TECNOLÓGICO NACIONAL DE MÉXICO
Instituto Tecnológico Superior de Teziutlán

“MODELO DE APRENDIZAJE AUTOMÁTICO PARA EL ANÁLISIS PREDICTIVO DE LAS HABILIDADES BÁSICAS DEL PENSAMIENTO”

TESIS QUE PRESENTA:

Jerónimo Aparicio Juárez

Como requisito parcial para obtener el título de:

MAESTRO EN SISTEMAS COMPUTACIONALES

Fracción I y II S/N, Aire Libre, Teziutlán, Puebla.
Agosto de 2020.

PÁGINA DE JURADO

La presente tesis titulada: **Modelo de Aprendizaje Automático para el análisis predictivo de las Habilidades Básicas del Pensamiento**, fue realizada bajo la dirección del comité de asesores indicado, ha sido aprobada por el mismo y aceptada como requisito parcial para obtener el título de:

MAESTRO EN SISTEMAS COMPUTACIONALES

DIRECTOR:

DRA. JULIETA DEL CARMEN VILLALOBOS ESPINOSA _____

1er. CO-DIRECTOR:

M.S.C. JACOBO ROBLES CALDERÓN _____

2do. CO-DIRECTOR:

M.S.C. HÉCTOR VICENTEÑO RIVERA _____

Fracción I y II S/N, Aire Libre, Teziutlán, Puebla. Mayo de 2020.

AGRADECIMIENTOS

El autor expresa su más sincero agradecimiento a la Supervisión Escolar de la Zona 058 de Educación Básica Estatal perteneciente a la Secretaría de Educación de Veracruz, así como al personal directivo, administrativo y docente por las facilidades otorgadas para la realización del presente proyecto de investigación.

DEDICATORIA

A mi madre y padre por ser los dos pilares más importantes de mi educación y formación a lo largo de toda mi vida, a mis amigos por confiar en mi desarrollo profesional y a Dios por acompañarme siempre en el recorrido de esta divertida aventura del conocimiento al encuentro de la verdad.

RESUMEN

El presente trabajo de investigación abordó un subcampo de las Ciencias de la Computación y a su vez una rama de la Inteligencia Artificial denominado Aprendizaje Automático. Se aplicaron técnicas de minería de datos con la finalidad de tomar decisiones con base en la predicción de los factores que inciden en el dominio de las Habilidades Básicas del Pensamiento, para extraer conocimiento de la información se examinaron diferentes colecciones de datos con la finalidad de parametrizar un modelo que permitiera resolver la problemática de las causas que inciden en el bajo rendimiento escolar. En la etapa de análisis, se adquirió y preparó los datos para explorarlos, modelizar y evaluar por medio de un árbol de decisión e implementar el clasificador multiclase Dummy Classifier para posteriormente compararlo con Regresión Logística (L1 LibLinear y ElasticNet Saga), Random Forest y XGBoost, finalmente se evaluó la calidad de entrenamiento y la predicción de nuevos ejemplos no etiquetados obtenidos desde una fuente de datos independiente.

Palabras clave: árboles, automático, aprendizaje, clasificadores, multiclase

ABSTRACT

The curent resarch approached a subfield of Computer Science and at the same time a branch of Artificial Inteligance known as Machine Learning. Data mining techniques were aplied with the purpose of making decisions based on the prediction of factors that fall upon the domain of the Basic Skills of Thinking, to extract knowledge from the information, different data collections were examined with the purpose of giving parameteres to models that will allow us to resolve the problem that cause an effect in the low scholar performance. In the analysis stage, the data was acquired and prepared for exploration, modeled and evaluated through a tree of decisions and impliment the classifiers multiclass Dummy Classifier to later compare it with Logistic Regression (L1 LibLinear and Elastic Saga), Random Forest and XGBoost, finally the quality of training was evaluated and the prediction of new unlabeled examples obtained from an independent data source.

Keywords: trees, machine, learning, classifiers, multiclass

ÍNDICE GENERAL

LISTADO DE TABLAS

LISTADO DE FIGURAS

LISTADO DE ECUACIONES

CAPÍTULO I: GENERALIDADES DEL PROYECTO

1.1 INTRODUCCIÓN

En la actualidad las tendencias tecnológicas propician nuevas formas de darle tratamiento a los datos, es por esta razón que hoy en día no es suficiente solo almacenar colecciones de datos para su análisis, sino que se requiere realizar predicciones que permitan alertar de manera oportuna y precisa situaciones críticas aprovechando los avances tecnológicos y científicos, así mismo, es necesario descubrir hechos contenidos en las bases de datos sin ninguna intervención del humano, pero con el objetivo de servir de apoyo para la toma de decisiones.

