanos (Casanova Cardiel, H. 2009).
Pero un problema visible para la educación mexicana, es la poca inversión pública por parte del Estado, ya que esto genera un deterioro dentro del sistema educativo y desafortunadamente ocasiona que México se mantenga como desde hace 15 años sobre una escala a la media de 500 puntos OCDE (Organización para la Cooperación y el Desarrollo Económicos), como lo podemos ver en la figura 1.1 en donde los alumnos mexicanos presentan déficit reprobatorio en las áreas de ciencias, lectura y matemáticas (López, 2019).

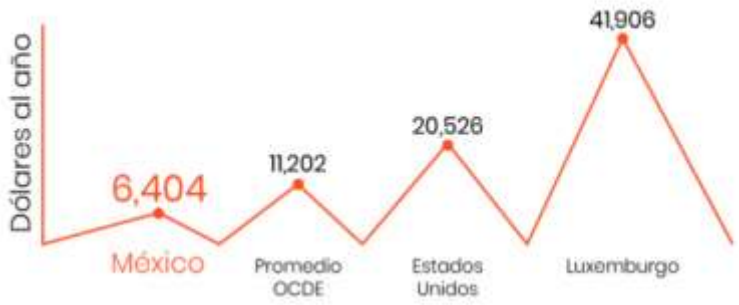


Figura1. 1 Inversión por alumno universitario de México y otros países de la OCDE.
Obtenida de: Rangel, M. (2020).

Como menciona Rangel, M. (2020), la deserción escolar en el sistema tecnológico es un fenómeno que se debe a varias causas tales como:

1. La falta de dinero, generado por la pérdida del empleo de los padres de familia, problemas de salud o endeudamientos inesperados; esto afectando directamente la solvencia económica familiar, es entonces cuando los hijos se ven obligados a dejar los estudios.

2. Tener que trabajar y estudiar, provocando una caída en el aprovechamiento escolar o más grave, la reprobación de las asignaturas (ver figura 1.2).

3. Nivel académico bajo, esto porque cuando un joven llega a un nivel universitario, se enfrenta a una exigencia mayor en cuanto a responsabilidades y cargas de trabajo.

4. Materias reprobadas, el reprobar constantemente las asignaturas crea problemas con los padres creando un desaliento y pensamientos negativos por parte del joven.

Figura1. 2 Ciclo de la deserción escolar.
Obtenida de: Rangel, M. (2020).

Una persona con título universitario gana hasta el doble de una persona que estudió solo hasta la preparatoria, aun así, solo el 24 % de los jóvenes mexicanos mayores de 18 años ingresan a una universidad y de este pequeño porcentaje es muy alto el número de estudiantes que suman a las estadísticas de la deserción escolar (Rangel, M. 2020).

Basándose en los índices que se viven en la actualidad dentro del sistema educativo y de los cuales ya se mencionó anteriormente se plantea la siguiente problemática:

¿Cómo predecir el rendimiento de los alumnos del sistema tecnológico para atender a tiempo a alumnos en situación de riesgo académico?

Con base a lo anterior, el Instituto opta por el desarrollo de modelos de LA y de una aplicación web que apoye a los docentes para identificar alumnos con posible riesgo académico.

1.3. Alcances y Limitaciones

En materia de rendimiento académico dentro del ámbito de la educación superior, la mayoría de las investigaciones relevantes presentan un marco de interés en la inclusión de factores personales (Daysi García-Tinizaray, K. O.-B.-D., 2014).

Además, al ser el learning Analytics un campo emergente en el caso de la educación aún no se ha desarrollado la cultura de utilizar y analizar los datos generados por el estudiante en los procesos de formación y aprendizaje en la plataforma virtual para determinar su influencia en el rendimiento académico, la deserción y/o graduación.

Bajo estas premisas la presente investigación se centra en determinar lo siguiente:

Alcances

- Contar con modelos de LA dentro del Tecnológico Nacional de México campus Chihuahua II.
- Visualizar dentro de una aplicación web, alumnos con posible riesgo académico.
- Tener una educación focalizada en alumnos con riesgo de deserción escolar.

Limitaciones

- No contar con el acceso a los datos concretos del Tecnológico Nacional De México Campus Chihuahua II.

1.4. Justificación

Actualmente las escuelas tienen información socioeconómica de sus estudiantes, calificaciones de materias y datos derivados del uso extensivo de los entornos virtuales de aprendizaje, entre otros.

Todos estos datos tienen el potencial de convertirse en información valiosa para la toma de decisiones y para anticiparse en el tiempo y así poder prevenir la deserción.

Por tal motivo la importancia de este proyecto, que surge de la necesidad de atender a tiempo a los estudiantes con riesgo académico, ya que no se cuenta con ningún modelo ni herramienta web dentro del instituto que estime el rendimiento de los estudiantes en los semestres consecutivos; lo cual auxiliará en la detección de alumnos en riesgo académico, asimismo que en un futuro próximo servirá de ayuda para la implementación y adaptación de los datos que se generan por parte del Tecnológico Nacional de México campus Chihuahua II.

Se decidió utilizar solo datos escolares ya que por la contingencia del Covid 19 no se logró tener acceso a los datos del SII de manera rápida, por ende, se decidió utilizar un data set publico el cual contenía datos escolares de estudiantes de secundaria. Es por ese motivo que tambien cambio el nombre del proyecto.

1.5. Objetivo

A continuación, se describen los objetivos específicos y el objetivo general para el desarrollo e implementación de los modelos y aplicación de Learning Analytics (LA).

1.5.1 Objetivo general

El objetivo del proyecto es la realización de modelos de LA, basados en información de data sets gratuitos con información escolar de alumnos para predecir el rendimiento y determinar a los que se encuentran en situación de riesgo académico. Además, realizar una aplicación web, en donde se verán visualizados los modelos desarrollados y recursos de ayuda para profesores.

1.5.2 Objetivos específicos

Para guiar el desarrollo técnico, empírico y aplicado en la presente investigación se plantean los siguientes objetivos específicos:

- Analizar los datos con los cuales contamos utilizando la metodología CRISP-DM para poder comenzar con el desarrollo de los modelos de LA.
- Desarrollar varios modelos de LA para verificar cuales realizan mejores predicciones del rendimiento escolar.
- Desarrollar una página web en donde se muestran a posibles alumnos en situación de riesgo.

- Desarrollar lo necesario para que la página web cuente con los lineamientos necesarios para que solo sea usada por alumnos y profesores registrados o parte del instituto.

1.6 Pregunta de investigación

¿Cómo predecir el rendimiento de los alumnos en situación de riesgo académico, para atenderlos a tiempo?

CAPÍTULO II. ESTADO DEL ARTE

A comienzos del siglo XXI, la tecnología web da un giro que facilita la participación directa de los usuarios, con la habilitación de lector-escritura, y que trajo consigo un ámbito de nuevas posibilidades, que incluye la recolección de nueva información acerca de las actividades de los usuarios (Berners-Lee, Hendler & Lassila, 2001).

La incorporación de entornos virtuales de aprendizajes (EVA), o sistemas de gestión de aprendizaje (LMS por su nombre en inglés), donde el uso de entornos virtuales de aprendizaje genera una huella virtual digital (como: registro de entrada, acciones, etc.) que aportan una gran cantidad de información acerca de las actividades de los usuarios, a estos datos también se les conoce como big data educacionales o big data a secas, que son definidos como un conjunto de datos cuyo tamaño excede la capacidad del software tradicional para su captura, gestión y análisis (Rojas-Castro, P. 2017).

El seguimiento de la huella digital trajo consigo una serie de nuevas áreas de estudio, tales como el Analytics o Educacional Data Mining, las cuales nos habla acerca del comportamiento de los estudiantes en estos ambientes o sobre las interacciones que allí acontecen (Campbell, DeBlois & Oblinger, 2007; Romero, 2008).

2.1 Antecedentes

A continuación, se describen los trabajos relacionados con el desarrollo e implementación de Learning Analytics (LA).

2.1.1 Pioneros de la analítica del aprendizaje

Dentro de un contexto de crecientes presiones para la rendición de cuentas dentro de la educación superior la aparición de Analytics viene a ayudar en la satisfacción y mejora de estas necesidades de aprendizaje y éxito académico; de esta manera, surgen una serie de experiencias dentro del área de la educación superior, en la implementación de LA, como los casos de: Baylor University, University of Alabama, Sinclair Community College, Northern Arizona University, o el célebre caso de la Purdue University y su programa llamado *Signals*, pionero en la aplicación exitosa de

LA. Donde estas iniciativas tempranas de analytics buscan predecir a los estudiantes que están en dificultades, permitiendo a los profesores o tutores personalizar la enseñanza o atender de forma precisa y acotada las necesidades de aprendizaje de los alumnos. No obstante, otros estudios comprenden las relaciones subyacentes entre interacciones y el rendimiento académico de los estudiantes o los niveles de participación y las tasas de deserción en los cursos en línea (Castro, 2016).

El sistema de *Signals* fue lanzado por Purdue University, este sistema fue el primero en su tipo el cual lograba rastrear el progreso académico de los estudiantes además de advertir a los mismos en tiempo real si necesitan trabajar en ciertas áreas. Este sistema cuenta con una amigable interfaz para el usuario la cual muestra por medio de luces (rojas, amarillas y verdes) parecidas a señales de tráfico, esas señales son seguidas por mensajes de su tutor o profesor brindando sugerencias sobre cómo pueden cambiar su comportamiento académico y mejorar sus calificaciones, como asistir a sesiones de ayuda o leer materiales adicionales (Signals, 2009).

Por ende el uso de las herramientas EVA exigen a los profesores que constantemente adapten sus cursos (tanto en lectura y contenido), para así asegurar comprensión, rendimiento y eficiencia en el aprendizaje de sus estudiantes, ya que se obtiene una retroalimentación comprensiva basada en datos tales como: actividades de aprendizaje (por ejemplo: lectura y discusión), así como los contenidos y resultados del aprendizaje y estudiantes, no obstante en la mayoría de los EVA hacen falta herramientas de EDM y LA que sean utilizables por los profesores y que soporten la investigación constante (ver figura 2.1) (Signals, 2009).

Figura2. 1 Pantalla Siglas desde navegador web.

Obtenido de: Signals: Applying Academic Analytics. (2010).

2.1.2 Learning analytics la narración del aprendizaje a través de los datos

La integración de las tecnologías en educación requiere de nuevas aproximaciones para conocer, controlar y mejorar los distintos contextos, roles y procesos implicados, es así que cuando los autores se encuentran cara a cara con un entorno virtual de aprendizaje (EVA), se ven en la necesidad de buscar alternativas reales para entender los procesos de enseñanza-aprendizaje, ya que todos los caminos de investigación e información convergieron hacia la analítica del aprendizaje que, aunque se lleva aplicando desde hace muchos siglos, toma un nuevo significado en nuestra actual era tecnológica digital (Santiago, R., 2017).

En consecuencia, la revolución tecnológica-educativa tenía que ir acompañada de un procedimiento analítico vinculado a la mejora, optimización y dominio de esta nueva era de cambio cuantitativo.

Ahora, tras algunos años de investigación y diversas publicaciones, se puede afirmar lo siguiente:

> *"En la analítica del aprendizaje se interpretan datos educativos mediante aproximaciones cuantitativas. Con ello se pueden entender, explicar y predecir los comportamientos de los alumnos. En consecuencia, se podrá mejorar el contexto educativo." (Santiago, R., 2017).*

2.1.3 Aplicación de análisis de aprendizaje para la predicción temprana del rendimiento académico de los estudiantes en el aprendizaje combinado

Como mencionan Lu, Huang & Lin (2018), "el aprendizaje combinado o mixto es donde puede combinar recursos digitales en línea con actividades tradicionales en el aula de tal modo que permite a los estudiantes lograr un mayor rendimiento en el aprendizaje a través de estrategias interactivas bien definidas. Por eso LA es un marco conceptual que forma parte de la educación en donde se analiza lo obtenido con los recursos digitales para predecir el desempeño de los estudiantes y proporciona intervenciones oportunas basadas en los perfiles de aprendizaje de los estudiantes".

2.1.4 El panorama actual de la analítica del aprendizaje en la educación superior

La integración generalizada de la tecnología digital en la educación superior (ES) influye tanto en las prácticas de enseñanza como de aprendizaje, y permite el acceso a datos, principalmente

disponibles en entornos de aprendizaje en línea, que se pueden utilizar para mejorar el aprendizaje de los estudiantes. El aprendizaje en línea que facilita el uso de la interacción y la comunicación asincrónicas y sincrónicas dentro de un entorno virtual, logrando de ese modo convertirse en una parte fundamental dentro de la ES y brindar de ese modo incremento de calidad (Viberg, Hatakka, Bälter & Mavroudi, 2018).

2.1.5 Predecir el desempeño de los estudiantes en instituciones de educación superior mediante el uso de análisis de aprendizaje por vídeo y técnicas de minería de datos

La digitalización ha infiltrado en todos los aspectos de la vida ya que las nuevas tecnologías emergentes tienen un impacto en nuestras vidas y cambian la forma en que hacemos nuestro trabajo diario, elevando de ese modo nuestro desempeño a una nueva altura. Actualmente el cambio que se ha presentado dentro del aprendizaje tradicional nos ha trasladado hacia un modelo en el que el aprendizaje puede tener lugar fuera del aula, facilitando de ese modo los diferentes atributos del alumno como son el aprendizaje virtual, verbal, auditivo y en solitario logrando de ese modo el aprendizaje mixto. El uso de tecnologías innovadoras para atender a diferentes estudiantes con aprendizaje combinado o mixto es donde se puede usar estas tecnologías para mejorar de ese modo sus habilidades cognitivas para sobresalir (Hasan, Oakaniappan & Mahmood, 2020).

2.1.6 ¿Les importa siquiera? Medición del valor de la privacidad de los estudiantes para el instructor en el contexto de la analítica del aprendizaje

Las instituciones dentro del área de educación superior están aumentando su capacidad y conocimiento dentro de las tecnologías del análisis del aprendizaje, tanto que investigadores y expertos señalan que la analítica del aprendizaje plantea importantes problemas de privacidad de los estudiantes y otras preocupaciones éticas. Si bien la analítica del aprendizaje es un punto muy importante el profesor es el principal usuario que utiliza dichos datos para poder ejecutar y efectuar cambios dentro de la asignatura para poder intervenir con los estudiantes (Jones, K., VanScoy, A., 2021).

Es ahí cuando surge el interés en agregar, extraer y analizar los datos de los estudiantes, ya que a menudo las prácticas socio-técnicas como la analítica del aprendizaje y minería de datos educativos intentan describir, predecir e intervenir dentro de sitios de aprendizaje para de ese modo mejorar los resultados del aprendizaje además de también reforzar y moldear la experiencia holística del estudiante de tal modo que aumente el éxito del mismo (Jones, K., VanScoy, A., 2021).

2.1.7 Optimizando el uso de análisis de aprendizaje a través de la dirección estratégica y la práctica de liderazgo: una perspectiva de la institución de educación superior

La analítica del aprendizaje es una práctica tecnológica multidisciplinaria emergente con el objetivo de producir aprendizaje efectivo para mejorar el logro de los estudiantes. Sin embargo, las aplicaciones de LA aún no han logrado cumplir con las expectativas, especialmente ya que aún LA se encuentra dentro de una etapa de infancia. (Lim, SM, Ghavifekr, S, 2021).

2.1.8 Apoyar el cambio a lo digital con análisis de aprendizaje centrados en el estudiante

La analítica del aprendizaje es conocida como el uso de métodos de la ciencia de datos para generar conocimientos educativos procesables, la cual se considera que tiene un gran potencial para impactar dentro de las prácticas de aprendizaje durante el cambio a lo digital, en lo particular puede ayudar a llenar una brecha de información crítica para los estudiantes creada por la ausencia de señales dentro del aula además de una necesidad mayor de autorregulación en el entorno en línea (Ocha & Wise 2021).

2.1.9 Predicción de calificaciones para el año escolar (Predicting Grades for the School Year)

Dentro del estado de arte también es importante el resaltar una aportación por parte Martínez Bachmann, (2018) quien comparte el proceso que realizó para poder predecir las calificaciones de estudiantes de secundaria en el área de matemáticas. Dentro de este proyecto se observa la implementación de algoritmos de regresión simple, algoritmos de muestreo con estadísticas básicas que ayudan a la predicción de la puntuación del alumno de forma individual, utilizando una gran serie de atributos y ver su comprensión o que tan útiles son para la predicción.

En el desarrollo de este proyecto se utilizan una serie de atributos para la obtención de la regresión simple, los más relevantes son:

- Calificaciones finales.
- Escuela de procedencia.
- Cantidad de integrantes dentro de la familia.
- Tutores.
- Trabajo de la madre.
- Trabajo del padre.
- Cantidad de tiempo libre
- Internet.
- Razón por la cual eligieron esta escuela.
- Tiempo semanal de estudio.
- Actividades extracurriculares.
- Calidad de relación con la familia.
- Cantidad de tiempo que sale con los amigos
- Estado de salud.
- Consumo de alcohol.
- Edad.
- Sexo.
- Educación máxima de la madre.

Una vez mencionados algunos de los trabajos relacionados con LA se elaboró una tabla comparativa en donde se visualizan todos los trabajos ya mencionados, pero de una manera más resumida (ver tabla 2.1).

Tabla 2. 1 Tabla comparativa de artículos valiosos

Obtenida de: Elaboración propia.

Tabla comparativa de artículos valiosos			
Nombre del artículo	Autores	Año	Resultados
Pioneros de la analítica del aprendizaje	Santiago, R.	2017	Dentro de los resultados obtenidos bajo el criterio de inclusión y exclusión se observó que los artículos recabados que contribuyen a esta revisión literaria ascienden a un total de 23 trabajos que cumplen con las características óptimas.
Aplicación de análisis de aprendizaje para la predicción temprana del rendimiento académico de los estudiantes en el aprendizaje combinado.	Lu, Huang & Lin	2018	Los resultados que se obtuvieron con el uso de diferentes tipos de algoritmos como lo son árboles de decisión y reglas CN2 aumentó la precisión de su proyecto de un 9% al 14%. Además de que también tuvieron mejoras de los algoritmos con los cuales lograron obtener correctamente 24 de 27 (89%) de aciertos de predicción.
El panorama actual de la analítica del aprendizaje en la educación superior	Viberg, Hatakka, Bälter & Mavroudi	2018	Este estudio presentó una amplia cobertura de investigación sobre LA en las ES. En donde la práctica e investigación en evolución prevalecen dentro de los estudios descriptivos y métodos de recopilación de datos interpretativos.
Predecir el desempeño de los estudiantes en instituciones de educación superior mediante el uso de análisis de aprendizaje por vídeo y técnicas de minería de datos.	Jones, K., VanScoy, A.	2021	La conclusión obtenida con la realización de este proyecto fue bastante buena ya que los datos que obtenían con sus algoritmos cuentan con un gran número de validez prediciendo de ese modo correctamente 629 estudiantes de 645. Una predicción del 88.39%.
Optimizando el uso de análisis de aprendizaje a través de la dirección	Lim, SM, Ghavifekr, S.	2021	Dentro de este caso de estudio se contribuyó a las prácticas estratégicas de dirección estratégica y liderazgo, logrando la

estratégica y la práctica de liderazgo: una perspectiva de la institución de educación superior.			optimización de LA. Los participantes en este estudio han expresado que sus puntos de vista, perspectivas, además de encuestados y hallazgos mostraron una solución estratégica para optimizar las implementaciones de LA.
Apoyar el cambio a lo digital con análisis de aprendizaje centrados en el estudiante.	Ocha & Wise.	2021	Dentro de este análisis se llegó a una conclusión donde LA puede ayudar a los estudiantes a comprender mejor cómo se involucran en las actividades de aprendizaje, pero deja en claro que es necesario el contar con paradigma centrado en el estudiante para poder ayudarlo a aceptar la analítica en su ecosistema de aprendizaje.
Predicción de calificaciones para el año escolar (Predicting Grades for the School Year)	Jaino Martínez Bachmann	2018	Con la realización de este Proyecto se obtuvo un análisis profundo para el desarrollo de algoritmos (regresión lineal simple y otros) que predicen que puntaje obtendrá un estudiante en funcionalidad de diferentes características.

CAPÍTULO III. MARCO TEÓRICO

3.1 ¿Por qué la minería de datos?

A lo largo de la historia dentro del mundo de LA se ha escuchado el término de la minería de datos o explotación de información, ésta es conocida como el proceso de extraer información útil comprensible y novedosa dentro de grandes volúmenes de datos, siendo su principal objetivo encontrar información implícita, que no es posible obtener mediante métodos estadísticos convencionales, los cuales están formados generalmente por registros provenientes de bases de datos operacionales o bien bodegas de datos (*Data Warehouse*) (Moine, 2012).

La minería de datos se ha centrado en su gran mayoría en la investigación de técnicas para la exportación de información y extracción de patrones (tales como: árboles de decisión, análisis de conglomerados y reglas de asociación); sin embargo, se ha profundizado el hecho de cómo ejecutar este proceso para poder obtener el "nuevo conocimiento", es decir con el uso de las metodologías las cuales permitirán llevar a cabo el proceso de la minería de datos en forma sistemática y no trivial (Como aplicar minería de datos., 2017).

3.1.1 Principales interesados y recolección de información

Dentro de este proyecto los estudiantes son los principales interesados asi como las instituciones y universidades, ya que el desempeño exitoso representa un papel muy importante en el crecimiento tanto social como económico es por ello que éste, es una preocupación primordial de las partes interesadas (educadores, administradores y corporaciones.); por tal motivo, la minería de datos educativos se ha convertido en un área de investigación muy importante para revelar conocimientos presentables y aplicables de grandes repositorios de datos educativos (Santiago, R, 2017).

El análisis de datos educativos ha generado recientemente el uso de herramientas como los son: el análisis del aprendizaje, el análisis académico, la minería de datos educativos, el análisis predictivo y el análisis de los alumnos, los cuales se convierten en un área de investigación innovadora;

creando así un punto en común entre todos estos términos. En la figura 3.1 se puede observar el proceso iterativo que se maneja dentro de los sistemas educacionales.

La cual explica cómo es que se inicia con os profesores diseñan el plan de enseñanza utilizan los sistemas educacionales ya sea tradicionales o tecnológicos, se utiliza el *Data Mining* para la recolección de datos que sean de utilidad de mejora y se pasa la retro alimentación tanto a estudiantes como a profesores y asi se cree un circulo de mejora.

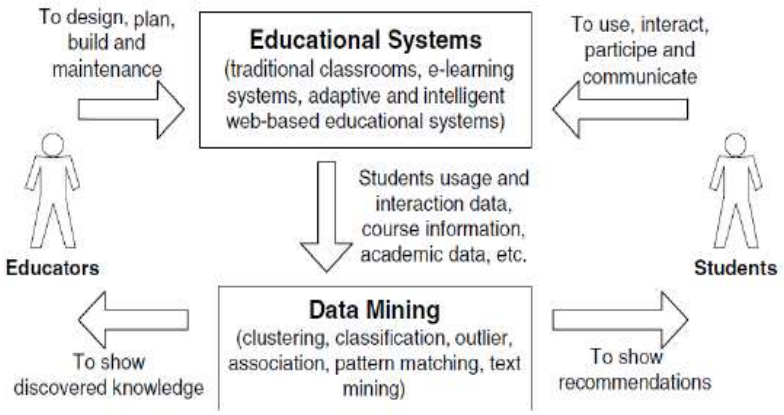


Figura 3. 1 Proceso para el desarrollo de hipótesis y pruebas en los sistemas educacionales. Obtenido de: Santiago, R. (2017).

La integración de las tecnologías en la educación requiere de nuevas aproximaciones para poder conocer, controlar y mejorar los distintos contextos, roles y procesos implicados ya que dentro de la analítica del aprendizaje se interpretan los datos educativos mediante aproximaciones cuantitativas; con ello se pueden entender, explicar y predecir los comportamientos de los alumnos con el fin de mejorar el contexto educativo (Santiago, R., 2017).

La recolección básica de los datos necesarios se realiza mediante la elaboración de encuestas u observaciones para posteriormente anotarlas, estas dos técnicas primarias de recolección de datos se encuentran a partir de datos de la interacción de los alumnos con distintos contextos tales como entornos virtuales de aprendizaje, páginas web o dispositivos móviles como se puede observar en la figura 3.2 cuáles son los principales elementos de colección primarios para la

obtención de datos los cuales en la actualidad estan a disposición de todos puesto que la tecnología está al alcance de

la mayoría de las personas como lo son celulares, ordenadores, correos, entre otros. (Santiago, R., 2017).

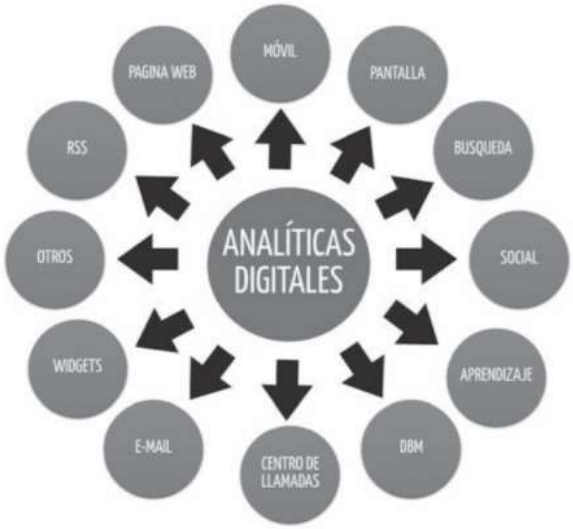


Figura 3. 2 Principales elementos para la recolección primaria de datos por parte de los estudiantes. Obtenida de: Santiago, R. (2017).

Según menciona Santiago, R. (2017), dentro de los procesos obtención de datos educativos existen varias fuentes que se dividen en fases, procesos, categorías, tipos de dato y técnicas de análisis.

Categorías

- Sistemas centralizados (LMS): Datos educativos donde el análisis proviene de una fuente específica.
- Sistemas descentralizados: El cual recupera datos de diferentes sistemas como los datos WWW y cursos en línea abiertos masivos.

Procesos

Predictivas

- Clasificación o discriminación. (en estadística).
- Clasificación suave.
- Estimación de probabilidad de clasificación.
- Categorización.
- Regresión.

Descriptivas

- Agrupamiento (clusterings).
- Correlaciones y factorizaciones.
- Reglas de asociación.

Tipos de datos

- Cualitativos: son los datos no estructurados, en los cuales se pueden encontrar conversaciones, imágenes y mensajes en foros y redes sociales.
- Cuantitativos: estos son datos estructurados, los cuales se pueden ordenar y utilizar directamente para cálculos matemáticos, dichos datos te pueden mostrar información como: mensajes enviados dentro de foros, número de accesos a recursos, intentos en un examen o tiempo de visualización de un video.

Técnicas de análisis

- Técnicas algebraicas y estadísticas.
- Técnicas bayesianas.
- Técnicas basadas en conteos de frecuencias y tablas de contingencia.
- Técnicas basadas en árboles de decisión y sistemas de aprendizaje de reglas.
- Técnicas relacionales, declarativas y estructurales.
- Técnicas basadas en redes neuronales artificiales.
- Técnicas basadas en núcleo y máquinas de soporte vectorial.
- Técnicas estocásticas difusas.
- Técnicas basadas en casos, en densidad o distancia.

3.1.2 Ayuda de la analítica del aprendizaje

La recolección de datos para su posterior análisis, tiene como objetivo primordial entender mejor al alumno, puesto que LA ayuda a estudiar el pasado, presente y el futuro del estudiante.

Sin duda el futuro es un tema complicado de trabajar ya que eso implica aplicar una serie de técnicas de predicción, modelos de datos predictivos y aproximaciones estadísticas de niveles superiores, los cuales requieren de altos conocimientos; algunas de las herramientas y modelos, facilitan la labor de cálculo y aplicación de conjuntos educativos ya que requiere conocer los datos para así saber cuándo aplicarlos en la figura 3.3 se puede observar el uso de la analítica del aprendizaje dentro del presente, pasado y futuro, en los cuales como brindan información, ayudan en la gestión de los datos y en la utilización de los mismos para la creación de modelos (Santiago, R., 2017).

Figura 3. 3 Uso de la analítica al paso del tiempo.
Obtenido de: Santiago, R. (2017).

3.2 Learning Analytics

Con el gran crecimiento que se ha tenido dentro de la tecnología y la innovación en este último tiempo, se ha podido lograr que las instituciones de educación superior (IES) puedan utilizar los diferentes tipos de sistemas de aprendizaje, pero primero es necesario revisar y definir la analítica del aprendizaje a continuación mencionamos a diferentes autores y como es que ellos definen a LA:

Como menciona Elías (2011), “la analítica del aprendizaje es un campo emergente en el que se utilizan sofisticadas herramientas analíticas para mejorar el aprendizaje y la educación. Se relaciona estrechamente en una serie de campos de estudio como son la inteligencia

empresarial, análisis web, análisis académico, datos educativos, minería de datos y análisis de acciones”.

Otra de las definiciones que podemos encontrar es la que mencionan Hasan, Palaniappan & Mahmood (2018) “LA se define como el uso de datos estadísticos y modelos explicativos predictivos para obtener información de los estudiantes para de ese modo actuar sobre cuestiones complejas que afectan al alumno, en donde LA implica el análisis de datos de los alumnos y sus actividades para mejorar la experiencia de aprendizaje del alumno”.

También como se menciona en el libro De Amo (2017), “la analítica del aprendizaje es donde se pueden interpretar los datos educativos mediante aproximaciones cuantitativas. Con ello se pueden

entender, explicar y predecir los comportamientos de los alumnos para en consecuencia mejorar el contexto educativo”.

3.3 Modelos y definiciones

Dentro de los elementos que se pretenden desarrollar en este proyecto se explican los modelos que se encargaran de hacer las predicciones para los estudiantes algunos de ellos son:

3.3.1 Modelo de Regresión simple

El análisis de regresión lineal simple es el más utilizado y el más sencillo de todos. Se trata de estudiar el efecto de una de las variables independientes sobre una única variable dependiente de la primera o que al menos a un nivel teórico se considera que es dependiente (Roldán, P. N., 2021).

Fórmula

$$y = B0 + B1\ x + \varepsilon$$

Donde y representa a la variable dependiente, x la variable independiente, B0, B1 son parámetros del modelo y ε representa el residuo o error. La función de ε es explicar la posible variabilidad de los datos que no pueden explicarse a través de la relación lineal de la fórmula (ver figura 3.4) (Roldán, P. N., 2021).

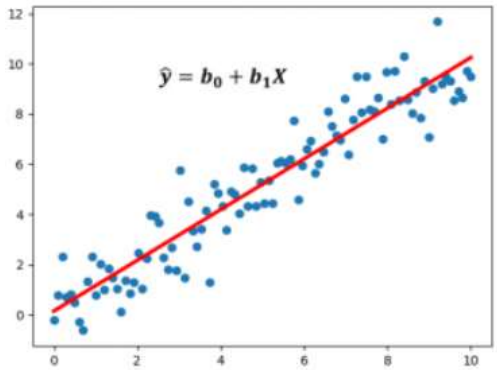


Figura 3. 4 Ejemplo regresión simple

Obtenido de: Roldán, P. N. (2021).

3.3.2 Regresiones múltiples

En el caso de la regresión lineal múltiple nos encontramos con un modelo que sencillamente cuenta con más de una variable independiente. Este modelo se aplica cuando se tienen razones para creer que hay más de un factor que afecta a la variable de estudio (Regresión lineal múltiple en Python., 2020).

Fórmula

$$Y = 0 + B1*X1 + B2*X2 + \ldots + Bn*Xn + \varepsilon$$

Donde Y es la variable dependiente X1, X2,… Xn son las variables independientes B1, B2,… Bn son los parámetros y ε sigue representando el posible error existente.

La evaluación de un modelo de regresión múltiple, así como la elección de qué predictores se deben incluir en el modelo ya que es uno de los pasos más importantes en la modelización estadística, además de que los valores de cada observación son independientes de los otros ya que esto es especialmente importante de comprobar cuando se trata de mediciones temporales (Regresión lineal múltiple en Python., 2020).

Otro de los puntos importantes sobre este método es lo que se mencionan los autores Lang, C., Siemens, G., Wise, A., & Gasevic, D. (Eds.) 2017). "Se recomienda que el número de observaciones sea como mínimo entre 10 y 20 veces el número de predictores del modelo".

A continuación, podemos ver en la figura 3.5 un ejemplo de como se ve una vez realizado e implementado el modelo de regresión lineal múltiple

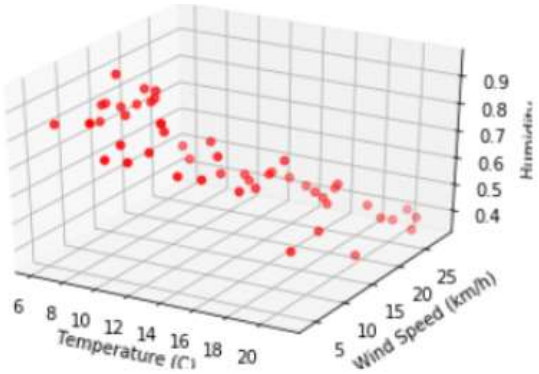


Figura 3. 5 Ejemplo de regresión lineal múltiple.

Obtenido de: Regresión lineal múltiple en Python. (2020).

3.3.3 Método K-Means

El algoritmo *K-means* (MacQueen, 1967) agrupa las observaciones en un número predefinido de *K clusters* de forma que, la suma de las varianzas internas de los *clusters*, sea lo menor posible.

Existen varias implementaciones de este algoritmo, la más común de ellas se conoce como *Lloyd's*. En la bibliografía es común encontrar los términos *inertia*, *within-cluster sum-of-squares* o varianza *intra-cluster* para referirse a la varianza interna de los *clústers* (Duk, D., 2019).

Este algoritmo garantiza que, en cada paso, se reduzca la intra-varianza total de los *clusters* hasta alcanzar un *óptimo local*. Debido a que el algoritmo de *K-means* no evalúa todas las posibles distribuciones de las observaciones sino solo parte de ellas, los resultados obtenidos dependen de la asignación aleatoria inicia. Por esta razón, es importante ejecutar el algoritmo varias veces (25-50), cada una con una asignación aleatoria inicial distinta, y seleccionar aquella que haya conseguido una menor varianza total (Duk, D., 2019).

En la figura 3.6 un ejemplo de cómo se ve una vez realizado e implementado el método *K-Means*.

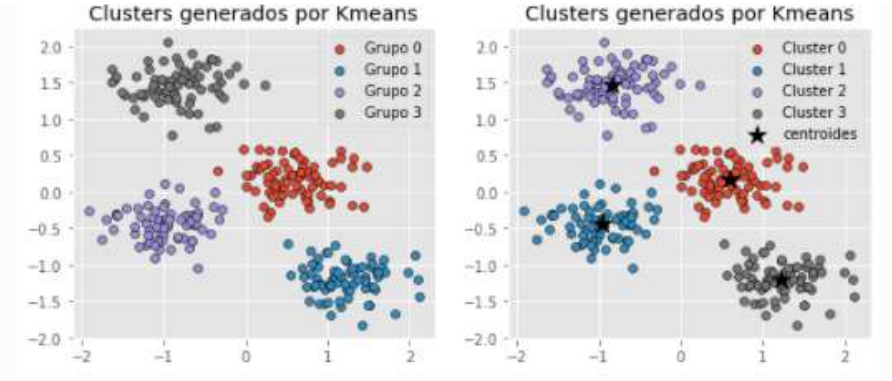


Figura 3. 6 Ejemplo del modelo K-Means con agrupaciones de 4 clustering y centroides.

Obtenido de: Joaquín Amat, R. (2020).

3.3.4 Método Redes Neuronales

Esté método es un modelo simplificado que emula el modo en que el cerebro humano procesa la información. Funciona simultaneando un número elevado de unidades de procesamientos interconectadas que parecen versiones abstractas de neuronas. Las cuales tienen unidades de procesamiento organizándose por capas (Bajo, S. N. 2002).

Permitiendo de ese modo extraer información útil y producir inferencias a partir de los datos disponibles gracias a su capacidad de aprendizaje. Sus propiedades como reconocedores de patrones altamente a errores permiten combinar las cualidades del razonamiento humano con la lógica precisa y la memoria de los ordenadores, por lo que resultan de gran utilidad (Bajo, S. N. 2002).

3.4 Entorno de desarrollo, herramientas y librerías

A continuación, se describen los entornos de trabajo que se utilizaron, las herramientas y las librerías implementadas para el desarrollo del proyecto.

3.4.1 Entorno de desarrollo

El desarrollo de este proyecto se llevará a cabo dentro del entorno de desarrollo Anaconda este es un entorno de distribución libre y abierta de los lenguajes de programación Python y R. Es muy utilizado dentro de la ciencia de datos y aprendizaje automático (machine learning). Ya que logra el procesamiento de grandes volúmenes de información análisis predictivo y cómputos científico (ANALITICA BIG DATA, 2021).

Las diferentes versiones de los paquetes se administran mediante el sistema de gestión de paquetes *conda*, el cual lo hace bastante sencillo de instalar, correr, y actualizar software de ciencia de datos y aprendizaje automático como puede ser *Scikit-team, TensorFlow y SciPy*.

Algunos de los entornos específicos que se estarán utilizando con ayuda de Anaconda son:

Spyder este es un entorno de desarrollo integrado (IDE) gratuito que se incluye con Anaconda. Está diseñado por y para científicos, ingenieros y analistas de datos. Cuenta con una combinación única de la funcionalidad avanzada de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y capacidades de visualización de un paquete científico. Además, *Spyder* ofrece integración incorporada con muchos paquetes científicos populares, incluidos *NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy,* entre otros.

3.4.2 Lenguajes de programación

Algunos de los lenguajes que se estarán utilizado para el desarrollo de este proyecto son:

Python

Este es un lenguaje de programación interactivo cuya filosofía hace hincapié en su código. Este lenguaje de programación es multiparadigma ya que soporta parcialmente la orientación a objetos, programación imperativa y en menor medida la programación funcional de lenguaje (Santander Universidades, 2022).

También posee una licencia de código abierto lo cual lo clasifica como uno de los lenguajes más populares.

C#

C# pronunciado *'si sharp'* en inglés, es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET. Es uno de los lenguajes de programación diseñados para lenguaje común, ya que su sintaxis deriva de C/C++ y utiliza el modelo de objetos de la plataforma *.NET* similar al que se utiliza en Java (Marketing, 2020).

JavaScript

Éste es un lenguaje de programación ligero que es conocido comúnmente como un lenguaje de scripting (secuencia de comandos) para las páginas web y que además es usado en muchos entornos fuera del navegador (Ramos, R., 2021).

JavaScript es un lenguaje de programación basado en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (Ramos, R., 2021).

React

React o también llamada *React.js*, es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página (Coalla, J., 2021).

3.4.3 Librerías

Dentro del entorno de programación en el cual se está desarrollando este proyecto llamado Anaconda se estarán manejando paquetes que ayudaran a la creación de los modelos, dichos modelos son conocidas como librerías.

Éstas son un conjunto de archivos que se utilizan para desarrollar software. Suelen estar compuestas de códigos y datos con el fin de ser utilizadas por otros programas de forma autónoma (Gómez, P., 2021).

Matplotlib

Matplotlib es una librería de Python especializada en la creación de gráficos en dos dimensiones, la cual ayuda con la creación y personalización de gráficos como lo son: diagrama de barras, histogramas, diagramas de sectores, etc. (Alberca, A. S., 2020).

Scikit-Learn

Scikit-lean es una librería que cuenta con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Además, presenta la compatibilidad con muchas más librerías útiles para las predicciones (Jauregui, A. F., 2022).

Numpy

Ésta librería se especializa en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos, además de que incorpora una clase de objetos llamados *arrays* que permite las correlaciones de datos de un mismo tipo en varias dimensiones y funciones muy eficientes para su manipulación (Alberca, A. S., 2020).

Pandas

Librería de Python especializada en el manejo y análisis de estructuras de datos. Definiendo nuevas estructuras de datos, pero con nuevas funcionalidades, además permite leer y escribir fácilmente ficheros en formato *CSV*, *Excel* y bases de datos *SQL*, también permite acceder a los datos mediante índices o nombres para filas y columnas, entre otras características (Alberca, A. S., 2020).

3.5 Metodología

Con el gran crecimiento dentro de la rama de la minería de datos y la analítica del aprendizaje surgen una serie de metodologías las cuales emplean un enfoque sistemático para poder llevar a cabo el proceso de extracción de información, una de esas metodologías es:

CRISP-DM (Cross Industry Standard Process for Data Mining) esta es una metodología creada por el grupo de empresas *SPSS* y Daimer Chryler en los años 2000, ésta se enfoca en la creación de un modelo de minería de datos que describa de manera más concreta todo lo que los expertos en esta materia abordan. Actualmente es una de las guías de referencia más utilizada dentro del desarrollo de proyectos de Data Mining (Moine et al., 2011) (ver figura 3.8).

Dentro de la metodología se trata de realizar un conjunto de tareas definidas en cuatro niveles (fases, tareas generales, tareas específicas e instancias del proceso), organizadas de forma jerárquica. El nivel superior está organizado por seis etapas y se caracteriza por enfatizar en los detalles de cada uno; dividiéndose en diferentes actividades y tareas (Aquino et al., 2015).

A continuación, se describen las fases dentro de esta metodología (ver figura 3.7) (Moine et al., 2011).

1. ***Comprensión del negocio:*** Probablemente la más importante y útil para reunir las tareas de comprensión de los objetivos y requerimientos del proyecto desde una perspectiva de negocio, con el fin de convertirlos en objetivos técnicos y en un plan de proyecto.

Dentro de esta fase se obtienen las siguientes tareas generales:

- Determinar los objetivos del negocio.
- Evaluación de la situación.
- Determinar los objetivos.

- Realizar plan del proyecto.

2. ***Comprensión de los datos****:* Esta fase comprende la recolección inicial de los datos con el objetivo de establecer un primer contacto con el problema, para así acostumbrarse a ellos, e identificar su calidad y establecer las relaciones más evidentes las cuales permitan definir las primeras hipótesis.

Dentro de esta fase obtienen las siguientes tareas generales:

- Recolectar datos iniciales.
- Descripción de los datos.
- Exploración de los datos.
- Verificar la calidad de los datos.

3. ***Preparación de los datos***: Esta fase procede a preparar y adaptar los datos en base a la técnica de minería de datos que se va a utilizar posteriormente, estas pueden ser visualización de datos, búsqueda de relaciones entre variables u otras medidas para explotación de datos.

Dentro de esta fase se obtienen las siguientes tareas generales:

- Seleccionar los datos.
- Limpiar los datos.
- Construir los datos.
- Integrar los datos.
- Formateo de los datos.

4. ***Modelado***: Esta fase de CRISP-DM es la que se encarga de seleccionar las técnicas de modelado más apropiadas para el proyecto de minería de datos. Utilizando una serie de criterios para una mejor persecución de la misma.

Dentro de esta fase se obtienen las siguientes tareas generales:

- Escoger técnica de modelado.
- General el plan de prueba.
- Construir el modelo.
- Evaluar el modelo.

5. ***Evaluación:*** Esta fase es la que se encarga de evaluar el modelo, teniendo en cuenta el cumplimiento de los criterios de éxito del problema a resolver. Debe considerar además que la fiabilidad calculada para el modelo se aplica solamente para los datos sobre los que se realizó el análisis.

Dentro de esta fase se obtienen las siguientes tareas generales:

- Evaluar los resultados.
- Revisar el proceso.
- Determinar los próximos pasos.

6. ***Despliegue o implementación:*** Esta es la última fase dentro de CRISP-DM, dentro de esta fase se transforma el conocimiento obtenido en acciones dentro del proceso de negocio.

Dentro de esta fase se obtienen las siguientes tareas generales:

- Planear la implementación.
- Planear la monitorización y mantenimiento.
- Producir el informe final.

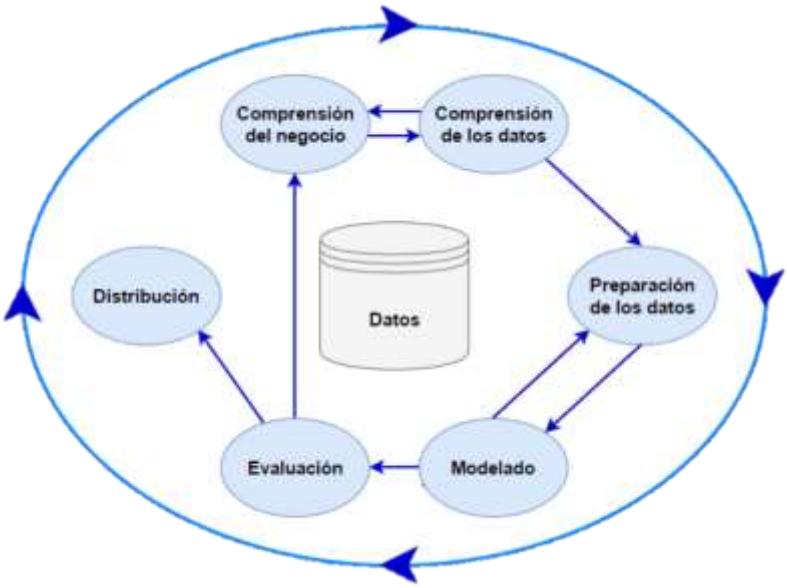


Figura 3. 7 Secuencia del proceso CRISP-DM.
Obtenida de: CRISP-DM: los 6 pasos del proceso de Data Mining - Blog Smartup. (2019).

Otro de los puntos importantes que podemos obtener con el uso de esta metodología es que la sucesión de las fases no es necesariamente rígida, ya que cada fase es descompuesta en varias tareas generales de segundo nivel que pasan a tareas específicas; es decir, *CRISP-DM* establece un conjunto de tareas y actividades para cada fase del proyecto, pero no especifica cómo llevarlas a cabo (Juan Miguel Moine, 2012).

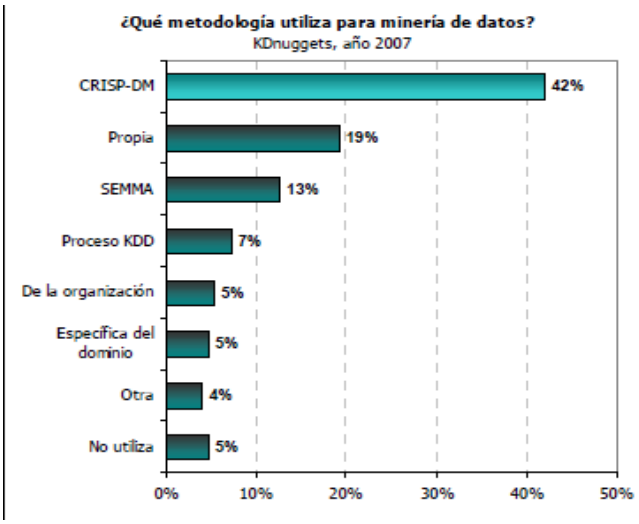


Figura 3. 8 Encuesta realizada por KDnuggets 2007.
Obtenida de: Castro, P. R. (

CAPÍTULO IV. DESARROLLO

Después de recabados los datos necesarios para este proyecto, se comenzó el desarrollo y análisis de los datos con ayuda de la metodología CRISP-DM lo primero que se realizó es:

4.1 Comprensión del negocio

El proyecto que se está desarrollando es para el Tecnológico Nacional De México Campus Chihuahua II, este establecimiento es una escuela de educación superior ubicada en la ciudad de Chihuahua México, fue fundada el 14 de septiembre de 1987, actualmente se encuentra ubicado en: Av. De las industrias 11101, complejo industrial sus colores académicos son el rojo y amarillo, su mascota es el bisonte su lema es: Enseñar para producir… Producir para crecer…, y sus siglas son: ITCH II (Red de Portales University Page., 2021).

En 2020, **Tecnológico Nacional De México campus Chihuahua II** tuvo 4,840 matriculados, de los cuales el 61.2% fueron hombres (2,964) y 38.8% fueron mujeres (1,876).

Actualmente la directora es Dra. Luisa Yolanda Quiñones Montenegro y las licenciaturas que ofrece son las siguientes:

- Licenciatura en Administración.
- Ingeniería en Informática.
- Arquitectura.
- Ingeniería Industrial.
- Ingeniería en Sistemas Computacionales.
- Ingeniería en Gestión Empresarial.
- Ingeniería en Diseño Industrial.

También se cuenta con las siguientes maestrías y doctorados:

- Maestría en Sistemas Computacionales.
- Maestría en Ingenieria Industrial.
- Maestría en Arquitectura.
- Doctorado en Ciencias de la Ingeniería.

4.1.1 Determinar los Objetivos del negocio

Dentro de esta tarea lo primero que se realizó fue la determinación del problema el cual es cómo podemos predecir el rendimiento de los alumnos del sistema tecnológico para así poder atenderlos a tiempo y reducir su riesgo académico. Por ende, es de vital importancia la utilización de LA ya que se necesita hacer un análisis de datos relacionados con el estudiante para de ahí poder aplicar una serie de algoritmos los cuales ayudarán con las predicciones (Red de Portales University Page., 2021).