En el contexto descrito anteriormente, el objetivo del proyecto dar solución a una problemática presente en el dominio de las habilidades básicas del pensamiento en educación básica las cuales inciden en índices de reprobación, bajo rendimiento y deserción escolar por situaciones poco conocibles, pero si observables en los resultados obtenidos, los cuales solo pueden predecirse con la intervención de un humano, pero de forma no eficaz. Por esta razón, se creará un aplicativo automatizado que muestre predicciones antes de que los eventos sucedan.

La línea de investigación que se abordó para tratar la problemática planteada es denominada Aprendizaje Automático, con ella se desarrolló nuestro modelo de predicción, así como también, se aplicaron técnicas de minería de datos con el objetivo de canalizar métodos estadísticos rudimentarios en los procesos que actualmente presentan problemas de eficiencia y escalabilidad.

"Las 'leyes del pensamiento' no solo dependen de las propiedades de las células cerebrales, sino del modo en que están conectadas"

- Marvin Lee Minsky[1], científico estadounidense.

[1] Marvin Lee Minsky, es considerado uno de los padres de la inteligencia artificial. Fue cofundador del laboratorio de inteligencia artificial del Instituto Tecnológico de Massachusetts o MIT.

1.2 PLANTEAMIENTO DEL PROBLEMA

Las escuelas de educación básica presentan un problema para la toma de decisiones de forma oportuna en lo referente a las habilidades básicas del pensamiento en los alumnos, las cuales están comprendidas en habilidades de escritura, lectura y matemáticas.

Para coleccionar datos sobre esas problemáticas la Secretaría de Educación Pública pone a su alcance el Sistema de Alerta Temprana (SiSAT) en el cual el cuerpo docente, administrativo y directivo suministra información sensible que posteriormente es proyectada en indicadores y gráficas en el sistema, sin embargo, este no realiza ninguna predicción oportuna sobre los datos conocibles, sino que, por el contrario, en algunos casos presenta inconsistencias en:

- La graficación visual de los datos
- Las escuelas presentan inconformidad en la veracidad del valor numérico que se asigna al alumno en determinado componente de lectura y escritura.

Debido a lo anterior, las problemáticas tratadas en los Consejos Técnicos Escolares recaen en los temas de:

- Índices de reprobación
- Cumplir con la normalidad mínima
- Mejorar el resultado de las evaluaciones de SiSAT el cual no permite predecir los siguientes rubros:
  - Identificar los factores que inciden en el avance y retroceso del alumno en la lectura.
  - Identificar los factores que dificultan la adquisición del cálculo mental.
  - Identificar los factores que interfieren en la redacción de textos.

Por lo tanto, se requiere el desarrollo de un modelo predictivo que pueda ser integrado a un sistema en producción que permita el manejo de la información de forma eficiente para lograr un resultado eficaz que dé solución a lo descrito anteriormente.

1.3 JUSTIFICACIÓN

El modelo de Aprendizaje Automático que se desarrolló a continuación permitirá dar predicciones con respecto a los procesos de adquisición de la lectura, escritura y cálculo mental que en la actualidad no existen como herramienta de apoyo en las instituciones de educación básica, al implementar este modelo los colegios podrán ejecutar un plan de contingencia basado en resultados significativos, pertinentes, factibles y oportunos.

El impacto social del modelo propiciará un beneficio colectivo para los alumnos y el centro escolar que se verá reflejado en el aumento de la calidad educativa del plantel a través del cambio en las metodologías de la práctica docente actual por la implementación de estrategias e instrumentación didáctica innovadoras que permitan al alumno desarrollarse en un entorno de aprendizaje favorable y armónico. Así mismo, se eliminará el uso exagerado de documentos en papel debido a que la aplicación del modelo en un sistema integral ofrecerá movilidad vía web y móvil para su implementación en producción.

El sistema permitirá automatizar procesos predictivos de forma rápida y confiable apoyando a la solución de problemas prácticos presentes en el centro escolar, haciendo uso de algoritmos de inteligencia artificial que han marcado las últimas tendencias tecnológicas. Así como también, será adaptable al nuevo modelo educativo.