Los objetivos generales que podemos encontrar dentro del Tecnológico Nacional de México campus Chihuahua II son:

- El reducir el índice reprobatorio.
- Aumentar el índice de alumnos que llegan al final de su carrera.
- El aviso oportuno de alumnos en riesgo de deserción.

Por último, se tiene la definición de criterios de éxito los cuales son regidos por tres ejes estratégicos (Tecnológico Nacional De México Campus Chihuahua II, 2019).

1. Calidad educativa, cobertura y formación integral;
2. Fortalecimiento de la investigación, el desarrollo tecnológico, la vinculación y el emprendimiento:
3. Efectividad organizacional, así como a su Eje transversal Evolución con inclusión, igualdad y desarrollo sostenible.

4.1.2 Evaluación de la situación

Dentro de la situación actual del instituto se observa que actualmente no se cuenta con ningún sistema u aplicación web que ayude a la determinación y/o detección de alumnos con posible situación de riesgo académico.

El conocimiento que se tiene sobre esto es que, si se cuentan con los datos necesarios para poder llevar a cabo este proyecto, además de que, si se tiene un balance con el costo y beneficio, esto porque no se necesita de grandes cantidades de dinero para poder desarrollarlo.

4.1.3 Determinar los objetivos

LA es una de las herramientas tecnológicas que están entrando dentro del ámbito de la educación el cual permite que los profesores analicen más a profundidad a sus alumnos. Por eso algunos de sus objetivos son:

- ~ Seguimiento del proceso del alumno.
- ~ Analizar a profundidad los datos generados por el alumno.
- ~ Anticipar y prevenir el fracaso o deserción escolar.
- ~ Mejora de estrategias de enseñanza y aprendizaje.

4.1.4 Realizar el plan del proyecto

El plan que se tiene para llevar a cabo este proyecto es el siguiente:

- ~ Analizar con ayuda de librerías especializadas de Python los datos proporcionados por parte de la escuela.
- ~ Implementar una serie de modelos como: regresiones simples y múltiples, clasificaciones, visualizaciones de estadísticas básicas, agrupaciones y predicciones con redes neuronales.
- ~ Realizar una aplicación web en donde se visualizará un modelo de learning analytics.

4.2 Comprensión de los datos

Dentro de esta sección se inicia con el análisis de los datos, que permite establecer el primer contacto directo con el problema a resolver. Aquí se identifica la cantidad de los datos, sus tipos y las relaciones entre ellos.

4.2.1 Recolectar datos iniciales

Los datos que se estarán utilizado como base para este proyecto son los siguientes:

Tabla 4. 1 Atributos utilizados en el desarrollo del proyecto.
Obtenido de: Elaboración propia.

student's school	extra paid classes within the course subject (Math or Portuguese)
student's sex	extra-curricular activities
student's age	attended nursery school
student's home address type	wants to take higher education
family size	Internet access at home
parent's cohabitation status	with a romantic relationship
mother's education	quality of family relationships
father's education	free time after school
mother's job	going out with friends
father's job	workday alcohol consumption
reason to choose this school	weekend alcohol consumption
student's guardian	health - current health status
home to school travel time	absences - number of school absences
weekly study time	first period grade
number of past class failures	second period grade
extra educational support	final grade
family educational support	

Dichos datos fueron obtenidos y recolectados de un proyecto enfocado en el desarrollo de algoritmos de regresión simple para predecir la calificación final del año.

Donde G3 será el dato de salida y el resto de las columnas serán las entradas principales.

4.2.2 Descripción de los datos

Las características que se usan en el proyecto son un total de 32 atributos o columnas y 392 registros u observaciones.

Cada uno de los datos cuenta con un acrónimo para un uso más óptimo como se observa en la tabla 4. 2.

Tabla 4. 2 Tabla de Acrónimos de atributos utilizados en el proyecto
Obtenido de: Elaboración propia.

Acrónimos de atributos utilizados en el proyecto	
school – escuela del estudiante	paid – pago de clases extras (matemáticas o portugués)
sex – sexo del estudiante.	activities – actividades extra curriculares
age – edad del estudiante.	nursery – estudiantes foráneos
address – tipo de hogar del estudiante.	higher – quiere estudiar una licenciatura o superior
famsize – tamaño de la familia.	internet – cuenta con internet en casa
Pstatus – estado de la familia.	romantic – está en una relación
Medu – educación de la madre.	famrel – calidad de la relación en la familia
Fedu – educación del padre.	freetime – tiempo libre después de la escuela
Mjob – trabajo de la madre.	goout – salida con amigos
Fjob – trabajo del padre.	Dalc – consumo de alcohol en fin de semana
reason – razón de elección de la escuela.	Walc – consumo de alcohol entre semana
guardian – encargado del alumno.	health – estado de salud
traveltime – tiempo libre del alumno.	absences – cantidad de faltas
studytime – tiempo de estudio del estudiante.	G1 – primer período de calificaciones
failures – cantidad de clases falladas.	G2 – Segundo periodo de calificaciones
schoolsup – soporte educativo adicional.	G3 – calificaciones finales
famsup – soporte familiar en la educación.	

4.2.3 Exploración de los datos

Dentro de esta etapa se comenzó con el análisis de los datos del *dataset* público con el cual se realiza el proyecto, aquí se utilizan técnicas de estadísticas básicas para profundizar en los datos.

Se realizó un análisis descriptivo en donde se muestra la frecuencia relativa, acumulada y relativa de los datos.

En la tabla 4.3 se muestra el análisis del *dataset* que se utiliza dentro del proyecto y en la cual se estan analizando los rangos de las calificaciones, su frecuencia entre otras cosas.

Tabla 4. 3 Tabla de frecuencia con data set público
Obtenido de: Elaboración propia

Tabla de frecuencia de calificaciones finales				
Rango de calificaciones	f1= Frecuencia Absoluta	F1=Frecuencia acumulada	hi = Frecuencia relativa	HI= Frecuencia relativa acumulada
0	38	38	0.09620253	0.09620253
7	32	70	0	0.09620253
10	116	186	0.29367089	0.38987342
20	209	395	0.52911392	0.91898734
Total	395		0.91898734	

Media	10.4151899
Desviación estándar	4.58144261

4.2.4 Verificar la calidad de los datos

En esta etapa se comienza a ver los datos desde Excel, aquí se pudo observar y determinar los datos, valores posibles y rangos.

1. Sex: F=Femenino y M=Masculino.
2. Age: rango de edad de 15 a 22.
3. Adress: U= urbano y R= rural.
4. Famsize: LE3= hasta 3 integrantes y GT3=mayor a 3 integrantes.
5. Pstatus: t= padres viviendo juntos y A= viviendo aparte.
6. Medu: 0= ninguna, 1= educación primaria (hasta 4° grado), 2= hasta 9°grado, 3= preparatoria y 4= educación superior.
7. Fedu: 0=ninguna, 1=educación primaria (hasta 4° grado), 2= hasta 9°grado, 3= preparatoria y 4 educación superior.
8. Mjob: maestro, salud, servicios civiles, ama de casa y otros.
9. Fjob: maestro, salud, servicios civiles, ama de casa y otros.
10. reason: escuela cerca de casa, la reputación de la escuela, curso, preferencia propia, otra.
11. guardian: padre, madre u otro.
12. traveltime: 1 - <15 min., 2 - 15 a 30 min., 3 - 30 min. a 1 hora, o 4 -> 1 hora.
13. studytime: 1 - <2 horas, 2 - 2 a 5 horas, 3 - 5 a 10 horas, o 4 -> 10 horas.

14. Failures: n sí 1 <= n <3, else 4.
15. schoolsup: sí o no.
16. Famsup apoyo educativo por parte de la familia: si o no.
17. Paid: sí o no.
18. activities: sí o no
19. Nursery: sí o no.
20. Internet: sí o no.
21. romantic: sí o no.
22. Famrel: 1 - Muy malo a 5 – Excelente.
23. FreeTime: 1 - muy bajo a 5 - muy alto.
24. Goout: 1 - muy bajo a 5 - muy alto.
25. Dalc – Wo: 1 - muy bajo a 5 - muy alto.
26. Walc: 1 - muy bajo a 5 - muy alto.
27. Health: 1 - muy bajo a 5 - muy alto.
28. absences: de 0 a 93 ausencias de clase.

Dentro de los análisis tenemos una serie de gráficas de barras en donde podemos observar la frecuencia de los datos (ve figura 4.1).

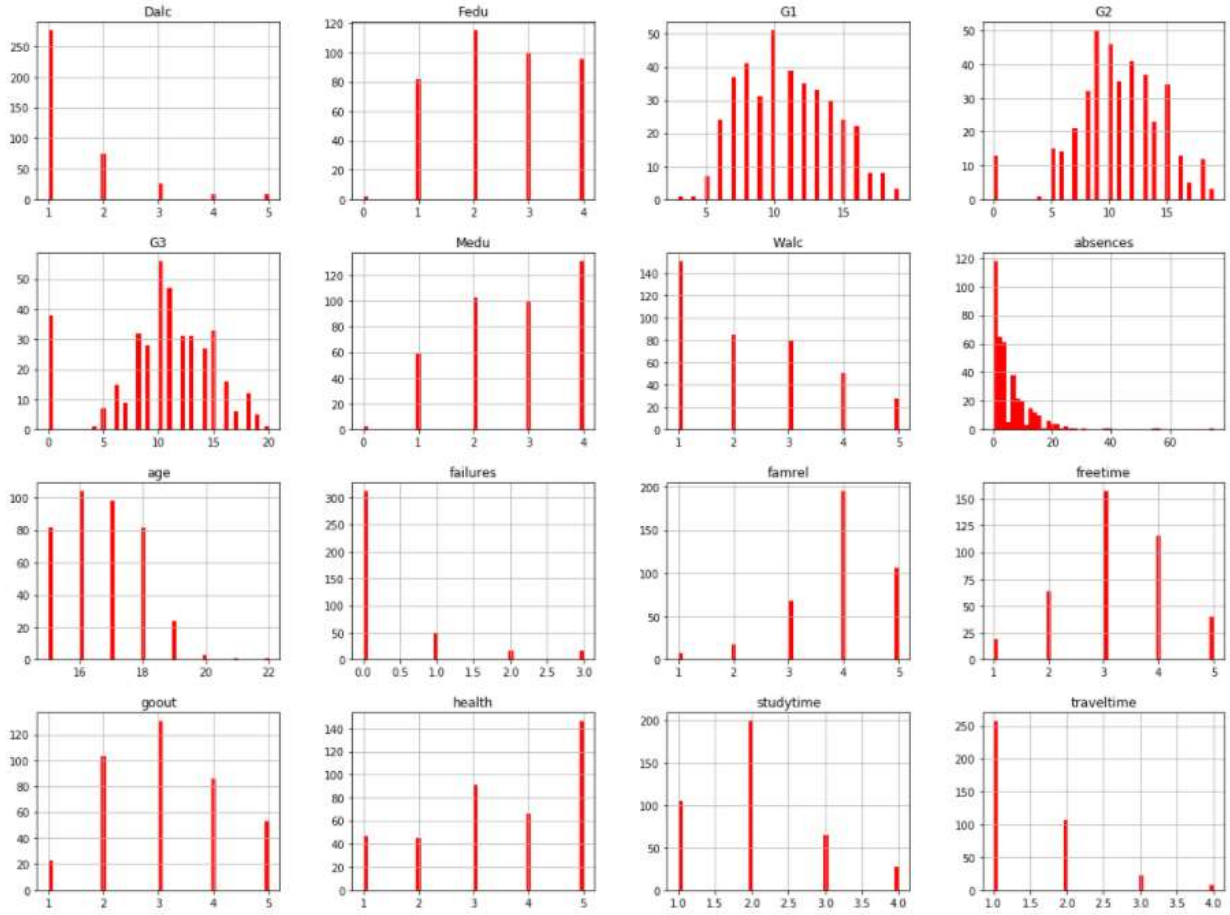


Figura 4. 1 Gráficas de frecuencia obtenidas con estadísticas descriptivas
Obtenida de: Martínez Bachmann, J. (2017).

4.3 Preparación de los datos

Dentro de esta etapa de la metodología se realizó la adaptación de los datos a las técnicas de minería de datos en donde veremos las relaciones entre las variables, etc.

4.3.1 Seleccionar los datos

Primero se seleccionó un subconjunto de datos los cuales servirán a realizar las siguientes sub fases de la metodología, los datos seleccionados son los siguientes: *Medu, failures, Fedu, age, sex, address, famsize, Pstatus, Mjob, Fjob, reason, guardian, traveltime, studytime y absences*. Estos porque fueron algunos de los atributos en donde se mostró mayor correlación tanto positiva como negativa.

4.3.2 Limpiar datos

Para esta fase se comenzó con una normalización dentro del archivo .csv, para esto se puso el *dataset* el cual contiene todos los datos, dentro del programa Python generó, el cual cuenta con todas las tablas y datos utilizados para el desarrollo de este proyecto.

Primero se comenzó con una cantidad de 21 atributos los cuales fueron examinados para explorar sus tipos de datos ver figura 4.2 .
También para el proceso de limpieza, se normalizaron los campos dentro del dataset; dicho proceso consistió en el chequeo de todas las columnas y atributos observando que no tuvieran espacios vacíos, un tipo de dato incorrecto o que se encontraran fuera de rango.

Pero al momento de examinar esta parte se llegó a la conclusión de que no se necesitaba realizar ningún cambio a los datos ya que estaban bien optimizados (ver figura 4.3).

sex	text	YES
age	int	YES
address	text	YES
famsize	text	YES
Pstatus	text	YES
Medu	int	YES
Fedu	int	YES
Mjob	text	YES
Fjob	text	YES
reason	text	YES
guardian	text	YES
traveltime	int	YES
studytime	int	YES
failures	int	YES

Figura 4. 2 Tipo de datos de los atributos del proyecto.
Obtenido de: Elaboración Propia.

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	studytime	failures
GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2	0
GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2	0
GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2	3
GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3	0
GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2	0
GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2	0
GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2	0
GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2	0
GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2	0
GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2	0
GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1	2	0
GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3	3	0
GP	M	15	U	LE3	T	4	4	health	services	course	father	1	1	0
GP	M	15	U	GT3	T	4	3	teacher	other	course	mother	2	2	0
GP	M	15	U	GT3	A	2	2	other	other	home	other	1	3	0
GP	F	16	U	GT3	T	4	4	health	other	home	mother	1	1	0
GP	F	16	U	GT3	T	4	4	services	services	reputation	mother	1	3	0
GP	F	16	U	GT3	T	3	3	other	other	reputation	mother	3	2	0
GP	M	17	U	GT3	T	3	2	services	services	course	mother	1	1	3
GP	M	16	U	LE3	T	4	3	health	other	home	father	1	1	0

Figura 4. 3 Atributos normalizados.
Obtenidos de: Elaboración propia.

4.3.3 Construcción, Integración y Formateo de los datos.

Dentro de esta fase se revisó si se tenía que eliminar, ajustar algún valor dentro del *dataset*, ya sea que se eliminen algunas comas, tabulaciones o caracteres.

Para esto se tomó el archivo *.cvs* generado con el modelo de clustering con el método K-Means el cual consiste en la creación de grupos basados en las calificaciones finales de los estudiantes dividiéndolos en tres grupos distintos (alumnos con buenas, medias y malas calificaciones).

Una vez terminado el análisis se continuó con la creación de nuevas estructuras o campos a partir de atributos existentes, en este caso se seleccionó: *Medu, failures, absences, G3*, entre otros. Dichos atributos mencionados anteriormente fueron implementados dentro del modelo de clustering y el cual creará una nueva columna en la que se verán reflejados los clusterings que se generaron con el modelo, además de que se verá a cual calificación está asociado dicho clúster (ve figura 4.4).

W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
internet	romantic	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	KMeans_Cluster
no	no	4	3	4	1	1	3	6	5	6	6	0
yes	no	5	3	3	1	1	3	4	5	5	6	0
yes	no	4	3	2	2	3	3	10	7	8	10	2
yes	yes	3	2	2	1	1	5	2	15	14	15	2
no	no	4	3	2	1	2	5	4	6	10	10	2
yes	no	5	4	2	1	2	5	10	15	15	15	2
yes	no	4	4	4	1	1	3	0	12	12	11	0
no	no	4	1	4	1	1	1	6	6	5	6	0
yes	no	4	2	2	1	1	1	0	16	18	19	2
yes	no	5	5	1	1	1	5	0	14	15	15	2
yes	no	3	3	3	1	2	2	0	10	8	9	2
yes	no	5	2	2	1	1	4	4	10	12	12	1
yes	no	4	3	3	1	3	5	2	14	14	14	2
yes	no	5	4	3	1	2	3	2	10	10	11	2
yes	yes	4	5	2	1	1	3	0	14	16	16	0
yes	no	4	4	4	1	2	2	4	14	14	14	0
yes	no	3	2	3	1	2	2	6	13	14	14	2
no	no	5	3	2	1	1	4	4	8	10	10	1
yes	no	5	5	5	2	4	5	16	6	5	5	1
yes	no	3	1	3	1	3	5	4	8	10	10	2
yes	no	4	4	1	1	1	1	0	13	14	15	0
yes	no	5	4	2	1	1	5	0	12	15	15	2
yes	no	4	5	1	1	3	5	2	15	15	16	1
yes	no	5	4	4	2	4	5	0	13	13	12	1
yes	no	4	3	2	1	1	5	2	10	9	8	2
yes	no	1	2	2	1	3	5	14	6	9	8	2
yes	no	4	2	2	1	2	5	2	12	12	11	2
yes	no	2	2	4	2	4	1	4	15	16	15	2
yes	no	5	3	3	1	1	5	4	11	11	11	1
yes	yes	4	4	5	5	5	5	16	10	12	11	2
yes	no	5	4	2	3	4	5	0	9	11	12	2
yes	no	4	3	1	1	1	5	0	17	16	17	1
yes	yes	4	5	2	1	1	5	0	17	16	16	1
yes	no	5	3	2	1	1	2	0	8	10	12	1
yes	no	5	4	3	1	1	5	0	12	14	15	2
no	no	3	5	1	1	1	5	0	8	7	6	1
yes	no	5	4	3	1	1	4	2	15	16	18	1

Figura 4. 4 Datos obtenidos con la creación del modelo de agrupamiento.
Obtenido de: Elaboración propia.

4.4 Modelado

A continuación, se describen las técnicas utilizadas, la generación, construcción y evaluación de los modelos que se desarrollaron para los modelos de LA.

4.4.1 Escoger la técnica de modelado

Lo primero que se realizó dentro de esta fase fue la selección del tipo de técnica para el desarrollo del objetivo principal que tenemos dentro del proyecto. En éste caso el problema a resolver es como predecir calificaciones, entonces se utilizarán: análisis de regresión simples, múltiples métodos de agrupamiento con el método K-Means y redes neuronales.

4.4.2 Generar el plan de prueba

Dentro de esta etapa se comienza con la construcción de las diferentes ventanas de la aplicación web en donde se muestran los modelos de LA.

Primero se inició con la creación de la página web con diferentes herramientas de desarrollo como los son: C# para la creación de una API la cual ayudará a la conexión de React con Python, React para la visualización por parte del cliente, JavaScript para procesos dentro de HTML, entre otras herramientas.

Luego se continua con el desarrollo de la ventana de estadísticas básicas en donde se visualizará toda la información y descripción de los datos que se estan utilizando dentro del proyecto, después se continua con la pestaña de modelos en donde se encuentra la ventana de regresión simple en donde se visualizaran gráficas de predicciones basadas en un solo atributo todo con la utilización del modelo que se desarrolló en Python, el cual hace la realización de las gráficas y las predicciones.

Posteriormente se continuará con la ventana de regresión múltiple en la cual se verán predicciones de las calificaciones basadas en tres atributos distintos obtenidos de datos proporcionados por el alumno, todo con ayuda del programa desarrollado con Python con el cual obtiene la fórmula para la predicción y las gráficas.

Enseguida se continua con la ventana de clustering en la cual se tiene una serie de graficas obtenidas con el programa Python y las cuales serán mostradas dentro de la aplicación web, dichas gráficas muestran diferentes clústers basándose en los atributos que posee un alumno para clasificarlos en grupos.

Por último, se tiene la ventana de redes neuronales y de recursos de ayuda en donde se mostrará una serie de graficas obtenidas con el modelo de redes neuronales y una tabla en la que se tendrá acceso a diferentes artículos de ayuda en los cuales se hablan de técnicas para mejorar las técnicas de enseñanza y técnicas de estudio.

4.4.3 Construir el modelo

Dentro de esta tarea se comienza con la preparación de los modelos a los cuales se les implementará la técnica seleccionada. En este caso se verán dentro de los modelos de regresión simple, múltiple, de agrupamiento con el método de K-Means y redes neuronales.

4.4.3.1 Construcción del modelo de regresión múltiple

Los elementos tomados en cuenta para la construcción del modelo de regresión múltiple son:

Medu, absences y failures dichos atributos fueron seleccionados porque se observó que contaban con una correlación positiva y negativa con el atributo G3 porque es el atributo del cual queremos sacar la predicción (ver tabla 4.4).

Dicho modelo se creó con ayuda de librerías como lo son: *numpy, pandas, matplotlib, sklearn lineal regression y sklearn merinus*, las cuales se seleccionan porque el código estará desarrollado en Python y el cual utiliza los atributos seleccionados para la predicción.

Tabla 4. 4 Atributos y sus correlaciones.

Obtenido de: Elaboración propia.

Atributo	Correlación
G3	1.000000
Medu	0.217147
b_higher_education	0.182465
Fedu	0.152457
b_paidxtraclasses	0.101996
b_internet	0.098483
studytime	0.097820
b_Pstatus	0.058009
b_nursery	0.051568
famrel	0.051363
absences	0.034247
b_xtraactivities	0.016100
freetime	0.011307
b_reason	-0.028738
b_famsup	-0.039157
b_school	-0.045017
Walc	-0.051939
b_guardian	-0.054193
Dalc	-0.054660

health	-0.061335
b_famsize	-0.081407
b_schoolsup	-0.082788
b_address	-0.105756
traveltime	-0.117142
b_romantic	-0.129970
goout	-0.132791
age	-0.161579
failures	-0.360415

4.4.3.2 Construcción del modelo de agrupamiento (Clustering con K-Means)

Dentro del modelo de agrupamiento la selección de atributos es distinta ya que este modelo consiste en la exclusión de elementos que no le sean de utilidad para la selección final de los grupos.

Los elementos excluidos dentro de este modelo son:

G1, G2, G3, address, Pstatus, reason, famsup, school, nursery, goout, sex, famsize, paid, activities, higher, internet, romantic, guardian, schoolsup, Mjob, Fjob, famrel. Se seleccionaron dichos atributos porque se consideró que no eran de vital importancia.

Una vez seleccionados los elementos que serán excluidos se comienza con el desarrollo el cual será con el lenguaje de programación Python el cual utiliza librerías iguales a las de regresión múltiple con la diferencia de que aquí usamos *sklearn cluster*.

4.4.3.3 Construcción del modelo de regresión simple

El modelo de regresión simple es algo más sencillo, ya que dentro del desarrollo solo se utiliza una variable para realizar predicciones y posteriormente mostrarlas dentro de una gráfica que muestra las predicciones de los alumnos con posibilidades de aprobar la materia, basándose en el atributo con el cual se está generando la predicción.

4.4.4 Evaluar el modelo

Para algunos de los modelos mencionados anteriormente se realizaron diferentes pruebas, en donde se logra observar cómo se comportan cuando se utilizan diferentes atributos.

4.4.1.1 Evaluación del modelo de regresión múltiple

Para el modelo de regresión múltiple se realizó una tabla comparativa en la cual se utilizan diversos atributos y se calculan sus errores y coeficientes de determinación y de ese modo determinar cuál es el más óptimo (ver tabla 4.5).

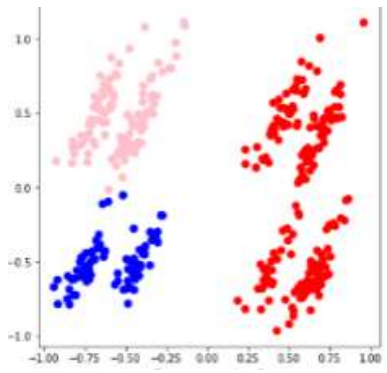
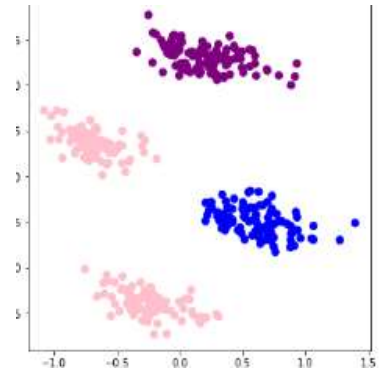
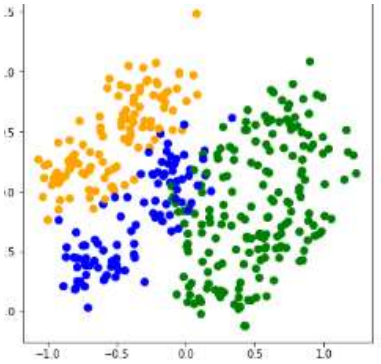
Tabla 4. 5 Tabla comparativa del modelo regresión múltiple para coeficientes y errores obtenidos con diferentes atributos.
Obtenido de: Elaboración propia.