Al mitigar las problemáticas que influyen en el proceso de adquisición de las habilidades básicas del pensamiento, el sistema aportará beneficios a la Ruta de Mejora Escolar (RME), no solo de manera local, sino nacional en caso de ser aceptada como propuesta factible en los centros educativos del sistema básico a través de la Secretaría de Educación Pública.

El aprendizaje automático y minería de datos aplicados en la implementación del modelo dará como resultados predicciones anticipadas y confiables que permitirán generalizar comportamientos a través de ejemplos que con los métodos rudimentarios no se podría dar solución de forma inmediata; permitiendo realizar un estudio preciso de los alumnos con bajo rendimiento escolar que se encuentra en estado crítico.

1.4 HIPÓTESIS

Variable independiente

- Indicador de rendimiento del alumno.

Variable dependiente

- Definir las necesidades de acciones de forma anticipada.

Hipótesis resultante

- Es posible determinar el indicador de rendimiento de un alumno utilizando de forma predictiva un modelo de aprendizaje automático para definir las necesidades de acciones de forma anticipada.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Desarrollar un modelo de predicción de las Habilidades Básicas del Pensamiento, para conocer sus factores de incidencia en los colegios de nivel básico, lo cual permita tomar decisiones para crear una Ruta de Mejora Escolar dinámica para qué anticipadamente los colegios puedan realizar una intervención pedagógica para tratar los resultados de predicción y evitar que un grupo llegue a un estado crítico que sea más difícil de tratar y genere más costos.

1.5.2 OBJETIVOS ESPECÍFICOS

- Predecir las Habilidades Básicas del Pensamiento (lectura, cálculo mental, redacción de textos).
- Representar de forma gráfica y predictiva los factores que inciden en el avance y retroceso en la lectura, dificultad en el cálculo mental y redacción de textos.
- Visualizar los factores más importantes que afectan las Habilidades Básicas del Pensamiento

1.6 ALCANCES

- El modelo tendrá la capacidad de predecir el dominio de las habilidades básicas del pensamiento desde el ámbito pedagógico para intervenir oportunamente en la modificación de aquellos factores que lo requieran, ya que la mayor parte de estos instrumentos de diagnóstico se usan en las escuelas, pero no hay un seguimiento institucional desde el sistema educativo nacional que permita predecir aquellos alumnos vulnerables y realizar los ajustes necesarios para brindarle la oportunidad de concluir su educación básica.
- Favorecer el acceso a la información de manera oportuna y consultarla desde diversas instancias (docente, director, supervisor).
- Eliminar la burocracia y carga administrativa permitiendo un sistema de interacción de la información concentrándola en un solo espacio.

1.7 LIMITACIONES Y DELIMITACIONES

Esta investigación se realiza con datos para escuelas primarias que se encuentran en el municipio de Tlapacoyan. Ubicado en el centro del estado Veracruz, en la región llamada del Nautla. Es uno de los 212 municipios de la entidad. Está ubicado a una altura de 430 msnm.

Los datos de la presente investigación se basan en los instrumentos utilizados por el personal de educación básica en el nivel primaria en los ciclos escolares 2017-2018 y 2018-2019.

La población en la que se aplica esta investigación es en alumnos de nivel primaria, aunque puede extenderse a nivel preescolar o secundaria pues comparten el perfil de egreso de educación básica.

- La deserción escolar es un problema que aqueja a todo el sistema educativo, sin embargo, no existe una estrategia o instrumento que permita al centro educativo predecir que alumnos están en riesgo y las causas que lo originan.

- Los factores que se consideran viables de investigar son: la asistencia, participación, convivencia, entrevistas a padres, entrevistas a alumnos (en estas identificar la situación económica), ficha de los alumnos (estado de salud del alumno), resultados de SiSAT, promedios, madurez intelectual, estilos de aprendizaje, así mismo, con respecto al docente, la metodología empleada y sus cuadrantes cerebrales.

- El modelo no incluirá instrumentos de diagnóstico para la recolección de datos, sino que será un concentrado general de dichos instrumentos para su posterior predicción.