Atributos	Error	Error cuadrado	Coeficiente de determinación
Educación de la madre y padre, faltas y ausencias.	4.427711745225454	0.15877225468411037	0.15877225468411037
Educación de la madre, tiempo libre y asistencias.	4.466023278619253	0.04733910819444753	0.05389213253944958
Tiempo de estudio, asistencias y faltas.	4.256335977266388	0.1346970660406004	0.1346970660406004
Tiempo de estudio, tiempo afuera, tomar alcohol entre semana.	4.5165574991249064	0.02565792022452129	0.02565792022452129

4.4.1.2 Evaluación modelo de agrupación (clustering)

Para el modelo de clustering también se realizó una tabla comparativa en la cual se observa atravez de graficas el comportamiento del modelo cuando utiliza diferentes combinaciones de atributos. (ver tabla 4.6).

Tabla 4.6 Tabla comparativa de clusterings obtenidos con combinaciones de diferentes atributos
Obtenida de: Elaboración propia.

Atributos incluidos	**Visualización del clustering**	**Grupos obtenidos**
Sexo, edad, tamaño de la familia, pago extra, actividades complementarias y situación sentimental.		Color Azul: Alumnos con calificicaciones altas (10 a 15). Color Rosa: Alumnos con calificaciones medias (7 a 9). Color Rojo: Alumnos con calificaciones bajas (0 a 6).
Sexo, edad, tamaño de la familia, pago extra, actividades complementarias, situación sentimental, educación de la madre y padre, soporte escolar, guardianes, internet, estado de salud y relación con la familia.		Color Morado: Alumnos con calificicaciones altas (10 a 15). Color Azul: Alumnos con calificaciones medias (7 a 9). Color Rosa: Alumnos con calificaciones bajas (0 a 6).
Sexo, edad, tamaño de la familia, pago extra, actividades complementarias, situación sentimental, educación de la madre y padre, soporte escolar, guardianes, internet, estado de salud, actividades extras y relación con la familia.		Color Verde: Alumnos con calificicaciones altas (10 a 15). Color Amarillo: Alumnos con calificaciones medias (7 a 9). Color Azul: Alumnos con calificaciones bajas (0 a 6).

4.5 Evaluación

A continuación, se describen las evaluaciones, revisiones y determinaciones de los modelos que se estan implementando dentro del proyecto.

4.5.1 Evaluar los resultados

Dentro del subtema de evaluación que se vio anteriormente se observó la exactitud y la generalidad de los modelos generados analizando su comportamiento al enfrentarse a la implementación y combinación de distintos atributos para la obtención de gráficas que muestran grupos, coeficientes y errores.

Se determina que las evaluaciones de los modelos aún tienen mucha oportunidad de mejora ya que en este proyecto se implementan datos obtenidos de un dataset público y se estima que para un proyecto próximo se implemente con datos obtenidos por parte de la institución académica logrando de ese modo un mayor incremento de coeficiente de determinación y clustering muchísimo más exactos.

4.5.2 Revisar el proceso

Dentro de esta etapa se detectó con posibilidad de mejora:

- Implementación de datos escolares obtenidos por el tecnológico.
- Mejora dentro del coeficiente de determinación del modelo de regresión múltiple.
- Mejora en el modelo de agrupamiento para que se dividan mucho mejor.

4.5.3 Determinar los próximos pasos

Los datos que se generaron durante la creación de los modelos de LA generaron resultados satisfactorios aptos para la implementación dentro de la página web que se está desarrollando, así que se continuará con el desarrollo de las tareas consecuentes de la metodología seleccionada.

4.6 Despliegue o implementación

A continuación, se describen la planeación de la implementación y la producción del informe final de los modelos que se desarrollaron para este proyecto.

4.6.1 Planear la implementación

Una vez obtenidos los resultados ya mencionados en las tareas anteriores, se comienza con el desarrollo de la estrategia de implementación.

Lo primero que se realizará con los resultados obtenidos, es la creación de una aplicación web en donde se podrán ver las gráficas de agrupamiento obtenidas con el modelo de clustering, al igual que un apartado en donde se tendrán las predicciones de calificaciones con el modelo de regresión múltiple y apartados en donde se tendrá acceso a las gráficas del modelo de regresión simple, las descripciones de los datos que se utilizan, las predicciones con redes neuronales y las recomendaciones de artículos para mejorar los estudios o tus tácticas de enseñanza.

4.6.2 Producir el informe final

Como reporte final se obtiene una gran experiencia ya que con cada uno de los modelos se obtuvieron nuevos conocimientos y experiencias; desde el funcionamiento de cada uno de los modelos, los cálculos que deben de realizar para el agrupamiento o la predicción, hasta las historias de logros al momento de usar LA.

4.7 Modelos de Learning Analytics

Una vez terminado todo el proceso dentro de la metodología utilizada se comenzó con el desarrollo de los modelos de LA.

4.7.1 Modelos de regresión Múltiple

Primero se comenzó con el estudio de cómo realizar un modelo de regresión múltiple en el cual se utilizan más de 2 atributos para la estimación y comprensión de las relaciones entre las variables,

enfocándose en una variable dependiente y una serie de otras variables combatientes, logrando de ese modo la predicción y el pronóstico (Apd, R., 2020).

En la figura 4.5 se observa el código desarrollado para el modelo de regresión múltiple en donde se utilizan diferentes librerías como *pandas, numpy, seaborn y matplotlib*. Además de las líneas de comando necesarias que ayudan a la realización de la regresión y evaluación de errores y coeficientes.

El modelo de regresión múltiple se probó con diferentes atributos para ver los diferentes resultados y coeficientes de determinación obtenidos para de ese modo elegir el mejor, algunas de esas pruebas pueden ser visualizadas en la tabla 4.5.

Si se quiere ver más a fondo el código que se desarrolló para el modelo vaya al anexo 2.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

sns.set_style('darkgrid')

datos= pd.read_csv('./input/student-mat.csv')

#datos que vamos a estar checando
nuevo= datos[['studytime','absences','failures','G3']]

#nombrar gráfica, colores dependientes de la edad y el histograma
g= sns.pairplot(nuevo,hue='failures', diag_kind='hist')

#creamos ciclo for para una mejor visión de las graficas
for ax in g.axes.flat:
    plt.setp(ax.get_xticklabels(),rotation=45)

datos=datos.replace(np.nan,'0')
Medu=datos['studytime'].values
Fedu=datos['absences'].values
age=datos['failures'].values
G3=datos['G3'].values

#generamos el arreglo que contendrá todas las caract. de la x
#ahora ponemos la traspuesta para que sea de columna a renglón
X=np.array([Medu,Fedu,age]).T
Y=np.array(G3)

#comenzamos con la parte del modelo de regresión
reg=LinearRegression()
#hacemos el ajuste del modelo
reg=reg.fit(X,Y)
#hacemos la predicción
y_pred=reg.predict(X)
#sacamos el error de la predicción entre Y y la predicción que se está sacando
error=np.sqrt(mean_squared_error(Y,y_pred))
r2=reg.score(X,Y)

# Obtener coeficiente de determinación
r_sq = reg.score(X,Y)
print('coefficient of determination:', r_sq)

print('El error es: ', error)

print('El error de r^2 es: ', r2)
```

Figura 4. 5 Código desarrollado para el modelo de regresión múltiple.
Obtenido de: Elaboración propia.

4.7.2 Modelos de Agrupamiento (Clustering)

El segundo modelo que se desarrolló fue el modelo de agrupamiento también conocido como clustering más específicamente con el método de K-Means, el cual consiste en el agrupamiento de elementos que poseen una característica en común. En este caso es el agrupamiento de los estudiantes que poseen un cierto grupo de características para posteriormente separarlos en grupos como: alumnos con buen rendimiento académico, rendimiento medio y alumnos en posible riesgo académico (ver figura 4.6).

Si quiere ver más a detalle el código que se desarrolló para este modelo vaya a ver el anexo 3.

Clustering Metodo K-Means con codo

```
import numpy as np #para calculos cientificos
import pandas as pd #para el analisis de datos
import matplotlib.pyplot as plt # para creacion de graficas
from sklearn.cluster import KMeans # para importacion del metodo
```

```
df=pd.read_csv('./input/student-mat.csv', engine='python')
```

```
df.info() #vemos que es lo que contiene el objeto datos
```

```
df.head() #vemos las filas de los datos
```

```
#Linea que se usa para eliminar o no tomar en cuenta un elemento o columna
#df_variables=df.drop(['school'], axis=1)
df_variables=df.drop(['G1', 'G2', 'G3', 'address', 'Pstatus', 'reason', 'famsup', 'school', 'nursery', 'goout', 'sex'], axis=1)
```

```
#Aqui podremos observar todo los estatisticos maximos, minimos, cuartiles, primedio
#desviasion estandar, etc.
df_variables.describe()
```

```
#Utilizamos el mismo metodo que se utiliza en el data set para convertir
#todo a numeros enteros.


# 0 stands for F and 1 stands for M. [F=Femenino, M=Masculino]
# Here we will convert all the binary columns to integers.
#df_variables['b_sex'] = df_variables['sex'].apply(lambda x: 0 if x == 'F' else 1)
#df_variables['b_sex'].value_counts()

# 0 stands for U and 1 stands for R. [U=Urban, R=Rural]
# Here we will convert all the binary columns to integers.
#df_variables['b_address'] = df_variables['address'].apply(lambda x: 0 if x == 'U' else 1)
#df_variables['b_address'].value_counts()

# Interestingly there are more students in families that are greater than 3.
# Could it be possible that all family members are in the same school? This might be a reason why it is higher.
# LE3 = Less than 3. [0], GE3 = Greater than 3.[1]
df_variables['b_famsize'] = df_variables['famsize'].apply(lambda x: 0 if x == 'LE3' else 1)
df_variables['b_famsize'].value_counts()

# T = Parents are living together [0], A = Parents living apart. [1]
#df_variables['b_Pstatus'] = df_variables['Pstatus'].apply(lambda x: 0 if x == 'T' else 1)
#df_variables['b_Pstatus'].value_counts()

# 0 = no and 1 = yes
#df_variables['b_famsup'] = df_variables['famsup'].apply(lambda x: 0 if x == 'no' else 1)
#df_variables['b_famsup'].value_counts()


# 0 = no and 1 = yes
# This is an interesting column when it comes to having a positive effect on G3.
df_variables['b_paidxtraclasses'] = df_variables['paid'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_paidxtraclasses'].value_counts()

# 0 = no and 1 = yes
df_variables['b_xtraactivities'] = df_variables['activities'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_xtraactivities'].value_counts()

# 0 = no and 1 = yes
# It has a high correlation however, we only have 20 students that are not interested in having a high education and
# thus this column should not be taken into consideration.
df_variables['b_higher_education'] = df_variables['higher'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_higher_education'].value_counts()
```

```
# continue with the analisis.
df_variables['b_internet'] = df_variables['internet'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_internet'].value_counts()

# Interestingly when people are not in a romantic relationship they tend to get better grades.
df_variables['b_romantic'] = df_variables['romantic'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_romantic'].value_counts()

#df_variables['b_nursery'] = df_variables['nursery'].apply(lambda x: 0 if x == 'no' else 1)
#df_variables['b_nursery'].value_counts()

df_variables['b_guardian'] = df_variables['guardian'].apply(lambda x: 0 if x == 'mother' else (1 if x=='father' else 2))
df_variables['b_guardian'].value_counts()

# Does not have any effect on G3. Low correlation.
#df_variables['b_reason'] = df_variables['reason'].apply(lambda x: 0 if x == 'home' else (1 if x=='reputation' else (3 if x=='co
#df_variables['b_reason'].value_counts()

# Does not have any effect on G3. Low correlation.
#df_variables['b_school'] = df_variables['school'].apply(lambda x: 0 if x == 'GP' else 1)
```

```
# Does not have any effect on G3. Low correlation.
#df_variables['b_school'] = df_variables['school'].apply(lambda x: 0 if x == 'GP' else 1)
#df_variables['b_school'].value_counts()

df_variables['b_schoolsup'] = df_variables['schoolsup'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_schoolsup'].value_counts()

#variable Mjob
df_variables['b_Mjob'] = df_variables['Mjob'].apply(lambda x: 0 if x == 'nominal' else (1 if x=='health' else (2 if x=='service
df_variables['b_Mjob'].value_counts()

#variable Fjob
df_variables['b_Fjob'] = df_variables['Fjob'].apply(lambda x: 0 if x == 'nominal' else (1 if x=='health' else (2 if x=='service
df_variables['b_Fjob'].value_counts()

df_variables_new=df_variables.drop(columns=['famsize','paid','activities','higher','internet',
                                            'romantic','guardian','schoolsup','Mjob','Fjob','famrel'])
```

```
#normalizamos los valores para que se pongan entre los mismo rangos
#ya que los valores estan muy distintos
df_norm=(df_variables_new-df_variables_new.min())/(df_variables_new.max()-df_variables_new.min())
df_norm
```

```
df_norm.describe()
```

```
#implementaremos el metdo codo de jambu
#crea difef tipos de clustering para ver que tan simirales son los vecinos
#e irlos mostrando o plasmandolos dentro de una grafica.

#wcss es la suma de los cuadrados de cada grupo


arreglowcss = []#variable para almacenar

for i in range (1,11):#loop para crear agrupaciones se pone hasta cual numero quieres +1
    kmeans = KMeans(n_clusters = i, max_iter=300)
    kmeans.fit(df_norm) #aplicamos K/means a la base de datos
    arreglowcss.append(kmeans.inertia_)
```

```
plt.plot(range(1,11), arreglowcss)
plt.title('codo de Jambu')
plt.xlabel('Numero de clusters')
plt.ylabel('WCSS')#indicador de que tan similares son los individuos dentro de los clusters
plt.show()
```

```
#aplicamos el metodo Kmeans a la BD

clustering = KMeans(n_clusters = 3, max_iter = 300)#creamos el modelo
clustering.fit(df_norm)#aplicamos el modelo a la BD
```

```
KMeans(n_clusters=3)
```

```
#agramos la clasificacion al archivo original

df['KMeans_Clusters'] = clustering.labels_ #los resultados se guardan en label_ dentro del modelo
df.head()
```

```
#visualizacion de los clustering que se fomaron
#utilizando graficos con analisis de componentes principales PCA

from sklearn.decomposition import PCA

pca = PCA(n_components=2)#modelo de 2 dimensiones
pca_df = pca.fit_transform(df_norm)#obtenemos los dos componentes principales
pca_df_data = pd.DataFrame(data = pca_df, columns = ['Componente_1', 'Componente_2']) #Creamos dataframe que contega los element
pca_nombres_df = pd.concat([pca_df_data, df[['KMeans_Clusters']]], axis=1) #agegamos la columna del clustering

pca_nombres_df

pca_nombres_df.to_csv('C:/Users/Admin/Documents/PrediccionDeCalificcionesSecundaria/clusters_creados/MetodoPCA1.csv')
```

```
#coloreamos los clustering para diferenciar mejor

fig = plt.figure(figsize = (6,6)) #tamano de la figura

ax = fig.add_subplot(1,1,1)  #creamos solo 1 grafico
ax.set_xlabel('Componente_1', fontsize = 15)
ax.set_ylabel('Componente_2', fontsize = 15)
ax.set_title('Componentes Principales', fontsize = 20)

color_theme = np.array(["blue", "pink", "purple"])
ax.scatter(x = pca_nombres_df.Componente_1, y = pca_nombres_df.Componente_2,
           c=color_theme[pca_nombres_df.KMeans_Clusters], s = 50)
plt.show()
```

Figura 4. 6 Código desarrollado para el modelo de Clustering.
Obtenido de: Elaboración propia.

4.7.3 Modelos de Regresión Simple

El tercer modelo desarrollado es el modelo de regresión simple, el cual consiste en la estimación y comprensión de las relaciones entre las variables o variable utilizada (Apd, R., 2020).

Aunque dentro del proyecto original ya existe un modelo de regresión simple dicho modelo no cuenta con una visualización de los resultados obtenidos.

En la figura 4.7 se observa las líneas de código desarrolladas para el modelo de regresión simple en ella se observa diferentes librerías como *numpy*, *pandas*, etc. Así como también los procesos de regresión y la graficación de los resultados.

Para verlo más a fondo vaya a anexo 1.

```
#Regresion Lineal Simple
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
#import pandas.util.testing as tm
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

#Cargamos el conjunto de datos
dataset = pd.read_csv('./input/student-mat.csv')
```

```
#models for 50% train and 50% test

X = np.array(dataset['Medu']).reshape(-1, 1)
y = np.array(dataset['G3']).reshape(-1, 1)
# Separating the data into independent and dependent variables
# Converting each dataframe into a numpy array
# since each dataframe contains only one column
#df_set.dropna(inplace = True)
# Dropping any rows with Nan values
X_train, X_test, y_train, y_test = train_test_split(X, y,train_size = 0.5,test_size = 0.5,random_state=0)
# Splitting the data into training and testing data
regr = LinearRegression()
regr.fit(X_train, y_train)
X_train.shape
```

```
(197, 1)
```

```
#predicting the test result and visualizing the test result
y_pred=regr.predict(X_test)
y_pred
plt.scatter(X_test,y_test,color='orange')
plt.plot(X_test,regr.predict(X_test),color='black')
plt.title('failures  vs G3(Test Data 50%)')
plt.xlabel('failures ')
plt.ylabel('G3')
plt.show()
```

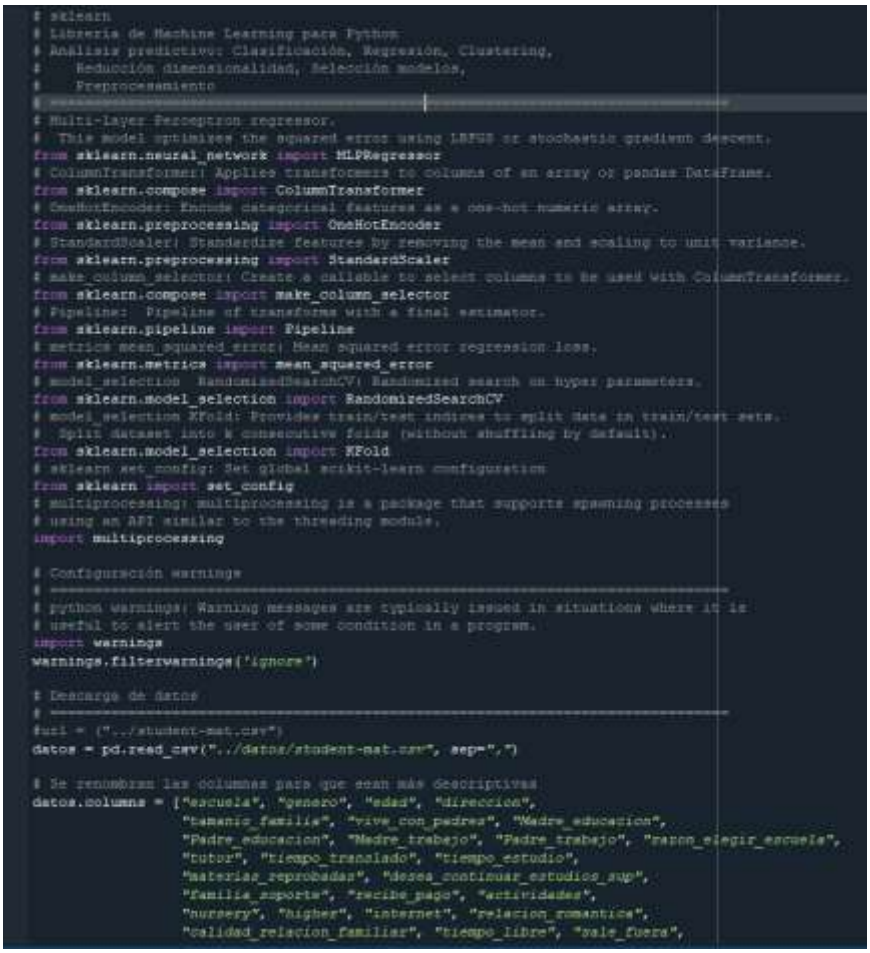
Figura 4. 7 Código desarrollado para el modelo de regresión simple.
Obtenido de: Elaboración propia.

4.7.4 Modelo de Redes Neuronales

El cuarto modelo desarrollado fue el modelo de redes neuronales, uno de los métodos de “Machine Learning”, en donde se utilizan algoritmos programados los cuales reciben y analizan datos de entrada para predecir los valores de salida dentro de un rango aceptable.

Este modelo fue desarrollado con ayuda de mi asesor a cargo y utiliza una gran serie de librerías para poder llevar el proceso de predicción al igual que una serie de razonamiento y atributos, así como se muestra en la figura 4.8.

En el anexo 4 se muestra todo el código desarrollado para este modelo.



```
# Distribución variable respuesta
# Histogramas de las variables G1, G2 y G3
# ==============================================================================
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G3', kde=True,ax=ax)
ax.set_title("Distribución G3")
ax.set_xlabel('G3');

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G2', kde=True,ax=ax)
ax.set_title("Distribución G2")
ax.set_xlabel('G2');

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G1', kde=True,ax=ax)
ax.set_title("Distribución G1")
ax.set_xlabel('G1');

# Gráfico de distribución para cada variable numérica
# Considerar la cantidad de variables numéricas para configurar correctamente los
# parámetros nrows y ncols, que definen las filas y columnas  para gráficas
# ==============================================================================
#fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(12, 7))

fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(12, 7))
axes = axes.flat
# Se obtienen las variables numéricas y se omiten G1, G2 y G3
columnas_numeric = datos.select_dtypes(include=['float','int64']).columns
columnas_numeric = columnas_numeric.drop('G1')
columnas_numeric = columnas_numeric.drop('G2')
columnas_numeric = columnas_numeric.drop('G3')

# Ciclo para obtener histograma de cada variable numérica
for i, colum in enumerate(columnas_numeric):
    sns.histplot(
        data    = datos,
        x       = colum,
        stat    = "count",
        kde     = True,
        color   =  "blue",
        #color   = (list(plt.rcParams['axes.prop_cycle'])*2)[i]["color"],
        line_kws= {'linewidth': 2},
        alpha   = 0.3,
        ax      = axes[i]
    )
    axes[i].set_title(colum, fontsize = 7, fontweight = "bold")
    axes[i].tick_params(labelsize = 6)
    axes[i].set_xlabel("")
    axes[i].set_ylabel("")


fig.tight_layout()
```

```
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough'
                )

# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline
pipe = Pipeline([('preprocessing', preprocessor),
                 ('modelo', MLPRegressor(solver = 'lbfgs',
                                         max_iter= 10000))])



# Espacio de búsqueda de cada hiperparámetro
# ==============================================================================
param_distributions = {
    'modelo__hidden_layer_sizes': [(10), (20), (10, 10)],
    'modelo__alpha': np.logspace(-3, 3, 10),
    'modelo__learning_rate_init': [0.001, 0.01],
}

# Búsqueda por validación cruzada
# ==============================================================================
grid = RandomizedSearchCV(
        estimator  = pipe,
        param_distributions = param_distributions,
        n_iter     = 10,
        scoring    = 'neg_mean_squared_error',
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = 5,
        verbose    = 0,
        random_state = 123,
        return_train_score = True
       )

grid.fit(X = X_train, y = y_train)

# Resultados del grid
# ==============================================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param.*|mean_t|std_t)')\
    .drop(columns = 'params')\
    .sort_values('mean_test_score', ascending = False)\
    .head(10)

# Resultados después de procesado
print("Resultados despues de proceso:")
print(resultados)
resultados.to_csv("resultados_proceso_redneuronal.csv")

# Error de test
# ==============================================================================
```

Figura 4. 8 Código desarrollado para el modelo de redes neuronales.
Obtenido de: Elaboración propia.

4.8 Desarrollo de la aplicación

Dentro de los requerimientos pensados para este proyecto fue la realización de una aplicación en la cual se logre visualizar los diferentes modelos desarrollados.

Por tal motivo se optó por el desarrollo de una aplicación web desarrollada con diferentes herramientas como lo son: JavaScript, C#, Python, HTML5, además React para el lado del cliente.

Lo primero que se realizó para la aplicación web fue la realización de una API con la cual se podrán hacer las peticiones como cuando se agrega un usuario, inicio de sesión, cálculos para el modelo de regresión múltiple, cálculos para la gráfica de clustering, etc.

En la figura 4.9 se puede observar una parte del código desarrollado dentro de la API para los cálculos y las autentificaciones que se generan dentro de la aplicación web. También en la figura 4.10 se muestran todos los archivos que se generaron para que la API se implementara y funcionara de manera correcta.

```
[Route("/api/[controller]/Clustering")]
[HttpGet]
public IActionResult ClusterMethod(int numClusters)
{
    Console.WriteLine(RunPythonScript(numClusters));
    return Ok(RunPythonScript(numClusters));
}

[Route("/api/[controller]/FirstModelSubmit")]
[HttpPost]
public IActionResult FirstModelSubmit(FirstModel fs)
{
    const double x1 = 15.813793035251104;
    const double x2 = 0.78151842;
    const double x3 = 0.06722984;
    const double x4 = -0.46158517;
    double result = 0;

    try
    {
        result = (x1 + (fs.MotherEducation * x2) + (fs.FatherEducation * x3) + (fs.StudentAge * x4));
        return Ok(result);
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return Forbid();
    }
}

[Route("/api/[controller]/SecondModelSubmit")]
[HttpPost]
public IActionResult SecondModelSubmit(SecondModel sm)
{
    const double x1 = 15.813793035251104;
    const double x2 = 0.78151842;
    const double x3 = 0.00748238;
    const double x4 = 0.02474765;
    double result = 0;

    try
    {
        result = (x1 + (sm.MotherEducation * x2) + (sm.StudentAbsences * x3) + (sm.StudentFreeTime * x4));
        return Ok(result);
```

Figura 4. 9 Código desarrollado para la API.
Obtenido de: Elaboración propia.

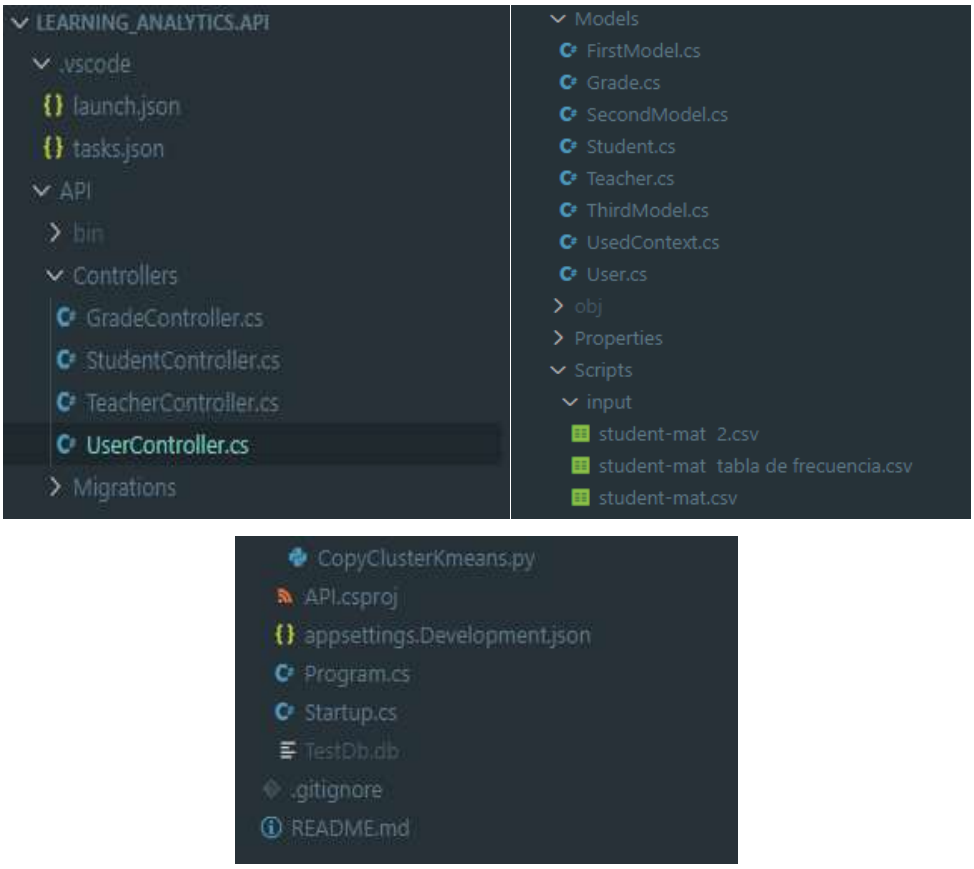


Figura 4. 10 Estructura del proyecto web.
Obtenido de: Elaboración propia.

4.8.1 Desarrollo ventana regresión simple

Dentro de esta pestaña se realizó la creación de un *tabbed* también conocido como pestañas, este es un componente dentro de *react bootstrap* el cual es un interfaz en la cual dentro de cada pestaña se puede mostrar información diferente.

Este tipo de diseño fue decidido de ese modo porque se piensa que es una manera fácil en la cual el usuario puede seleccionar el modelo a utilizar.

Dentro de la ventana el usuario se contará con tres opciones de modelos diferentes, en la cual al seleccionar un modelo se generará una gráfica en la cual se verán los diferentes resultados con las variables que se utilizan dentro de ese modelo logrando tener posibles predicciones para las calificaciones de los estudiantes.

En la figura 4.11 se muestra una parte del código desarrollado para la ventana de regresión simple, este código muestra los *imports* requeridos, la creación de la ventana principal y las pestañas en las cuales se mostrarán los modelos, cada uno con sus especificaciones además de la tabla generadas con Python.

```
import React from "react";
import {Form, Button, Row, Col, Tabs, Tab} from "react-bootstrap";

export default function SimpleReg(){

    return (
        <>
        <div>
            <h1>Pagina Regresión Simple</h1>
        </div>
        <div className="">
            <p>
               Dentro de esta página vemos una regresión simple en donde se visualizara la gráfica de predicciónes
               <br/>Instrucciones: Seleccione el atributo con el cual quiere generar la gráfica de predicción.
            </p>
        </div>
        <div className="container">
            <Tabs defaultActiveKey="profile" id="uncontrolled-tab-example" className="mb-3">
            <Tab eventKey="home" title="Modelo 1">
            <Form.Group as={Row} className="mb-3" controlId="model1info">
                <p>
                    Tiempo de estudio (studytime en inglés):<br/>
                    •   1 = Menos de 2 horas.<br/>
                    •   2 = De 2 a 5 horas.<br/>
                    •   3 = De 5 a 10 horas.<br/>
                    •   4 = Mayor a 10 horas.<br/>
                    <br/>
                    G3 (Calificación final y a predecir)
                    •   Va desde 0 hasta 20 puntos máximos.
                </p>
            </Form.Group>
                <div className="container">
                    <Button variant="primary" type="submit">
                    Enviar
                </Button>
                    <hr/>
                    <h1>Resultado obtenido: </h1>
                </div>
            </Tab>
            <Tab eventKey="profile" title="Modelo 2">
```

Figura 4. 11 Código de la ventana regresión simple.
Obtenido de: Elaboración propia.

4.8.2 Desarrollo ventana regresión Múltiple

Dentro de esta pestaña al igual que en la ventana de regresión simple se realizó la creación de un *tabbed* en el cual se mostrarán los diferentes modelos posibles a utilizar para la predicción de calificaciones.

Dentro de estos modelos se mostrarán diferentes opciones para que el usuario seleccione y así logre la predicción de la calificación, todo basándose en información que contenga del mismo estudiante

En la figura 4.12 y figura 4.13 se muestra el código realizado para la página de regresión múltiple en la cual se muestran los *imports*, el proceso desarrollado para la selección de atributos que posee el alumno, además de mostrar la creación de las pestañas en donde se muestran los atributos que el usuario puede seleccionar.

```
import React, {useState} from "react";
import {Form, Button, Row, Col, Tabs, Tab} from "react-bootstrap";
import axios from "axios";


export default function MultipleReg(){


    const [firstModelValues, firstModelSetValues] = useState({
        motherEducation: "",
        fatherEducation: "",
        studentAge: 0
    });

    const [secondModelValues, secondModelSetValues] = useState({
        motherEducation2: "",
        absences: 0,
        studentFreeTime2: 0
    });

    const [thirdModelValues, thirdModelSetValues] = useState({
        studentStudyTime3: 0,
        absences: 0,
        rejectedGrades3: 0
    });

    function handleSubmitFirstModel(evt) {
        evt.preventDefault();
        //Llamado a Endpoint del primer modelo aquí
        axios.post("https://localhost:5001/api/User/FirstModelSubmit", {
            motherEducation: firstModelValues.motherEducation,
            fatherEducation: firstModelValues.fatherEducation,
            studentAge: firstModelValues.studentAge
        }).then((res) => {
            if (res.status === 200) {
                console.log(res);
            }
        }).catch((err) => {
            console.log("Error en handleSubmitFirstModel" + err);
        });
```

Figura 4. 12 Código de la ventana regresión múltiple (1).
Obtenido de: elaboración propia.

```
<div className="container">
    <Tabs defaultActiveKey="profile" id="uncontrolled-tab-example" className="mb-3">
    <Tab eventKey="home" title="Modelo 1">
    <Form onSubmit={handleSubmitFirstModel}>
    <Form.Group as={Row} className="mb-3" controlId="formMotherInfo">
        <Form.Label column sm={5}>
        Nivel de educacion de la madre:
        </Form.Label>
        <Col sm={7}>
        <Form.Control as="select" placeholder="Seleccionar..." value={firstModelValues.motherEducation} onChange={handleChangeFirstModel}>
            <option value="0">Sin estudios</option>
            <option value="1">Educacion primaria (hasta 4° grado)</option>
            <option value="2">Hasta 3° de secundaria</option>
            <option value="3">Preparatoria</option>
            <option value="4">Licenciatura o superior</option>
        </Form.Control>
        </Col>
    </Form.Group>
    <Form.Group as={Row} className="mb-3" controlId="formFatherInfo">
        <Form.Label column sm={5}>
        Nivel de educacion del padre:
        </Form.Label>
        <Col sm={7}>
        <Form.Control as="select" placeholder="Seleccionar..." value={firstModelValues.fatherEducation} onChange={handleChangeFirstModel}>
            <option value="0">Sin estudios</option>
            <option value="1">Educacion primaria (hasta 4° grado)</option>
            <option value="2">Hasta 3° de secundaria</option>
            <option value="3">Preparatoria</option>
            <option value="4">Licenciatura o superior</option>
        </Form.Control>
        </Col>
    </Form.Group>
    <Form.Group as={Row} className="mb-3" controlId="formStudAge">
        <Form.Label column sm={5}>
        Edad del Alumno:
        </Form.Label>
        <Col sm={1}>
        <Form.Control as="input" type="number" min="15" max="22" value={firstModelValues.studentAge} onChange={handleChangeFirstModel}>
        </Form.Control>
        </Col>
```

Figura 4. 13 Código de la ventana regresión múltiple (2).
Obtenido de: Elaboración propia.

4.8.3 Desarrollo ventana clustering

Dentro de esta ventana se estará utilizando una serie de componentes uno de los principales es un *select* también llamado selector en el cual el usuario tendrá que elegir cuantos agrupamientos quiere realizar.

Una vez que realice la selección de 2, 3 o 4 agrupamientos se mostrará una gráfica en la cual se muestran los grupos que se generaron con el programa Python que se explicó anteriormente.

En las figuras 4.14, se muestra el desarrollo de la ventana de clustering la cual muestra los *imports* para el uso del *API* desarrollada con C++, el cual contiene el desarrollo para la lectura de los programas Python dentro de la aplicación web con React. También figura 4.15 muestra el desarrollo del elemento que serán visualizados por el usuario como lo son los elementos *select* los cuales contendrán las opciones y el botón para la ejecución de la predicción.

```
import React from "react";
import Form from "react-bootstrap/Form";
import axios from "axios";
import { useState, useEffect } from "react";
import { BubbleChart } from "../../components/Chart";

export default function Clustering(){
  const [chartData, setChartData] = useState([]);
  const [loading, setLoading] = useState(false);

  async function fetchClustering(clusters = 2) {
    setLoading(true);
        axios.get("https://localhost:5001/api/User/Clustering", {
          params: {
            numClusters: clusters
          }
        })
```

```
.then(response => {
  const data = response.data;
  console.log(data);
  setChartData({
    datasets: [
      {
        label: 'blue dataset',
        data: Object.values(data)?.map((comp) => ({
          x: comp.Array_0.Componente_1,
          y: comp.Array_0.Componente_2,
          r: 5
        })),
        backgroundColor: 'rgba(53, 162, 235, 0.5)'
      },
      {
        label: 'red dataset',
        data: Object.keys(data)?.map((comp) => ({
          x: comp.Array_1.Componente_1,
          y: comp.Array_1.Componente_2,
          r: 3
        })),
        backgroundColor: 'rgba(255, 99, 132, 0.5)'
      }
```

Figura 4. 14 Código de la ventana Clúster.
Obtenido de: Elaboración propia.

```
useEffect(() => {
    fetchClustering()

      return () => {
        setChartData({});
      };
    }, [])

if (loading) {
  return <h1>Data is Loading </h1>
}
  return (
    <>
      <div>
          <h1>Clustering Page</h1>
      </div>
      <div className="container">
          <Form>
          <Form.Label>Selecciona una cantidad de clusters(agrupamientos):</Form.Label>
          <Form.Control as="select">
              <option>Seleccionar...</option>
              <option value="2">Dos</option>
              <option value="3">Tres</option>
              <option value="4">Cuatro</option>
          </Form.Control>

          </Form>
      </div>
      <div className="container">
          <BubbleChart data={chartData}/>
      </div>
    </>
  );
```

Figura 4. 15 Código de la ventana clustering para selección de agrupamientos.
Obtenido de: Elaboración propia.

4.8.4 Desarrollo ventana estadísticas básicas

Dentro de esta ventana se mostrarán y describirán todos los elementos que se están utilizando dentro del proyecto como lo son: el tipo de datos que poseen los atributos, una descripción de los datos que son utilizados dentro de este proyecto para que el usuario entendiera más a fondo los datos que realizan el proceso de predicción y agrupación.

La figura 4.16 muestra lo desarrollado para esta ventana, es la creación de pestañas las cuales mostraran las diferentes tablas de descripción de datos como lo son correlaciones tipos de datos, etc.

```
<div>
    <h1>Pagina Estadisticas Basicas</h1>
</div>
<div className="">
    <p>
    Dentro de esta ventana usted podrá observar diferentes elementos y caracteristicas de los atributos con los cuales se está trabajando.
    Un poco de historia: Este proyecto está utilizando un data set público que se obtuvo de un proyecto llamado "Predicting Grades for the
    Este data set contiene un total de 392 elementos los cuales contienen información sobre los alumnos de la institución.
    </p>
</div>

<div className="container">
    <Tabs defaultActiveKey="profile" id="uncontrolled-tab-example" className="mb-3">
    <Tab eventKey="home" title="Atributos utilizados">
    <Form.Group as={Row} className="mb-3" controlId="Atributos">
    </Form.Group>
        <div className="container">
            <h1>poner archivo csv</h1>
        </div>
        </Tab>


    <Tab eventKey="profile" title="Descripción de atributos">
    <Form.Group as={Row} className="mb-3" controlId="Description">
        <p>Los datos que un alumno pueden ser de gran beneficio para nosotros ya que nos brindan una gran cantidad de información sobre su
            estaremos usando los cuales serán descritos a continuación.</p>
    </Form.Group>
    <Table striped bordered hover size="sm">
    </Table>
        </Tab>

    <Tab eventKey="contact" title="Tipos de datos">
    <Form.Group as={Row} className="mb-3" controlId="tipe">
        <p>Otro de los elementos que describiremos en este proyecto es el tipo de dato que posee dicha caracteristica del estudiante ya qu
            el usuario logre entender un poco más los elementos de entrada que necesita</p>
    </Form.Group>
    <Table striped bordered hover size="sm">
    </Table>
        </Tab>
```

Figura 4. 16 Código de la ventana estadísticas básicas.
Obtenido de: Elaboración propia.

4.8.5 Desarrollo ventana redes neuronales

Dentro de esta ventana se mostrará una serie de gráficas que integran las predicciones de calificaciones obtenidas, basadas en las regresiones.

En la figura 4.17 se ven las gráficas obtenidas las cuales serán acompañadas de una descripción de los resultados para una mejor comprensión por parte del usuario.

```jsx
import React from "react";
import {Form, Button, Row, Col, Tabs, Tab} from "react-bootstrap";

export default function SimpleReg(){

    return (
        <>
        <div>
            <h1>Página Redes Neuronales</h1>
        </div>

        <div className="" align="justify">
            <p>
            Dentro de esta pestaña de muestran las predicciones obtenidas con un programa desarrollado con Python
             en donde se usan una serie de atributos como son las regresiones, agrupaciones, pre procesamientos, transformada,
             selecciones entre otros, para asi lograr obtener la predicción de calificaciones y predicciones de otros atributos
            </p>