1.8 ESTADO DEL ARTE

En los últimos años la ciencia y tecnología han avanzado considerablemente, los primeros inicios del aprendizaje automático surgen desde el inicio del Perceptrón creado por el psicólogo norteamericano Frank Rosenblatt en el año de 1958 y el cual hace mención a la unidad básica de inferencia, siendo este el primer algoritmo que presentaba una red neuronal simple, Rosenblatt se basó en las ideas presentadas por McCulloch y Pitts en el cual se hablaba por primera vez de la posibilidad de crear redes neuronales como si fueran un ordenador, a partir de entonces, han sido muchas las investigaciones que han hecho esfuerzos agigantados por aplicar modelos de inteligencia artificial a diferentes campos de la ciencia. [1]

Una rama de la inteligencia artificial que ha tenido mucho éxito es el aprendizaje automático, por tal motivo, el área de la educación le ha puesto como una solución esencial de aplicación para la ruta de mejora escolar y con ello resolver diferentes problemáticas que agobian a este sector. Ejemplo de ello, son las investigaciones que se presentan a continuación:

Un análisis comparativo de las técnicas de aprendizaje automático para la gestión de la retención de estudiantes.

Delen en el 2010, reportó que “los métodos de minería de datos son capaces de predecir las tasas de deserción de estudiantes de primer año con aproximadamente un 80% de precisión cuando los datos son suficientes y se seleccionan las variables adecuadas". [2] Denle, comenta que la inteligencia artificial (IA) es una herramienta que permite de manera efectiva la retención de estudiantes.

Por lo tanto, este artículo de investigación, parece apoyar la idea de que la Inteligencia Artificial (IA) se puede usar de manera efectiva para predecir la retención de estudiantes. Esta conclusión está respaldada por el hecho de que varias universidades ya están utilizando IA de esta manera, incluida la Universidad de Derby en Inglaterra.

El objetivo era “Prevenir la deserción en la educación superior, a través del aprendizaje automático”. Para cumplir tal objetivo se basó en una muestra de 40,000 estudiantes.

Delen en el 2010 reporta que “la mejora de la retención estudiantil debe comenzar por un profundo conocimiento de los modos de deserción. El comprender esto es el pilar para predecir qué estudiantes están en riesgo de dejar sus estudios, e intervenir de manera apropiada, de modo de retenerles."

Delen concluyó que, en su muestra, las variables educativas y financieras eran los principales predictores de deserciones.

Este tipo de estudios tienen ya varias aplicaciones en la práctica. Con un algoritmo apropiado, sumados a la entrega adecuada de datos, el Machine Learning permite:

- Detectar tempranamente a los estudiantes con alto riesgo de deserción.
- Identificar los factores de riesgo más recurrentes.
- Entregar informes de las principales causas y factores de riesgo.

Una pregunta que parte del estudio que realizó el Dr. Delen es: ¿Cómo ayuda el Machine Learning a prevenir la deserción escolar en educación superior? A través del aprendizaje continuo de las tecnologías de la información de los patrones que generan los grandes datos o big data.

El aprendizaje automático en la práctica: Desarrollo de investigación

A continuación, se muestra un ejemplo práctico: [2]

Tomemos el caso de Pedro:

1. Fue aceptado en Ingeniería Mecánica, en una de las mejores universidades de México, a la que asisten más de 40.000 estudiantes.
2. En su postulación, destacó que esta era su segunda opción, luego de que fuera rechazado para entrar a Ingeniería en Computación en la misma casa de estudios.
3. Demostró ser un estudiante destacado, con notas que le permitieron adjudicarse una media beca. Incluso, los registros de la biblioteca ratificaban que pasaba mucho tiempo buscando libros de reserva.

Sin embargo, desde el segundo año de universidad, sucedió algo:

1. Comenzó a faltar a clases, poco antes que empezaran los fines de semana.
2. De acuerdo a sus registros, comenzó a vivir en una residencia estudiantil, puesto que su hogar estaba a 700km de la casa central de la universidad.
3. Sus registros financieros consignaban que estudiaba gracias a un préstamo bancario obtenido por sus padres, pero cuyos pagos estaban retrasados.

En ese sentido, había peligro de que Pedro dejara la escuela por motivos financieros y geográficos.

De más de 5.000 registros de estudiantes en estado de riesgo, y antes de empezar el tercer año de estudios, la Dirección de Asuntos Estudiantiles recibió una alerta de la situación, por cuanto Pedro todavía no había inscrito ramo alguno, y había faltado a varios cursos de nivelación de verano correspondientes a su carrera.