         <div>
            <p>La primera grafica obtenida con redes neuronales muestra las variables numéricas que poseen ciertos
                 atributos mostrando así la cantidad de alumnos que tienen esa caracteristica en especial.
            </p>
                <img width="100%"
                    className="d-block w-20px"
                    src="/images/imagen_redesneuro1.png"
                    alt=""
                />
        </div>
            <br/>
        <div>
            <p>Está siguiente grafica tambien muestra la cantidad de alumnos que poseen esa caracteristica en especial solo que todos estos atributos
                son los que contienen resultados y/o estudiantes con esa caracteristica en especial.
            </p>
                <img width="100%"
                    className="d-block w-20px"
                    src="/images/imagen_redesneuro2.png"
                    alt=""
                />
        </div>
            <br/>
```

Figura 4. 17 Código de la ventana redes neuronales.
Obtenido de: Elaboración propia.

CAPÍTULO V. RESULTADOS Y DISCUSIÓN

5.1 Modelos

Los resultados obtenidos dentro de los modelos consisten en unas tablas comparativas donde se logra observar cómo interactúan los modelos de regresión múltiple y de agrupamiento (clúster) con diferentes atributos, los cuales pueden verse en el capítulo anterior en la tabla 4.4 y en la tabla 4.5.

Se esperaba que los coeficientes de determinación obtenidos con el modelo de regresión múltiple y simple, fueran mucho más altos al igual que en el modelo de agrupación donde se esperaba una mayor certeza a la hora de hacer los agrupamientos; con el que obtuvo un mejor índice de certeza y aproximación de calificaciones fue con el modelo de redes neuronales manifestando un 0.89 de precisión el cual es bastante y comparado con las calificaciones reales esta excelente.

También dentro de los resultados obtenidos en los modelos están: las tablas obtenidas con el modelo de regresión simple la cual muestra una predicción de calificaciones con base a un solo atributo.

En la figura 5.1 se observa la gráfica obtenida con una regresión simple, utilizando como atributo principal las fallas obtenidas por parte del alumno contra la calificación final.

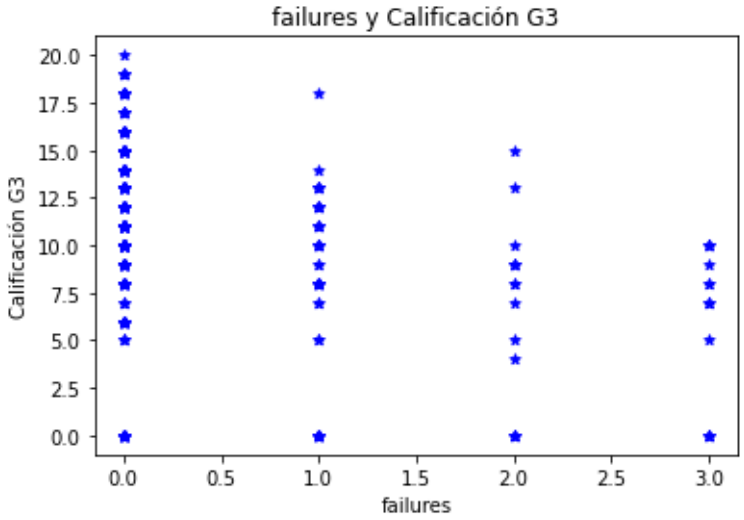


Figura 5. 1 Grafica obtenida con modelo de regresión simple
Obtenido de: Elaboración propia.

5.2 Aplicación web

Los resultados obtenidos para la aplicación web son realmente gratificantes ya que se logró una serie de ventanas en las cuales se logra observar de manera óptima los modelos de LA.

Lo primero que se generó dentro de la aplicación web es una pantalla de inicio en la cual se muestra un carrusel en donde se muestran tres aspectos principales los cuales son: página principal del tecnológico, página principal del SII y un enlace a información sobre LA.

En la figura 5.2 se muestra como quedo el desarrollo del carrusel mencionado anteriormente, el cual cuenta con la utilización de imágenes y videos creados especialmente para la aplicación web, además de que cada imagen envía al usuario a las diferentes páginas.

Figura 5. 2 Carrusel final de la aplicación web
Obtenido de: Elaboración web.

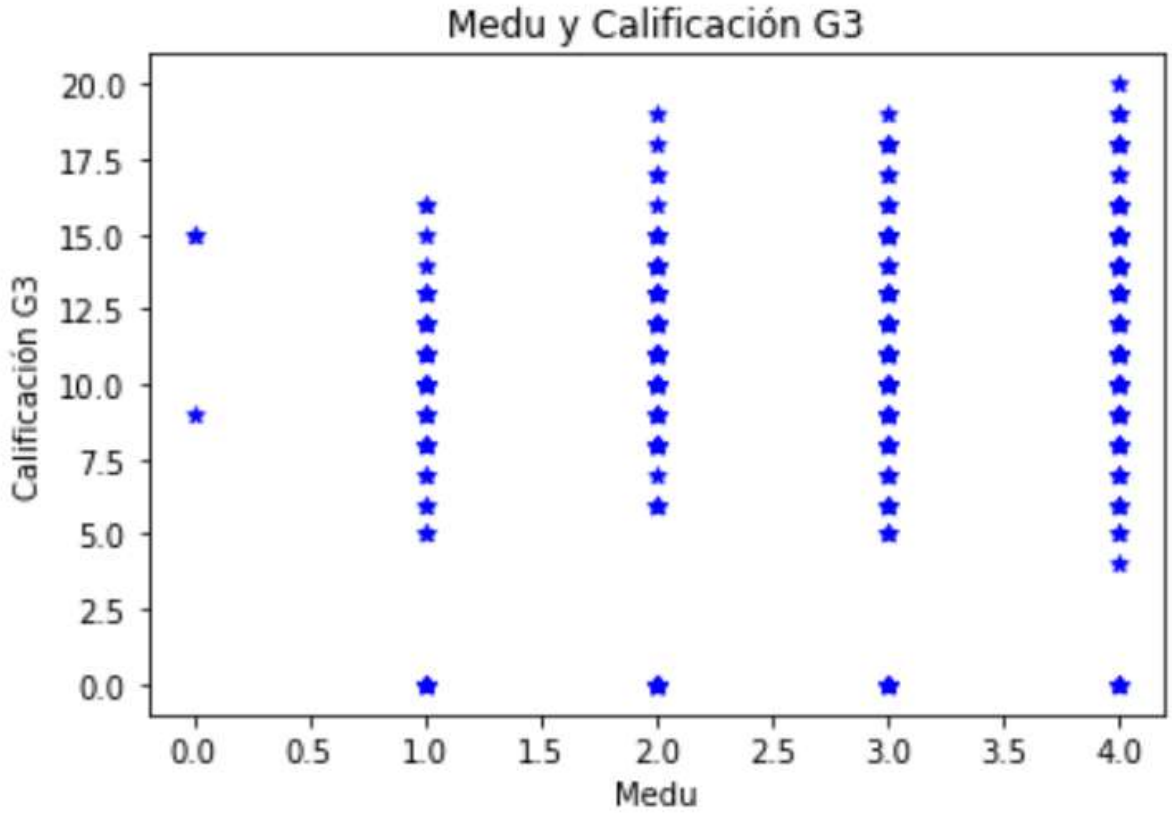
5.2.1 Ventana Regresión Simple

Como se observa en la figura 5.3 dentro de esta pestaña se logró crear 3 modelos distintos para que el usuario pueda seleccionar y crear una tabla de predicción en la cual mostrara a todos los alumnos que posean esa característica contra la calificación final, además de tener una descripción de lo que representa cada una de las tablas.

Otro de los resultados obtenidos y que se le mostraran al usuario, son el coeficiente de determinación que tiene el modelo que están utilizando en ese momento y una lista de las posibles predicciones de calificaciones que se lograron obtener con ese modelo y atributo en específico.

Resultado obtenido:

Dentro de este modelo estamos utilizando los estudios máximos de la madre para obtener la predicción de G3 (calificaciones finales). En la gráfica podemos observar en el eje de las X 5 elementos los cuales hacen referencia al nivel de estudios que poseen las madres de los estudiantes y en donde se observa que la mayoría de los estudiantes cuentan con una madre que estudio hasta la universidad o más.



Nota

Para este modelo tambien se determinó lo siguiente:

* El coeficiente de determinación es de: 0.047153035079265826

* La media de calificaciones es de: 7.916681857662159

Algunas de las calificaciones que se obtienen con este modelo son las siguientes:

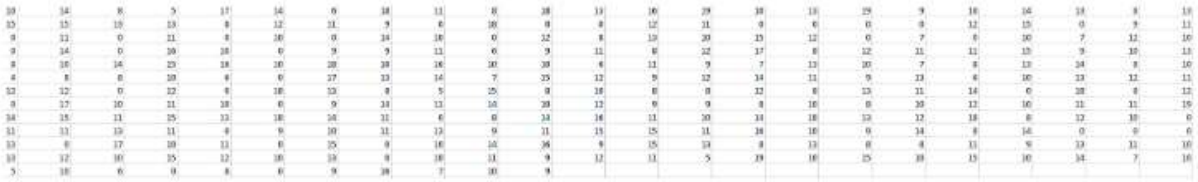


Figura 5. 3 Página final de regresión simple
Obtenido de: Elaboración propia.

5.2.2 Ventana Regresión Múltiple

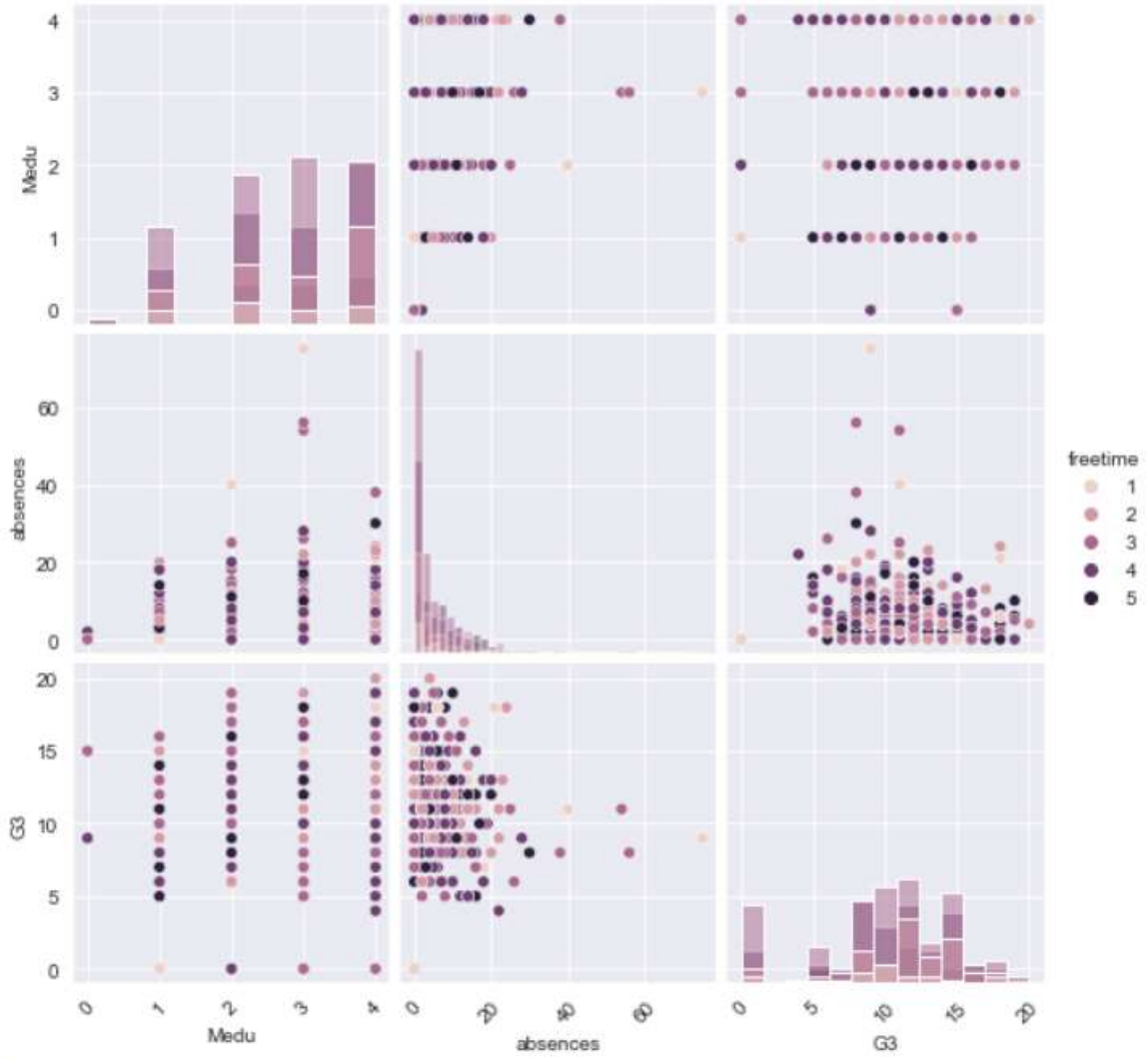
Como se muestra en la figura 5.4 las pestañas que estarán disponibles dentro de la ventana de regresión múltiple tendrán diferentes atributos, éstos representan a diferentes modelos posibles a generar por ejemplo, en el modelo 2 el usuario estará utilizando la educación de la madre, la educación del padre y la edad del estudiante para poder predecir la calificación de los estudiantes que contengan las características que seleccione.

Una vez que el usuario presione el botón enviar generará la predicción y será acompañada de una gráfica, en la cual se muestran representados todos los alumnos que posean esas características y se obtiene una descripción para un mejor entendimiento.

Además igual que en la ventana de regresión simple, este modelo también sera acompaña del coeficiente de determinación del modelo para que el usuario vea que tan certero es dicho modelo.

Esta gráfica fue obtenida con base a todos los elementos que contienen los atributos ya mencionados. En esta gráfica se puede observar la cantidad de alumnos que poseen esa característica contra la calificación final.
Como se observa en la parte inferior en medio en donde esta G3 vs absences y en la cual se tiene el freetime de los alumnos y la educación máxima de la madre con base a eso.

Ejemplo: Con esta tabla se logra observar que un alumno tiene 1 hora libre, tiene más de 60 faltas y que aproximadamente tiene una calificación de 90.



Nota

Para este modelo también se determinó lo siguiente:

* El coeficiente de determinación es de: 0.04733910819444753

* Con un error de: 4.466023278619253

Figura 5. 4 Ventana final ventana regresión múltiple
Obtenido de: Elaboración propia.

5.2.3 Ventana Clustering

Como se muestra en la figura 5.5 la gráfica generada con ayuda del programa Python, dicha grafica muestra la creación de un clustering de dos agrupamientos los cuales estan divididos por colores los cuales son: azul (rendimiento alto), rojo (rendimiento bajo) y los cuales muestran su nivel de riesgo académico.

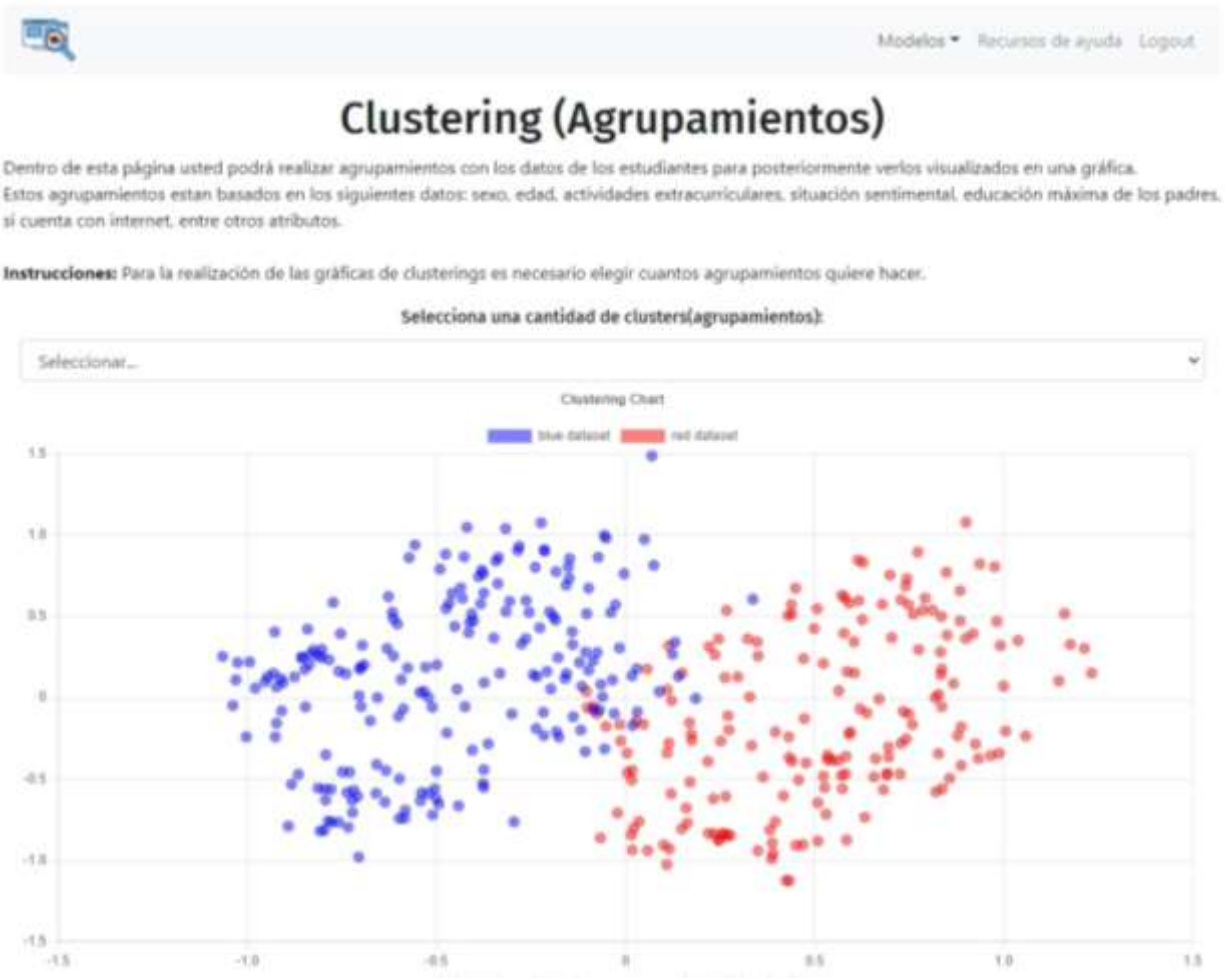


Figura 5. 5 Ventana final Clustering con creación de 3 grupos.
Obtenido de: Elaboración propia.

5.2.4 Ventana Estadísticas Básicas

Como se observa en la figura 5.6, figura 5.7, figura 5.8, figura 5.9 y figura 5.10, dentro de esta ventana se muestran una serie de pestañas en la cuales se tienen diferentes tablas donde se describirán los diferentes aspectos de los datos con los que se está trabajando, dichas pestañas son: atributos utilizados, tipo de datos, correlaciones entre los datos y gráficas con descripción de lo que está sucediendo con dicha gráfica.

school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	reason	guardian	traveltime	stud
GP	F	18	U	GT3	A	4	4	at_home	teacher	course	mother	2	2
GP	F	17	U	GT3	T	1	1	at_home	other	course	father	1	2
GP	F	15	U	LE3	T	1	1	at_home	other	other	mother	1	2
GP	F	15	U	GT3	T	4	2	health	services	home	mother	1	3
GP	F	16	U	GT3	T	3	3	other	other	home	father	1	2
GP	M	16	U	LE3	T	4	3	services	other	reputation	mother	1	2
GP	M	16	U	LE3	T	2	2	other	other	home	mother	1	2
GP	F	17	U	GT3	A	4	4	other	teacher	home	mother	2	2
GP	M	15	U	LE3	A	3	2	services	other	home	mother	1	2
GP	M	15	U	GT3	T	3	4	other	other	home	mother	1	2
GP	F	15	U	GT3	T	4	4	teacher	health	reputation	mother	1	2
GP	F	15	U	GT3	T	2	1	services	other	reputation	father	3	3
GP	M	15	U	LE3	T	4	4	health	services	course	father	1	1

Figura 5. 6 Ventana final estadísticas básicas pestaña atributos.
Obtenido de: Elaboración propia.

Atributos utilizados | Descripción de atributos | Tipos de datos | correlaciones | graficas de estadisticas

Los datos que un alumno pueden ser de gran beneficio para nosotros ya que nos brindan una gran cantidad de información sobre su rendimiento por eso es importante es conocer más profundidad los elementos que estaremos usando los cuales serán descritos a continuación.

Atributo	Descripción	Abreviación
School	Escuela del estudiente	'GP' - Gabriel Pereira o 'MS' - Mousinho da Silveira
Sex	Sexo del estudiante	'F' - femenino o 'M' - masculino
Age	Edad del estudiante	Desde 15 años a 22 años
Address	Dirección	'U' - urbano o 'R' - rural
Famsize	Tamaño de la familia	'LE3' - menor o igual a 3 o 'GT3'- mayor que 3
Pstatus	Estado de la familia	T'- Viviendo junto a sus padres o 'A' - Padres separados
Medu	Educación de la madre	0 = Sin estudios. 1= Educación primaria (hasta 4° grado). 2= Hasta 3 de secundaria. 3= Preparatoria. 4 = Licenciatura o superior.
Fedu	Educación del padre	0 = Sin estudios. 1= Educación primaria (hasta 4° grado). 2= Hasta 3 de secundaria. 3= Preparatoria. 4 = Licenciatura o superior.
Mjob	Trabajo de la madre	maestro, trabajos de salud, servicios civiles, casa, otro
Fjob	Trabajo del padre	maestro, trabajos de salud, servicios civiles, casa, otro
Reason	Razón de elección de la escuela	cerca de casa, reputación de la escuela, otro
Guardián	Encargado del estudiante	Madre, padre u otro

Figura 5. 7 Ventana final estadísticas básicas pestaña descripción.
Obtenido de: Elaboración propia.

Atributos utilizados | Descripción de atributos | Tipos de datos | correlaciones | graficas de estadisticas

Otro de los elementos que describiremos en este proyecto es el tipo de dato que posee dicha caracteristica del estudiante ya que es importante conocer qué tipo de elementos acepta para que de ese modo el usuario logre entender un poco más los elementos de entrada que necesita

#	Dato	Tipo de dato
1	Escuela	Objeto (elemento que contiene letras)
2	Sexo	Objeto (elemento que contiene letras)
3	Edad	Int64 (elemento que contiene solo números)
4	Dirección(elemento que contiene letras)	Int64 (elemento que contiene solo números)
5	Tamaño de la familia	Objeto (elemento que contiene letras)
6	Situación de los padres	Objeto (elemento que contiene letras)
7	Educación máxima de la madre	Int64 (elemento que contiene solo números)
8	Educación máxima del padre	Int64 (elemento que contiene solo números)
9	Trabajo de la madre	Objeto (elemento que contiene letras)
10	Trabajo del padre	Objeto (elemento que contiene letras)
11	Razón de elección de escuela	Objeto (elemento que contiene letras)

Figura 5. 8 Ventana final estadísticas básicas pestaña tipos de dato
Obtenido de: Elaboración propia.

Atributos utilizados | Descripción de atributos | Tipos de datos | correlaciones | graficas de estadisticas

Con ayuda de esta tabla se puede observar las correlaciones de los atributos. Las correlaciones se definen como: una medida que expresa hasta qué punto dos variables estan relacionadas literalmente. Permitiendo de ese modo describir las relaciones simples que tenemos con la utilización de nuestros datos.

Atributo	Correlación
Calificación final (G3)	1.000000
Educación máxima de la madre (Medu)	0.217147
Nivel Máximo de estudio (b_higher_education)	0.182465
Education máxima del padre(Fedu)	0.152457
Pago de clases extras (b_paixtraclasses)	0.101996
Disposición de internet(b_internet)	0.098483
Tiempo de estudio (studytime)	0.097820
Estado familiar (b_Pstatus)	0.058009
Recidencia (b_nursery)	0.051568
Relación familiar (famrel)	0.051363
Faltas (absences)	0.034247
Actividades extras (b_xtraactivities)	0.016100

Figura 5. 9 Ventana final estadísticas básicas pestaña correlaciones.
Obtenido de: Elaboración propia.

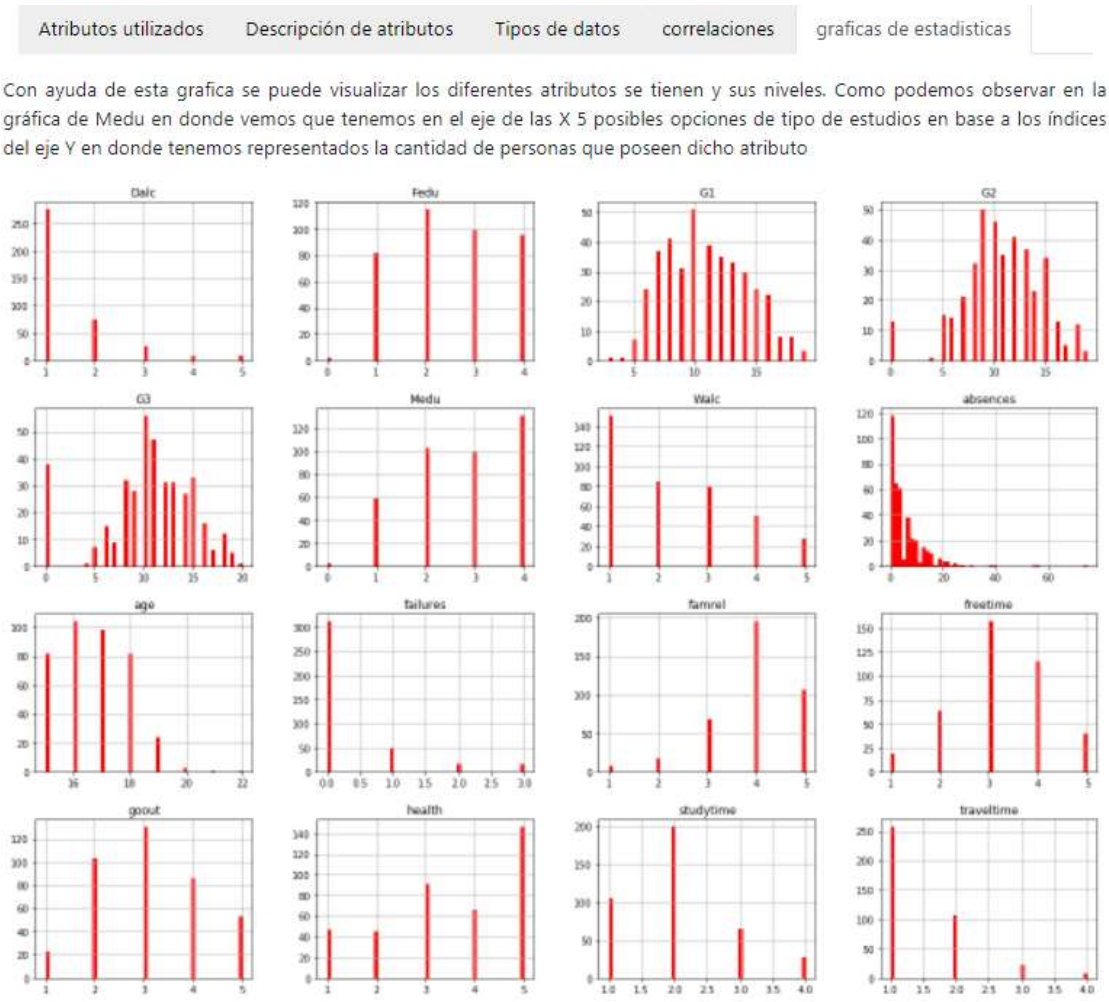


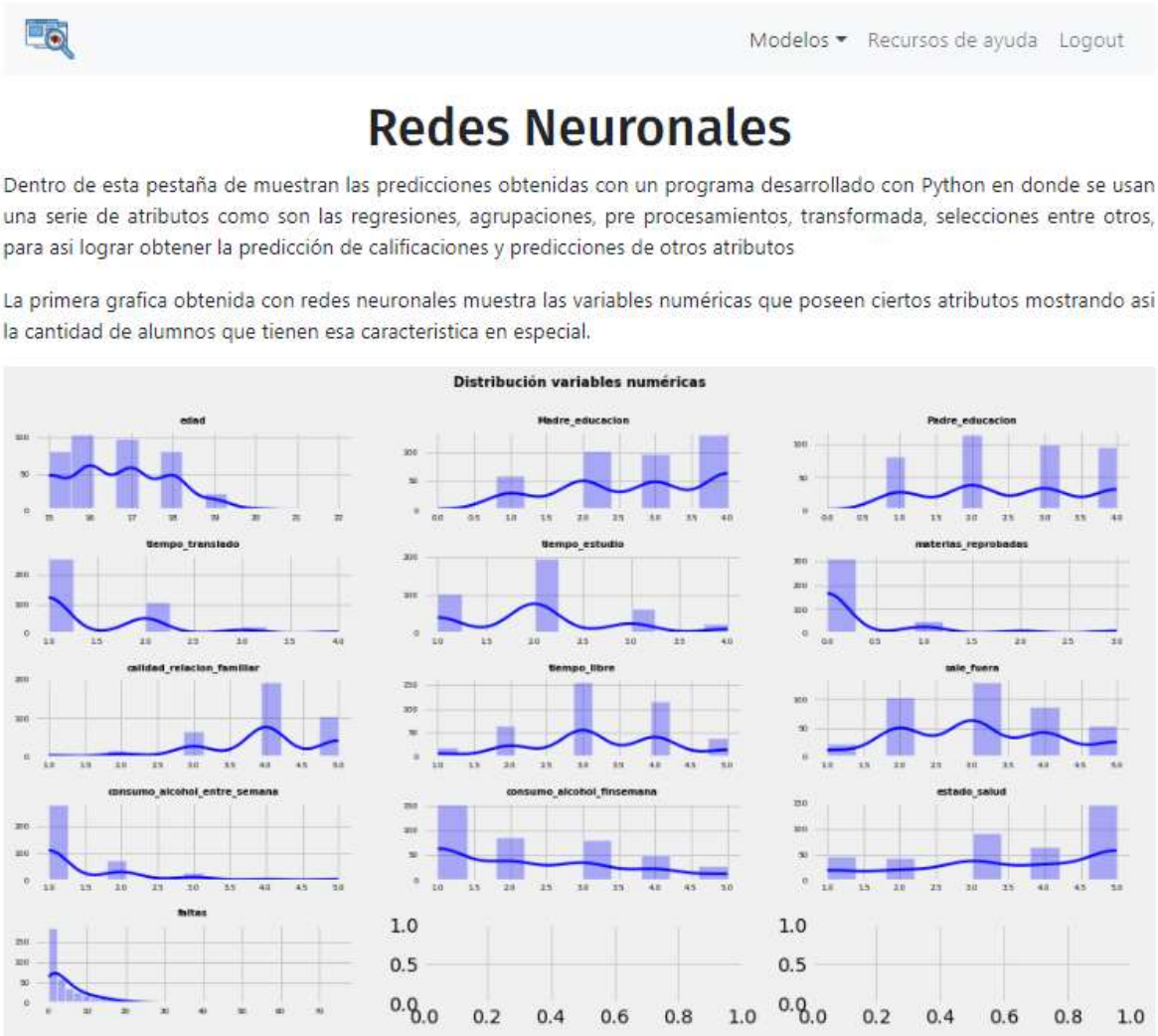
Figura 5. 10 Ventana final estadísticas básicas pestaña Gráficas.
Elaborado de: Elaboración propia.

5.2.5 Ventana Redes Neuronales

Como se muestra en la figura 5.11 la ventana creada para redes neuronales cuenta con varias tablas generadas con ayuda del modelo desarrollado con Python, las cuales muestran una serie de predicciones de estadísticas básicas para el pronóstico de las calificaciones de alumnos.

También se observa una gráfica la cual muestra la obtención de las predicciones con ayuda de este modelo donde se comparan con las calificaciones reales.

Y al igual que en los modelos de regresión simple y múltiple aquí también se muestra el coeficiente de determinación y la lista de calificaciones posibles a obtener.



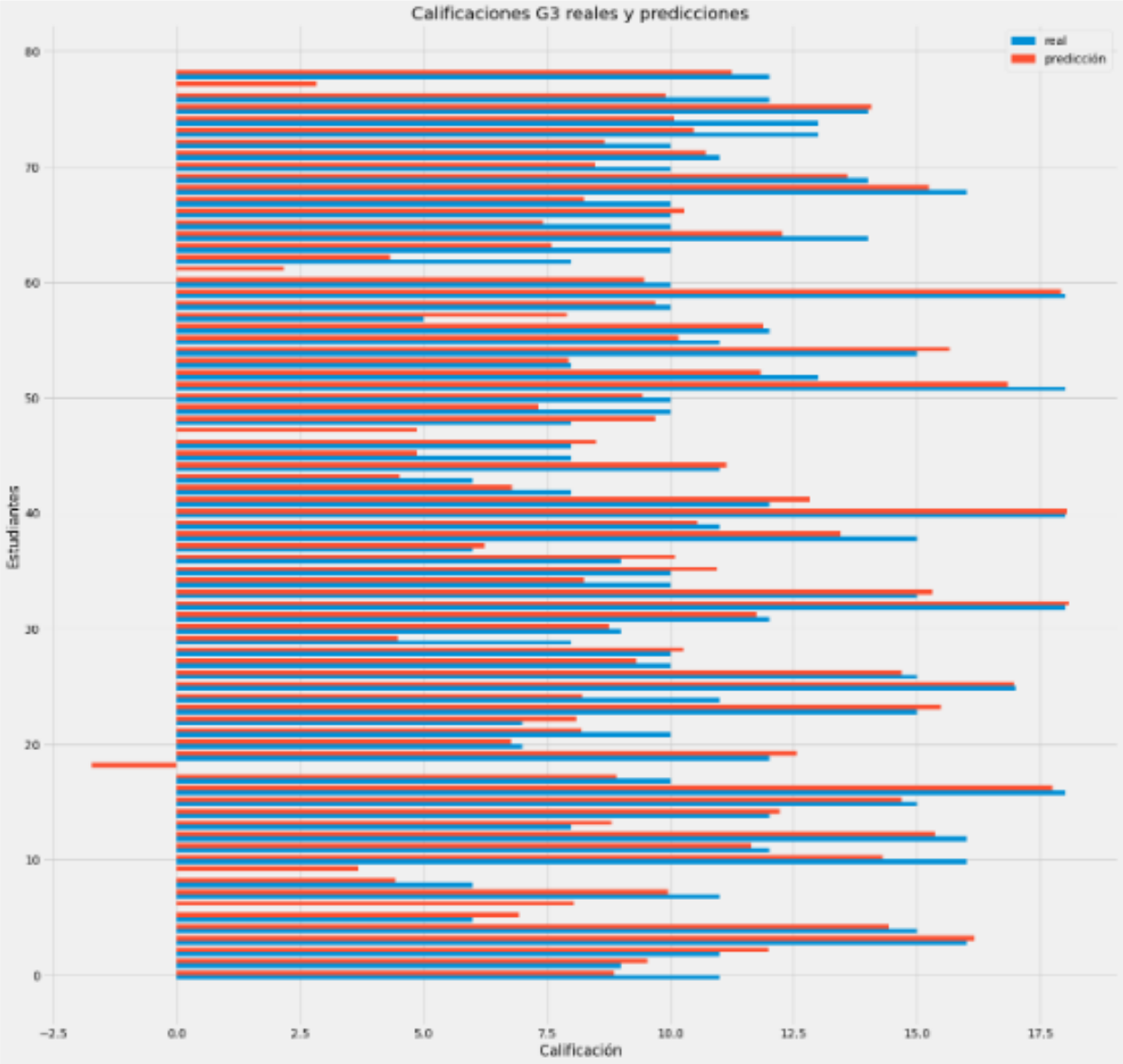
Por ultimo tenemos la gráfica en donde se muestran las predicciones obtenidas con este modelo en donde tenemos dos colores, la linea azul es la que muestra las calificaciones que estan dentro del dataset que estamos utilizando el cual es el que contiene toda la información del y los estudiantes y en el cual contiene las calificaciones finales, para compararla con la linea roja la cual es la predicción obtenida después de ser procesada con el modelo desarrollado de redes neuronales.

Nota: Si gusta comparar las calificaciones puede verlas dentro de la pestaña de estadísticas básicas en el apartado de Atributos utilizados.

Importante

Una de las cosas que se logró analizar con esta grafica fue la obtención del coeficiente de determinación y el error que genero el modelo desarrollado.

- Error obtenido: 1.8039812250113978
- Coeficiente de determinación: 0.8307890736326884



Logrando obtener tambien estas posibles predicciones de calificaciones, las cuales pueden ser compardas con las calificaciones que se encuentran en estadisticas basicas en la pestaña de Atributos utilizados.

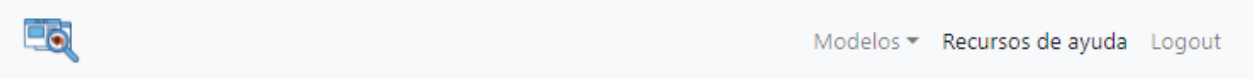
8.92847037	8.92847037	8.9284704	8.92847037	8.92847037	8.92847037
8.33116472	10.0421301	4.7197388	3.57222709	14.28375212	11.66887098
15.39717669	8.03788457	12.210169	14.71575222	17.77314794	8.80820949
17.1109722	12.6770696	6.1040968	8.0615123	8.01700758	15.5203179
8.01337716	17.0604604	14.652527	9.37149311	10.50123873	4.64023771
8.70182299	11.5563703	18.026731	15.39654398	8.18786855	10.91696402
9.98162656	5.70891204	13.375689	10.57933405	18.05240591	12.86020836
6.80268816	4.30864936	11.291972	4.51153901	8.60006126	4.3009419
9.61054674	7.38460488	9.6736106	16.83368813	12.1655297	8.38276978
15.68465154	10.0871387	11.737607	7.79648568	9.56207077	17.85169477
9.78251084	1.88672972	4.1509112	7.66914638	12.21017764	7.57525154
10.37004066	8.14669049	15.307087	13.66147419	8.52463862	10.68200445
8.62372897	10.7760522	10.610733	14.18176383	9.81867697	2.69480227
11.23151045					

Figura 5. 11 Ventana final redes neuronales.
Obtenido de: Elaboración propia.

.

5.2.6 Ventana Recursos de ayuda

En la figura 5.12 se muestra esta ventana con diferentes enlaces a los artículos en los cuales tanto profesores como alumnos, podrán obtener consejos de cómo mejorar las técnicas de estudio y aprendizaje para reducir también riesgos académicos.



#	Nombre del artículo	Autor	Link
	Recursos para profesores		
1	Cómo ayudar a los estudiantes a mejorar su aprendizaje	Desconocido	Link
2	6 estrategias que usan los maestros para ayudar a los niños que piensan y aprenden diferente	El equipo de Understood	Link
3	¿De qué manera puedo apoyar mejor a mi estudiante?	Desconocido	Link
4	¿Cómo ayudar al desarrollo personal del estudiante?	Universidad de Navarra	Link
5	Comparación de estrategias de estudio y autorregulación en universitarios.	Irma Rosa Alvarado Guerrero	Link
6	Metas académicas, estrategias cognitivas y estrategias de autorregulación del estudio	Antonio Valle, Ramón G.	Link
	Recursos de ayuda para estudiantes		
1	Formas de estudiar y ser más eficaces	Desconocido	Link
2	Técnicas de estudio	Desconocido	Link
3	12 Técnicas de estudio para potenciar tu aprendizaje	Idat	Link
4	Las mejores técnicas de estudio	Educación 3.0	Link
5	Métodos de estudios para la universidad	Aliat Universidades	Link

Figura 5. 12 Ventana final recursos de ayuda.
Obtenido de: Elaboración propia.

CAPÍTULO VI. CONCLUSIONES

Dentro de diversos artículos se observa que los expertos en innovación pedagógica, se muestran partidarios de introducir las técnicas de la analítica del aprendizaje dentro de diversos procesos de enseñanza y aprendizaje; esto, porque se pueden obtener diversos beneficios que se hacen patentes tanto para el alumno como para el centro educativo (Rodríguez Canfranc, P., 2019).

Como conclusión principal se tiene que la realización, recopilación y análisis de información obtenida por parte de los distintos sistemas de información, son de gran utilidad puesto que LA, no sólo debe ser aplicada dentro del proceso final de recolección de los datos, sino que debe aplicarse durante todo el procedimiento de información, ya que no sólo impacta a los contenidos, sino que también podemos ver efectos dentro de los alumnos y los profesores.

Por ende, el realizar y hacer uso de estos modelos de LA sería de gran beneficio para futuros estudiantes del sistema tecnológico, ya que sólo sería tomar la información generada por los sistemas computacionales de la institución educativa.

Dentro de cualquier investigación se tienen puntos de mejora, en este caso en el modelo de regresión múltiple, es importante aumentar la precisión del coeficiente de determinación para mayor certeza en los resultados. Al igual que en el modelo de clustering que tiene una gran división de elementos es importante incrementarla, para que sea mucho más exacta y de ese modo mejoren sus agrupamientos y siendo más precisos a la hora de realizar los grupos.

Pero como se menciona dentro del artículo de Heredia, D. P. (2021)

> *"La toma de decisiones humana, aunque a menudo tenga defectos, tienen una gran virtud: puede evolucionar, los sistemas automatizados permanecen congelados en el tiempo hasta que los ingenieros bucean en ellos para modificarlos. Debemos integrar de forma explícita mejores valores en nuestros algoritmos y crear modelos de big data que sigan nuestro ejemplo ético. Y a veces eso significa dar prioridad a la justicia antes que a los beneficios"***.**

Resumiendo, se logró elaborar y plantear el objetivo propuesto de la creación y visualización de modelos de LA, en el cual se abordan los principales aspectos como lo son: obtención de datos, limpieza, procesamiento, obtención de estadísticas básicas, modelos de regresión, agrupamientos y redes neuronales, en donde este último modelo obtuvo la mayor precisión.

Definitivamente con la ayuda de un modelo de LA se puede lograr una mejora dentro del aprovechamiento académico. Así mismo los resultados obtenidos durante el desarrollo de este proyecto servirán como base para mejoras futuras dentro del mismo, permitiendo el incremento de certeza en las predicciones y agrupamientos, mejorando de ese modo, la atención oportuna a estudiantes que se encuentran en riesgo académic

CAPÍTULO VII. REFERENCIAS

ALogos E-learning. (2021). *Learning Analytics en la educación. De primaria a bachillerato*. Blog Emagister. https://www.emagister.com/blog/learning-analytics-la-educacion-primaria-bachillerato/

Alberca, A. S. (2020a, octubre 4). *La librería Matplotlib*. Aprende con Alf. https://aprendeconalf.es/docencia/python/manual/matplotlib/

Alberca, A. S. (2020b, octubre 4). *La librería Numpy*. Aprende con Alf. https://aprendeconalf.es/docencia/python/manual/numpy/

Alberca, A. S. (2021, 14 mayo). *La librería Pandas*. Aprende con Alf. https://aprendeconalf.es/docencia/python/manual/pandas/

Apd, R. (2020). *¿Cuáles son los tipos de algoritmos del machine learning?* APD España. https://www.apd.es/algoritmos-del-machine-learning/

Bajo, S. N. (2002). *Redes neuronales: concepto, aplicaciones y utilidad en medicina | Atención Primaria*. Atención Primaria. https://www.elsevier.es/es-revista-atencion-primaria-27-articulo-redes-neuronales-concepto-aplicaciones-utilidad-13033737

Basak, S. B., Wotto, M. W., & Bélanger, P. B. (2018). *SAGE Journals: Your gateway to world-class research journals*. SAGE Journals. https://journals.sagepub.com/action/cookieAbsent

Berners-Lee, T., Hendler, J. y Lassila, O. (2001). The semantic web. *Scientific american, 284*(5), 28-37.

Castro, P. R. (2016). *Universidad de La Sabana*. Obtenido de https://www.redalyc.org/jatsRepo/834/83449754006/html/index.html

Casanova Cardiel, H. (2009). *Plan Educativo Nacional*. Plan Educativo Nacional UNAM. http://www.planeducativonacional.unam.mx/CAP_00/Text/00_05a.html

Coalla, J. (2021, 8 noviembre). *React | Qué es, para qué sirve y cómo funciona | Descúbrelo todo*. Tribalyte Technologies. https://tech.tribalyte.eu/blog-que-es-react

Como aplicar minería de datos. (2017). Apuntes & Cursos. https://www.apuntesycursos.com/como-aplicar-mineria-de-datos.html

Cortina, V. G. (2015). *Aplicación de la metodología CRISP-DM a un proyecto de minería de datos en el entorno universitario.* Marid.

Crisp-DM: *los 6 pasos del proceso de Data Mining - Blog Smartup*. (2019). Retrieved 21 December 2021, from https://blog.smartup.es/crisp-dm-6-pasos-proceso-data-mining/

Daysi García-Tinizaray, K. O.-B.-D. (2014). *Learning analytics para predecir la deserción de estudiantes a distancia*. Ecuador: Revista Científica de Tecnología Educativa; ISSN: 2255-1514.

Daysi Garcia, J. T. (2014). Research Gate. Obtenido de https://www.researchgate.net/publication/284031779_Learning_analytics_para_predecir_la_desercion_de_estudiantes_a_distancia_Learning_analytics_to_predict_dropout_of_distance_students

Denley, T. (2012). Austin Peay State University. Obtenido de Educause review: https://er.educause.edu/articles/2012/9/austin-peay-state-university-degree-compass

Duk, D. (2019). *K-Means: Agrupamiento con Minería de datos [Introducción]*. ESTRATEGIAS DE TRADING. https://estrategiastrading.com/k-means/

Elias, T. (2011). Learning analytics. *Learning*, 1-22.

E learning, A. (2017). *Emagister*. Obtenido de https://www.emagister.com/blog/learning-analytics-la-educacion-primaria-bachillerato/

García Tinisaray, D. K. (2016). *Depósito de Investigación Universidad de Sevilla*. Obtenido de https://idus.us.es/handle/11441/40436

Galán Cortina, V. (2016). Aplicación de la metodología CRISP-DM a un proyecto de minería de datos en el entorno universitario (Bachelor's thesis).

Gómez, P. (2021). Qué es una librería en programación. DevCamp. https://devcamp.es/que-es-libreria-programacion/#:%7E:text=En%20este%20sentido%2C%20una%20librer%C3%ADa,llanamente%2C%20es%20un%20archivo%20importable.

Lang, C., Siemens, G., Wise, A., & Gasevic, D. (Eds.). (2017). Handbook of learning analytics. New York: SOLAR, Society for Learning Analytics and Research.

ANALITICA BIG DATA - Fundamentos - Curso - luisamayateacher. (2021*). luisamayateacher.* https://sites.google.com/site/luisamayateacher/analitica-de-datos---curso

Hassan, R., Palaniappan, S., Mahamood, S., Abbas, A., Sarker, K., & Sattar, M. (2020, 4 junio). *Predecir el desempeño de los estudiantes en instituciones de educación superior mediante el uso de análisis de aprendizaje por vídeo y técnicas de minería de datos. MDPI, 1–20.* https://www.mdpi.com/2076-3417/10/11/3894

Heredia, D. P. (2021). *Learning analytics: El poder de los datos en educación*. Canal Educación y Sociedad. https://revistadigital.inesem.es/educacion-sociedad/learning-analytics/?fbclid=IwAR0EjFF_MMhVWHqnH8OJY4dzpIX3oWWbljR2BhSa0LnRcD1lKzqSyRDeEYk

Hubert.ai. (26 de 02 de 2017). *AI in education. Obtenido de* https://medium.com/@Hubert.ai/ai-in-education-the-genie-of-deakins-university-e29fbcde27d1

Jauregui, A. F. (2022). *Tutorial Sklearn Python. Ander Fernández.* https://anderfernandez.com/blog/tutorial-sklearn-machine-learning-python/

Joaquín Amat, R. (2020). *Clustering con Python. Ciencia de datos.* https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html

Jones, K., VanScoy, A., Bright, K. y Harding, A. (enero de 2021*). ¿Les importa siquiera? Medición del valor de la privacidad de los estudiantes por parte del instructor en el contexto de la analítica del aprendizaje. En Actas de la 54ª Conferencia Internacional de Ciencias de Sistemas de Hawái (p. 1529).*

Kleine, K., Giones, F., & Tegtmeier, S. (2019). The learning process in technology entrepreneurship education—Insights from an engineering degree. *Journal of Small Business Management*, 57, 94-110.

López, M. (2019). *MangoLife*. Obtenido de https://mangolife.mx/blog/universidad-mexico-6-causas-desercion-escolar

María Carolina Niño Rivera, J. A. (2015). *Herramienta de learning analytics para el proceso de aprendizaje en un aula virtual.* Bogotá: Fundación Universitaria los Libertadores.

Marketing. (2020, 11 mayo). *¿Qué es C#?* Besoftware. https://bsw.es/que-es-c/

Martinez Bachmann, J. (2017). *Predicting Grades for the School Year*. Kaggle. https://www.kaggle.com/janiobachmann/predicting-grades-for-the-school-year/notebook

Moine, J. M., Silvia Haedo, A., & Gordillo, S. (2011). *Estudio comparativo de metodologías para minería de datos. sedici.unlp.edu.ar.* http://sedici.unlp.edu.ar/bitstream/handle/10915/20034/Documento_completo.pdf?sequence=*1*

Ochoa, X. y Wise, AF (2021). Apoyar el cambio a lo digital con análisis de aprendizaje centrados en el estudiante. *Investigación y desarrollo de tecnología educativa*, *69* (1), 357-361.

Powered by Drexel University Online. (s.f.). Obtenido de https://virtuallyinspired.org/portfolio/deakin-university/#1489967899766-9ae8f741-5508

Ramos, R. (2021, 12 septiembre). *¿Qué es JavaScript y para qué sirve?* Agencia de Marketing Digital Sevilla - Rafa Ramos. https://soyrafaramos.com/que-es-javascript-para-que-sirve/

Red de Portales University Page. (2021). *universia.es.* https://www.universia.net/es/universidades/instituto-tecnologico-chihuahua-ii.01169.html

Release, M. (2015). *IBM Watson helps Deakin drive the digital frontier. Obtenido de:* https://www.deakin.edu.au/about-deakin/media-releases/articles/ibm-watson-helps-deakin-drive-the-digital-frontier

Regresión lineal múltiple en Python. (2020). *ICHI.PRO.* https://ichi.pro/es/regresion-lineal-multiple-en-python-21324445754792

Rodríguez Canfranc, P. (2019). *Learning Analytics: el poder del big data en la educación*. *Telos Fundación Teléfonica.* https://telos.fundaciontelefonica.com/la-cofa/learning-analytics-el-poder-del-big-data-en-la-educacion/

Romero, J. (2019). *jorgeromero.net. Obtenido de* https://jorgeromero.net/metodologias-de-mineria-de-datos/

Regresión lineal múltiple en Python. (2020). ICHI.PRO. https://ichi.pro/es/regresion-lineal-multiple-en-python-21324445754792

Rojas-Castro, Pablo (2017). Learning Analytics: Una revisión de la literatura. Educación y Educadores, 20(1),406-127. [fecha de Consulta 2022]. ISSN: 0123-1294. Disponible en: https://www.redalyc.org/articulo.oa?id=83449754006

Roldán, P. N. (2021). *Modelo de regresión*. Economipedia. https://economipedia.com/definiciones/modelo-de-regresion.html

Santiago, R. (2017). *Learning Analytics: la narración del aprendizaje a través de los datos. Editorial UOC.* https://elibro.net/es/ereader/itchihuahuaii/58637

Santander Universidades. (2022). *¿Qué es Python? | Blog*. *Becas Santander.* https://www.becas-santander.com/es/blog/python-que-es.html

Scikit-Learn, herramienta básica para el Data Science en Python. (s. f.). Máster en Data Science. Recuperado 2021, de https://www.master-data-scientist.com/scikit-learn-data-science/

Scikit-Learn, herramienta básica para el Data Science en Python. (2019). Máster en Data Science. https://www.master-data-scientist.com/scikit-learn-data-science/

Signals. (2009). Purdue University.edu. https://www.purdue.edu/uns/x/2009b/090827ArnoldSignals.html

Tecnológico Nacional De México Campus Chihuahua II. (20019). *Programa de Desarrollo Institucional 2019–2024.* Departamento de Planeación, Programación y Presupuestación. http://www.chihuahua2.tecnm.mx/wp-content/uploads/2021/10/Programa-de-Desarrollo-Institucional-PDI-2019-2024-Autorizado-por-TecNM.pdf

Tempelaar, D., Rienties, B. y Nguyen, Q. (2021). La contribución de la analítica del aprendizaje disposicional a la educación de precisión. Tecnología y sociedad educativas, 24 (1), 109-122.

Tinisaray, D. K. (2015). *Construcción de un modelo para determinar el rendimiento de los estudiantes basado en learning analytics, mediante el uso de técnicas multivariantes.* Sevilla.

Yesa, S. R. (2015). *Modelo de evaluación automática de competencia en el laboratorio remoto visir, a través de learning analytics y rúbricas de aprendizaje.* Deusto: University of Deus

LICENCIA DE USO OTORGADA POR LUISA YOLANDA QUIÑONES MONTENEGRO, de nacionalidad mexicana, mayor de edad, con domicilio **ubicado en Av. De las Industrias 11101 Complejo Industrial Chihuahua C.P.31130, en mi calidad de DIRECTORA** del Tecnológico Nacional de México campus Chihuahua II y titular de los derechos patrimoniales y morales de la tesis denominada **"MODELO DE LEARNING ANALYTICS BASADO EN INFORMACIÓN DEL SII Y DATOS GENERALES PARA PREDECIR EL RENDIMIENTO DE ALUMNOS"** en adelante **"LA OBRA"** quien para todos los fines del presente documento se denominará **"EL AUTOR Y/O EL TITULAR"**, a favor del Tecnológico Nacional de México, la cual se regirá por las cláusulas siguientes:

PRIMERA – OBJETO: "EL AUTOR Y/O TITULAR", mediante el presente documento otorga al Tecnológico Nacional de México, licencia de uso gratuita e indefinida respecto de **"LA OBRA"**, para almacenar, preservar, publicar, reproducir y/o divulgar la misma, con fines académicos, por cualquier medio en forma física y a través del repositorio institucional y del repositorio nacional, éste último consultable en la página: (https://www.repositorionacionalcti.mx/).

SEGUNDA - TERRITORIO: La presente licencia se otorga, de manera no exclusiva, sin limitación geográfica o territorial alguna, de manera gratuita e indefinida.

TERCERA -ALCANCE: La presente licencia contempla la autorización para formato uso de "LA OBRA" en cualquier formato o soporte material y se extiende a la utilización, de manera enunciativa más no limitativa a los siguientes medios: óptico, magnético, electrónico, virtual (red), mensaje de datos o similar conocido o por conocerse.

CUARTA – EXCLUSIVIDAD: La presente licencia de uso aquí establecida no implica exclusividad en favor del Tecnológico Nacional de México; por lo tanto, **"EL AUTOR Y/O TITULAR"** conserva los derechos patrimoniales y morales de **"LA OBRA"**, objeto del presente documento.

QUINTA – CRÉDITOS: El Tecnológico Nacional de México reconoce que el **"AUTOR Y/O TITULAR"** es el único, primigenio y perpetuo titular de los derechos morales sobre **"LA OBRA"**; por lo tanto, siempre deberá otorgarle los créditos correspondientes por la autoría de esta.

SEXTA – AUTORÍA: **"EL AUTOR Y/O TITULAR"** manifiesta ser el único titular de los derechos de autor que derivan de **"LA OBRA"** y declara que el material objeto del presente fue realizado por él, sin violentar o usurpar derechos de propiedad intelectual de terceros; por lo tanto, en caso de controversia sobre los mismos, se obliga a ser el único responsable.

Dado en la Ciudad de Chihuahua, Chih., a los 26 días del mes de mayo del 2022.

"EL AUTOR Y/O TITULAR"	**"EL INSTITUTO TECNOLÓGICO DE CHIHUAHUA II"**
VALERIA SARAÍ ÁVILA GRAJEOLA	**LUISA YOLANDA QUIÑONES MONTENEGRO**

ANEXOS

Anexo 1. Código modelo de regresión simple