Un orientador académico le llamó a su oficina.

- Sí, era cierto. La familia de Pedro pasaba por problemas financieros, y estaba trabajando los fines de semana – viernes incluido – solo para pagarse la residencia.
- Efectivamente, estaba pensando en dejar sus estudios.

Poco después, la universidad le presentó una opción. No era conveniente seguir estudiando Ingeniería Mecánica en la sede central. Sin embargo, hacía algunos años

que la casa de estudios había abierto un nuevo campus a pocos kilómetros de su ciudad natal. Una de las especialidades que impartía era Ingeniería en Computación.

Aunque la universidad le había rechazado en su primera postulación, con su registro académico, Pedro había demostrado méritos suficientes para realizar un cambio interno de carrera, y podía postular por adelantado al programa local.

Ningún orientador académico tenía idea de los detalles completos del registro de Pedro, porque era confidencial. Solo obtuvieron un resumen luego de que el sistema les alertara de que era un alumno en grave riesgo de dejar sus estudios.

En el ejemplo de Machine Learning aplicado a la deserción escolar en el nivel superior, se tuvo como resultados que por cuanto estaba configurado para detectar ese tipo de patrones, y la evidencia académica comparada había concluido que un estudiante que cambiara de manera tan dramática sus hábitos de estudio estaba en riesgo de deserción.

Muchas de estas soluciones o negociaciones ya existen en las universidades. Sin embargo, muchas instituciones de educación superior carecen de la capacidad de análisis necesario para reaccionar en tiempo a situaciones que las lleven a tener pérdida de estudiantes.

Análisis de deserción-permanencia de estudiantes universitarios utilizando técnica de clasificación en minería de datos

K. B. Eckert y R. Suénaga en el año 2015, realizaron un estudio de clasificación con técnicas de minería de datos para analizar la deserción-permanencia de estudiantes universitarios. [3]

Analizaron información académica con el objetivo de identificar factores que influyen sobre la deserción de los estudiantes de la carrera de Ingeniería en Informática de la Universidad Gastón Dachary en Argentina, mediante la aplicación de una técnica de minería de datos. La fuente de datos contiene información proporcionada al ingreso (personales y antecedentes educativos) y la que se genera durante el periodo de estudios. Se realiza la selección y depuración de datos, utilizando diferentes criterios de representación y aplicación de algoritmos de clasificación como árboles de decisión, redes bayesianas y

reglas. Se identifica como variables influyentes en la deserción, asignaturas aprobadas, cantidad y resultado de asignaturas cursadas, procedencia y edad de ingreso del estudiante. Mediante este proceso fue posible identificar los atributos que caracterizan a los casos de deserción y su relación con el desempeño académico, especialmente en el primer año de la carrera.

La muestra sobre la cual se trabajó correspondía a los estudiantes de la carrera de Ingeniería en Informática, modalidad presencial, 5 años de duración y la tesis de grado. El período seleccionado para el estudio corresponde a los estudiantes de periodo 2000 al 2009, lo que produjo un total de 855 casos analizados.

El atributo para determinar los casos de deserción, es decir si los estudiantes abandonan o no sus estudios es de tipo dicotómico, deserta (“Des”) y no deserta (“NoDes”). Para ello se tomó como atributo la condición final, de tipo nominal, que puede adoptar uno de los cuatro valores posibles: egresado, en curso, baja temporal y baja definitiva. La baja definitiva indica que el alumno ha desertado de la carrera (“Des”) y las demás condiciones, para su procesamiento, fueron agrupadas como no deserción (“NoDes”); cabe aclarar que la baja temporal hace referencia a la suspensión temporal de la actividad del alumno. Como rendimiento académico se considera el grado de éxito de los estudiantes, relacionado con la obtención de buenas calificaciones, escasos exámenes reprobados, pocos o ninguna materia re-cursada, cursado y aprobación sin retraso respecto al plan de estudios de la carrera. La Tabla 1, muestra los atributos predictores y el tipo de dato al que pertenecen.