```
"""
Created on Tue Jun 29 22:32:59 2021
@author: Valeria Saraí Avila Grajeola
"""

#Regresion Lineal Simple
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
#import pandas.util.testing as tm
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

#Cargamos el conjunto de datos
dataset = pd.read_csv('./input/student-mat.csv')

#models for 50% train and 50% test

X = np.array(dataset['Medu']).reshape(-1, 1)
y = np.array(dataset['G3']).reshape(-1, 1)
# Separating the data into independent and dependent variables
# Converting each dataframe into a numpy array
# since each dataframe contains only one column
#df_set.dropna(inplace = True)
# Dropping any rows with Nan values
X_train, X_test, y_train, y_test = train_test_split(X, y,train_size = 0.5,test_size = 0.5,random_state=0)
# Splitting the data into training and testing data
regr = LinearRegression()
regr.fit(X_train, y_train)
X_train.shape

#predicting the test result and visualizing the test result
y_pred=regr.predict(X_test)
y_pred
plt.scatter(X_test,y_test,color='orange')
plt.plot(X_test,regr.predict(X_test),color='black')
plt.title('failures  vs G3(Test Data 50%)')
plt.xlabel('failures ')
plt.ylabel('G3')
plt.show()

from sklearn.metrics import mean_squared_error

#sacamos el error de la predicción entre Y y la predicción que se está sacando
error=np.sqrt(mean_squared_error(y_test,y_pred))
r2=regr.score(X_test,y_test)
```

```
# Obtener coeficiente de determinación
r_sq = regr.score(X_test,y_test)
print('coefficient of determination:', r_sq)

print('El error es: ', error)

print('El error de r^2 es: ', r2)
```

Anexo 2. Código modelo regresión múltiple

```
"""
Created on Tue Jun 29 22:32:59 2021
@author: Valeria Saraí Avila Grajeola
"""
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

sns.set_style('darkgrid')

datos= pd.read_csv('./input/student-mat.csv')

#datos que vamos a estar checando
nuevo= datos[['studytime','absences','failures','G3']]

#nombrar gráfica, colores dependientes de la edad y el histograma
g= sns.pairplot(nuevo,hue='failures', diag_kind='hist')

#creamos ciclo for para una mejor visión de las graficas
for ax in g.axes.flat:
    plt.setp(ax.get_xticklabels(),rotation=45)

datos=datos.replace(np.nan,'0')
studytime=datos['studytime'].values
absences=datos['absences'].values
failures =datos['failures'].values
G3=datos['G3'].values

#generamos el arreglo que contendrá todas las caract. de la x
#ahora ponemos la traspuesta para que sea de columna a renglón
X=np.array([studytime ,absences, failures]).T
Y=np.array(G3)

#comenzamos con la parte del modelo de regresión
reg=LinearRegression()
#hacemos el ajuste del modelo
reg=reg.fit(X,Y)
#Hacemos la predicción
y_pred=reg.predict(X)
```

```
#sacamos el error de la predicción entre Y y la predicción que se está sacando
error=np.sqrt(mean_squared_error(Y,y_pred))
r2=reg.score(X,Y)

# Obtener coeficiente de determinación
r_sq = reg.score(X,Y)
print('coefficient of determination:', r_sq)

print('El error es: ', error)

print('El error de r^2 es: ', r2)

#valor de los coeficientes
print("Los coeficientes son: \n", reg.coef_)
#redefinimos valors para ver la prediccion
Medu=1
Fedu=2
age=18
print("Predicción de calificación: \n",reg.predict([[Medu,Fedu,age]]))
```

Anexo 3. Código modelo clustering con método K-Means

```
"""
Created on Tue Jun 29 22:32:59 2021
@author: Valeria Saraí Avila Grajeola
"""
import numpy as np #para cálculos científicos
import pandas as pd #para el análisis de datos
import matplotlib.pyplot as plt # para creación de graficas
from sklearn.cluster import KMeans # para importación del método

df=pd.read_csv('./input/student-mat.csv', engine='python')
df.info() #vemos que es lo que contiene el objeto datos
df.head() #vemos las filas de los datos

#linea que se usa para eliminar o no tomar en cuenta un elemento o columna
#df_variables=df.drop(['school'], axis=1)
df_variables=df.drop(['G1', 'G2', 'G3', 'address', 'Pstatus', 'reason', 'famsup', 'school', 'nursery', 'goout', 'sex'], axis=1)

#Aqui podremos observar todo los estatisticos maximos, minimos, cuartiles, primedio
#desviasion estandar, etc.
df_variables.describe()


#Utilizamos el mismo metodo que se utiliza en la data set para convertir
#todo a numeros enteros.


# 0 stands for F and 1 stands for M. [F=Femenino, M=Masculino]
# Here we will convert all the binary columns to integers.
#df_variables['b_sex'] = df_variables['sex'].apply(lambda x: 0 if x == 'F' else 1)
#df_variables['b_sex'].value_counts()
```

```
# 0 stands for U and 1 stands for R. [U=Urban, R=Rural]
# Here we will convert all the binary columns to integers.
#df_variables['b_address'] = df_variables['address'].apply(lambda x: 0 if x == 'U' else 1)
#df_variables['b_address'].value_counts()

# Interestingly there are more students in families that are greater than 3.
# Could it be possible that all family members are in the same school? This might be a reason why it is higher.
# LE3 = Less than 3. [0], GE3 = Greater than 3.[1]
df_variables['b_famsize'] = df_variables['famsize'].apply(lambda x: 0 if x == 'LE3' else 1)
df_variables['b_famsize'].value_counts()

# T = Parents are living together [0], A = Parents living apart. [1]
#df_variables['b_Pstatus'] = df_variables['Pstatus'].apply(lambda x: 0 if x == 'T' else 1)
#df_variables['b_Pstatus'].value_counts()

# 0 = no and 1 = yes
#df_variables['b_famsup'] = df_variables['famsup'].apply(lambda x: 0 if x == 'no' else 1)
#df_variables['b_famsup'].value_counts()


# 0 = no and 1 = yes
# This is an interesting column when it comes to having a positive effect on G3.
df_variables['b_paidxtraclasses'] = df_variables['paid'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_paidxtraclasses'].value_counts()

# 0 = no and 1 = yes
df_variables['b_xtraactivities'] = df_variables['activities'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_xtraactivities'].value_counts()

# 0 = no and 1 = yes
# It has a high correlation however, we only have 20 students that are not interested in having a high education and # thus this
column should not be taken into consideration.
df_variables['b_higher_education'] = df_variables['higher'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_higher_education'].value_counts()


# continue with the analisis.
df_variables['b_internet'] = df_variables['internet'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_internet'].value_counts()

# Interestingly when people are not in a romantic relationship they tend to get better grades.
df_variables['b_romantic'] = df_variables['romantic'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_romantic'].value_counts()

#df_variables['b_nursery'] = df_variables['nursery'].apply(lambda x: 0 if x == 'no' else 1)
#df_variables['b_nursery'].value_counts()

df_variables['b_guardian'] = df_variables['guardian'].apply(lambda x: 0 if x == 'mother' else (1 if x=='father' else 2))
df_variables['b_guardian'].value_counts()

# Does not have any effect on G3. Low correlation.
#df_variables['b_reason'] = df_variables['reason'].apply(lambda x: 0 if x == 'home' else (1 if x=='reputation' else (3 if
x=='course' else 4)))
#df_variables['b_reason'].value_counts()
```

```
# Does not have any effect on G3. Low correlation.
#df_variables['b_school'] = df_variables['school'].apply(lambda x: 0 if x == 'GP' else 1)
#df_variables['b_school'].value_counts()

df_variables['b_schoolsup'] = df_variables['schoolsup'].apply(lambda x: 0 if x == 'no' else 1)
df_variables['b_schoolsup'].value_counts()

#variable Mjob
df_variables['b_Mjob'] = df_variables['Mjob'].apply(lambda x: 0 if x == 'nominal' else (1 if x=='health' else (2 if x=='services'
else (3 if x=='at home' else 4))))
df_variables['b_Mjob'].value_counts()

#variable Fjob
df_variables['b_Fjob'] = df_variables['Fjob'].apply(lambda x: 0 if x == 'nominal' else (1 if x=='health' else (2 if x=='services'
else (3 if x=='at home' else 4))))
df_variables['b_Fjob'].value_counts()

df_variables_new=df_variables.drop(columns=['famsize','paid','activities','higher','internet',
                                    'romantic','guardian','schoolsup','Mjob','Fjob','famrel'])

df_variables_new.info()

#normalizamos los valores para que se pongan entre los mismos rangos
#ya que los valores estan muy distintos
df_norm=(df_variables_new-df_variables_new.min())/(df_variables_new.max()-df_variables_new.min())
df_norm

df_norm.describe()


#implementaremos el método codo de jambu
#crea diferencia de tipos de clustering para ver qué tan similares son los vecinos
#e irlos mostrando o plasmándolos dentro de una gráfica.

#wcss es la suma de los cuadrados de cada grupo


arreglowcss = []#variable para almacenar

for i in range (1,11):#loop para crear agrupaciones se pone hasta cual numero quieres +1
   kmeans = KMeans(n_clusters = i, max_iter=300)
   kmeans.fit(df_norm) #aplicamos K/means a la base de datos
   arreglowcss.append(kmeans.inertia_)

plt.plot(range(1,11), arreglowcss)
plt.title('codo de Jambu')
plt.xlabel('Numero de clusters')
plt.ylabel('WCSS')#indicador de que tan similares son los individuos dentro de los clústers
plt.show()

#aplicamos el metodo Kmeans al BD

clustering = KMeans(n_clusters = 4, max_iter = 300)#creamos el modelo
clustering.fit(df_norm) #aplicamos el modelo al BD
```

```
#agregamos la clasificación al archivo original

df['KMeans_Clusters'] = clustering.labels_ #los resultados se guardan en label_ dentro del modelo
df.head()


#visualización de los clustering que se formaron
#utilizando gráficos con análisis de componentes principales PCA

from sklearn.decomposition import PCA

pca = PCA(n_components=2) #modelo de 2 dimensiones
pca_df = pca.fit_transform(df_norm) #obtenemos los dos componentes principales
pca_df_data = pd.DataFrame(data = pca_df, columns = ['Componente_1', 'Componente_2']) #Creamos dataframe que contega los elementos principales
pca_nombres_df = pd.concat([pca_df_data, df[['KMeans_Clusters']]], axis=1) #agregamos la columna del clustering

pca_nombres_df
print(pca_nombres_df.query('KMeans_Clusters == 1'))
pca_nombres_df.to_csv('C:/Users/Admin/Documents/PrediccionDeCalificcionesSecundaria/clusters_creados/MetodoPCA1.csv')

#coloreamos los clustering para diferenciar mejor

fig = plt.figure(figsize = (6,6)) #tamaño de la figura

ax = fig.add_subplot(1,1,1)  #creamos solo 1 grafico
ax.set_xlabel('Componente_1', fontsize = 15)
ax.set_ylabel('Componente_2', fontsize = 15)
ax.set_title('Componentes Principales', fontsize = 20)

color_theme = np.array(["blue", "pink", "purple", "red"])
ax.scatter(x = pca_nombres_df.Componente_1, y = pca_nombres_df.Componente_2,
        c=color_theme[pca_nombres_df.KMeans_Clusters], s = 50)
plt.show()

#por ultimo guardamos los clústers dentro de nuestro disco duro
df.to_csv('C:/Users/Admin/Documents/PrediccionDeCalificcionesSecundaria/clusters_creados/Clusterde4(1).csv')
```

Anexo 4. Código modelo redes neuronales

```
# -*- coding: utf-8 -*-
"""

Red neuronal ejempló 2

Tomado de: https://www.cienciadedatos.net/documentos/py35-redes-neuronales-python.html

Redes neuronales con Python

Joaquín Amat Rodrigo

Mayo, 2021
```

```
Fecha: 19 enero 2022
Pero aplicado al dataset: student-mat.csv
Y con algunas modificaciones de: Leonardo Nevarez Chávez
@author: Valeria Avila Grajeola
"""
# Tratamiento de datos
# ==============================================================================
import numpy as np
import pandas as pd
# Gráficos
# ==============================================================================
import matplotlib.pyplot as plt
import seaborn as sns
#%matplotlib inline
plt.style.use('fivethirtyeight')
# Modelado
# sklearn
# Librería de Machine Learning para Python
# Análisis predictivo: Clasificación, Regresión, Clustering,
#    Reducción dimensionalidad, Selección modelos,
#    Preprocesamiento
# ==============================================================================
# Multi-layer Perceptron regressor.
#  This model optimizes the squared error using LBFGS or stochastic gradient descent.
from sklearn.neural_network import MLPRegressor
# ColumnTransformer: Applies transformers to columns of an array or pandas DataFrame.
from sklearn.compose import ColumnTransformer
# OneHotEncoder: Encode categorical features as a one-hot numeric array.
from sklearn.preprocessing import OneHotEncoder
# StandardScaler: Standardize features by removing the mean and scaling to unit variance.
from sklearn.preprocessing import StandardScaler
# make_column_selector: Create a callable to select columns to be used with ColumnTransformer.
from sklearn.compose import make_column_selector
# Pipeline:  Pipeline of transforms with a final estimator.
from sklearn.pipeline import Pipeline
# metrics mean_squared_error: Mean squared error regression loss.
from sklearn.metrics import mean_squared_error
# model_selection  RandomizedSearchCV: Randomized search on hyper parameters.
from sklearn.model_selection import RandomizedSearchCV
```

```
# model_selection KFold: Provides train/test indices to split data in train/test sets.
#  Split dataset into k consecutive folds (without shuffling by default).
from sklearn.model_selection import KFold
# sklearn set_config: Set global scikit-learn configuration
from sklearn import set_config
# multiprocessing: multiprocessing is a package that supports spawning processes
# using an API similar to the threading module.
import multiprocessing
# Configuración warnings
# ==============================================================================
# python warnings: Warning messages are typically issued in situations where it is
# useful to alert the user of some condition in a program.
import warnings
warnings.filterwarnings('ignore')
# Descarga de datos
# ==============================================================================
#url = ("../student-mat.csv")
datos = pd.read_csv("../datos/student-mat.csv", sep=","
# Se renombran las columnas para que sean más descriptivas
datos.columns = ["escuela", "genero", "edad", "direccion",
                 "tamanio_familia", "vive_con_padres", "Madre_educacion",
                 "Padre_educacion", "Madre_trabajo", "Padre_trabajo", "razon_elegir_escuela",
                 "tutor", "tiempo_translado", "tiempo_estudio",
                 "materias_reprobadas", "desea_continuar_estudios_sup",
                 "familia_soporte", "recibe_pago", "actividades",
                 "nursery", "higher", "internet", "relacion_romantica",
                 "calidad_relacion_familiar", "tiempo_libre", "sale_fuera",
                 "consumo_alcohol_entre_semana", "consumo_alcohol_finsemana",
                 "estado_salud", "faltas", "G1", "G2", "G3"
                 ]
print("Descripción general de los datos")
print(datos.info())
# Distribución variable respuesta
# Histogramas de las variables G1, G2 y G3
# ==============================================================================
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G3', kde=True,ax=ax)
ax.set_title("Distribución G3")
ax.set_xlabel('G3');
```

```
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G2', kde=True,ax=ax)
ax.set_title("Distribución G2")
ax.set_xlabel('G2');
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(6, 3))
sns.histplot(data=datos, x='G1', kde=True,ax=ax)
ax.set_title("Distribución G1")
ax.set_xlabel('G1');
# Gráfico de distribución para cada variable numérica
# Considerar la cantidad de variables numéricas para configurar correctamente los
# parámetros nrows y ncols, que definen las filas y columnas para gráficas
# ======================================================================
#fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(12, 7))
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(12, 7))
axes = axes.flat
# Se obtienen las variables numéricas y se omiten G1, G2 y G3
columnas_numeric = datos.select_dtypes(include=['float','int64']).columns
columnas_numeric = columnas_numeric.drop('G1')
columnas_numeric = columnas_numeric.drop('G2')
columnas_numeric = columnas_numeric.drop('G3')
# Ciclo para obtener histograma de cada variable numérica
for i, colum in enumerate(columnas_numeric):
   sns.histplot(
      data    = datos,
      x       = colum,
      stat    = "count",
      kde     = True,
      color   =  "blue",
      #color   = (list(plt.rcParams['axes.prop_cycle'])*2)[i]["color"],
      line_kws= {'linewidth': 2},
      alpha   = 0.3,
      ax      = axes[i]
   )
   axes[i].set_title(colum, fontsize = 7, fontweight = "bold")
   axes[i].tick_params(labelsize = 6)
   axes[i].set_xlabel("")
   axes[i].set_ylabel("")
fig.tight_layout()
plt.subplots_adjust(top = 0.9)
```

```
fig.suptitle('Distribución variables numéricas', fontsize = 10, fontweight = "bold");
# Obtener histogramas sencillos de las variables numéricas
import matplotlib.pyplot as plt
datos.hist(bins=50, figsize=(20,15), color='r')
plt.show()
# Gráfico para cada variable cualitativa
# ==============================================================================
#fig, axes = plt.subplots(nrows=6, ncols=3, figsize=(12, 5))
fig, axes = plt.subplots(nrows=6, ncols=3, figsize=(24, 10))
axes = axes.flat
columnas_object = datos.select_dtypes(include=['object']).columns
for i, colum in enumerate(columnas_object):
    datos[colum].value_counts().plot.barh(ax = axes[i])
    axes[i].set_title(colum, fontsize = 14)
    axes[i].set_xlabel("")
# Se eliminan los axes vacíos
#for i in [7, 8]:
#    fig.delaxes(axes[i])
fig.tight_layout()
fig.savefig('../imagenes/redneuronalGraficaVariablesNumericas.png')
# Con el objetivo de poder estimar el error que comete el modelo al predecir
# nuevas observaciones, se dividen los datos en dos grupos, uno de entrenamiento
# y otro de test (80%, 20%).
# Reparto de datos en train y test
# ==============================================================================
# sklearn.model_selection.train_test_split
# Split arrays or matrices into random train and test subsets.
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
                                    datos.drop('G3', axis = 'columns'),
                                    datos['G3'],
                                    train_size   = 0.8,
                                    random_state = 1234,
                                    shuffle      = True
                                )
# Obtener los registros de X_test
print("X_test")
print(X_test)
X_test.to_csv('x_test.csv')
```

```
# Obtener valor G3 del conjunto de prueba, para comparar con predicciones
print("y_test")
print(y_test)
y_test.to_csv('y_test.csv')
#Tras realizar el reparto, se verifica que los dos grupos son similares, en cuanto a
#estadísticas básicas: promedios, máximos, mínimos, desviación estándar, etc.
print("Partición de entrenamiento")
print("-----------------------")
print(y_train.describe())
print(X_train.describe())
print(X_train.describe(include = 'object'))
print(" ")
print("Partición de test")
print("-----------------------")
print(y_test.describe())
print(X_test.describe())
print(X_test.describe(include = 'object'))
# Los modelos de redes neuronales requieren como mínimo de dos tipos de preprocesado:
# binarización (One hot ecoding) de las variables categóricas y estandarización de las
# variables continuas.
# Selección de las variables por tipo
# ==============================================================================
# Se estandarizan las columnas numéricas y se hace one-hot-encoding de las
# columnas cualitativas. Para mantener las columnas a las que no se les aplica
# ninguna transformación se tiene que indicar remainder='passthrough'.
# Identificación de columnas numéricas y categóricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int64']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()
# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )
# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore'))]
                          )
preprocessor = ColumnTransformer(
                    transformers=[
                      ('numeric', numeric_transformer, numeric_cols),
```

```
                    ('cat', categorical_transformer, cat_cols)
                ],
                remainder='passthrough'
            )
set_config(display='diagram')
preprocessor
print("Datos pre-procesados")
print(preprocessor)
set_config(display='text')
# Se aprenden y aplican las transformaciones de preprocesado
# ==============================================================================
X_train_prep = preprocessor.fit_transform(X_train)
X_test_prep  = preprocessor.transform(X_test)
# Convertir el output en dataframe y añadir el nombre de las columnas
# ==============================================================================
encoded_cat = preprocessor.named_transformers_['cat']['onehot']\
            .get_feature_names(cat_cols)
labels = np.concatenate([numeric_cols, encoded_cat])
datos_train_prep = preprocessor.transform(X_train)
datos_train_prep = pd.DataFrame(datos_train_prep, columns=labels)
datos_train_prep.info()
# Mostrar resultado de preprocesado y nombres de columnas
print("Datos preprocesados y nombres de columnas:")
print(datos_train_prep)
print("Descripción de datos preprocesados:")
print(datos_train_prep.info())
# Modelado
# Pipepeline de preprocesado + modelado
# Pipeline de preprocedado y modelado
# ==============================================================================
# Identificación de columnas numéricas y categóricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int64']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()
# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                    steps=[('scaler', StandardScaler())]
                )
# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
```

```
                    steps=[('onehot', OneHotEncoder(handle_unknown='ignore'))]
                   )
preprocessor = ColumnTransformer(
                transformers=[
                   ('numeric', numeric_transformer, numeric_cols),
                   ('cat', categorical_transformer, cat_cols)
                ],
                remainder='passthrough'
            )
# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline
pipe = Pipeline([('preprocessing', preprocessor),
             ('modelo', MLPRegressor(solver = 'lbfgs',
                                     max_iter= 10000))])
# Espacio de búsqueda de cada hiperparámetro
# ==============================================================================
param_distributions = {
   'modelo__hidden_layer_sizes': [(10), (20), (10, 10)],
   'modelo__alpha': np.logspace(-3, 3, 10),
   'modelo__learning_rate_init': [0.001, 0.01],
}
# Búsqueda por validación cruzada
# ==============================================================================
grid = RandomizedSearchCV(
       estimator  = pipe,
       param_distributions = param_distributions,
       n_iter     = 10,
       scoring    = 'neg_mean_squared_error',
       n_jobs     = multiprocessing.cpu_count() - 1,
       cv         = 5,
       verbose    = 0,
       random_state = 123,
       return_train_score = True
      )
grid.fit(X = X_train, y = y_train)
# Resultados del grid
# ==============================================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param.*|mean_t|std_t)')\
   .drop(columns = 'params')\
```

```
    .sort_values('mean_test_score', ascending = False)\
    .head(10)
# Resultados después de procesado
print("Resultados despues de proceso:")
print(resultados)
resultados.to_csv("resultados_proceso_redneuronal.csv")

# Error de test
# ================================================================================
modelo_final = grid.best_estimator_
predicciones = modelo_final.predict(X = X_test)
rmse = mean_squared_error(
        y_true = y_test,
        y_pred = predicciones,
        squared = False
       )
print('Error de test (rmse): ', rmse)
# Verificar este código y cálculo. Agregado por mi, pero falta validar
# score obtiene el coeficiente de determinación de la predicción
print("Score")
print(modelo_final.score(X_test, y_test))
print("Predicciones:")
print(predicciones)
# Guardar arreglo numpy de prediccion de calificación G3 como csv
from numpy import savetxt
savetxt('prediccionesG3.csv', predicciones, delimiter=',')
# Conclusión
# La combinación de hiperparámetros con la que se obtienen
# mejores resultados acorde a las metricas de validación cruzada es:
modelo_final['modelo'].get_params()
print("Modelo final:")
print(modelo_final['modelo'].get_params())
# Intentar obtener "accuracy" de la red neuronal
"""
Este código genera el error:
Classification metrics can't handle a mix of multiclass and continuous
targets
Creo debido que la variable a predecir G3 es una variable continua
from sklearn.metrics import accuracy_score
```

```
print("Accuray score:")
print(accuracy_score(y_test,predicciones))
"""
# Información de sesión
#from sinfo import sinfo
#sinfo()
# Intentar obtener gráficas del modelo obtenido
# Mostrar grafica de regresion
figura = plt.figure(figsize = (15, 15))
plt.scatter(y_test, y_test, color = "blue", marker = "*", s = 30)
 # plotting the regression line
plt.scatter(y_test, predicciones, color = "green")
# putting labels
plt.xlabel('Calificacion')
plt.ylabel('Calificacion')
plt.title('Red neuronal, calificaciones reales y predicciones')
plt.show()
figura.savefig('../imagenes/redneuronal.png')

# Otra gráfica
plt.plot(y_test)
plt.plot(predicciones)
plt.show()
# Grafica de barras entre calificaciones G3 reales y predicciones
X = X_test
Yreal = y_test
Ypredicciones = predicciones
X_axis = np.arange(len(X))
figura = plt.figure(figsize = (20, 20))
plt.barh(X_axis - 0.2, Yreal, 0.4, label = 'real')
plt.barh(X_axis + 0.2, Ypredicciones, 0.4, label = 'predicción')
#plt.xticks(X_axis, X)
plt.xlabel("Calificación")
plt.ylabel("Estudiantes")
plt.title("Calificaciones G3 reales y predicciones")
plt.legend()
plt.show()
figura.savefig('../imagenes/redneuronalGraficaBarras.png')
```