Tabla 1. Atributos seleccionados y estandarizados [3]
Fuente: https://www.redalyc.org/articulo.oa?id=373544192002

Atributos Seleccionados	Estandarización	Tipo de dato
Condición de Deserción	Deserción	Nominal: Des, NoDes
Total de Finales Aprobados de 1°año	1°Apr	Numérico
Proporción de Materias Cursadas del Año 1 (calendario)	Curs1	Numérico
Cantidad de Fracasos de Cursado del Año 1 (calendario)	FracC1	Numérico
Número de Finales Aprobados en el Año 1 (calendario)	Apro1	Numérico
Promedio General de 1° Año	PromA1°	Numérico
Promedio Materias Aprobadas de 1° Año	PromG1°	Numérico
Edad de Ingreso	EdadI	Numérico
Establecimiento educativo (previo)	Est	Nominal: Bachi, EscEdMed, Cen, Tec, Com, Inst, Col, EdSup, Nor, Otros.
Localización geográfica (de origen)	Loc	Nominal: Pdas, IntProv, Otras

Como metodología de análisis se implementa el Proceso KDD, que consta de las fases: integración y recopilación de datos, filtrado de datos, minería de datos y evaluación e interpretación de resultados. Durante el desarrollo del proceso de KDD, como consecuencia de los resultados intermedios, es frecuente interrumpir la secuencia de fases del proceso, para volver a retomar en alguno de los pasos anteriores, siendo así un proceso iterativo e interactivo necesario para lograr una alta calidad del conocimiento a descubrir.

En minería de datos, existe la necesidad de determinar en qué nivel de madurez se encuentran los procesos y modelos, y si son adecuados para resolver el o los problemas planteados, por lo que deben ser revisados, interpretados y evaluados, y finalmente concluir si es posible extraer conocimiento significativo.

Partiendo de los datos operacionales provenientes de la base de datos de la Universidad, se llevó a cabo una interpretación del dominio de la aplicación que se refiere a la información registrada de índole académico (fase de integración y recopilación de datos). Debido a la gran cantidad de atributos disponibles, más de 50 tablas almacenadas en una base de datos relacional, que cuentan con información de los estudiantes de índole personal, asistencias a clases, calificaciones, entre otros; se realizó una recopilación de atributos para determinar los de mayor relevancia respecto a la condición de deserción. En la fase de filtrado de datos se realizó un control y depuración exhaustiva de los datos para hallar una coherencia completa de las tablas y subconjuntos de datos a utilizar.

Se utilizaron dos técnicas de selección de atributos disponibles en la herramienta Weka. La primera técnica utiliza algoritmos que se distinguen por su forma de evaluar los atributos, clasificándose en: filtros, donde se seleccionan y evalúan los atributos en forma independiente del algoritmo de aprendizaje, y envoltorios (wrappers), los cuales usan el desempeño de algún clasificador (algoritmo de aprendizaje) para determinar lo deseable de un subconjunto.

Otra técnica aplicada es la denominada ‘selección de atributos’, utilizada para identificar, en base a un atributo en particular, cuáles son los que más inciden sobre el atributo objeto (en este caso la condición de deserción). Esto permite a su vez, optimizar posteriores

pruebas y resultados a obtener con la técnica de clasificación, sobre todo para evitar clasificaciones muy complejas, como por ejemplo árboles de decisiones extensos y por ende difíciles de interpretar. El método de evaluación aplicado es CfsSubsetEval y el de búsqueda BestFirst, los que ofrecen una selección de subconjuntos de atributos de mayor calidad según Eckert y Suénaga (2015). Se han probado alternativas a los algoritmos para cada método, pero a los efectos prácticos, no se han encontrado variaciones significativas en los resultados finales.

El modo de evaluación utilizado en los algoritmos de selección de atributos y de clasificación, es el de validación cruzada, el cual divide n veces el mismo conjunto de datos mutuamente excluyente y de igual tamaño; n-1 conjuntos se utilizan para construir el clasificador y con el conjunto restante se válida (particiones estratificadas); y así las particiones de test no se superponen. El clasificador final se construye con todos los subconjuntos de datos y la precisión se obtiene del promedio total. El número de subconjuntos o pliegues de validación cruzada utilizados en el estudio es de 10, lo que provoca que la evaluación sea lenta, pero precisa.

Como atributo-indicador se consideraron diversos parámetros, entre los más relevantes podemos mencionar: promedio de asignaturas aprobadas; promedio general; condición final de cursado de materias (si el alumno regulariza o no la materia al final del curso); calificaciones obtenidas en exámenes finales; graduación final (obtención del título); abandono (deserción); entre otros.

Una vez definido, dispuesto y adecuado el conjunto de datos a procesar (vista minable) se procede a la aplicación de técnicas y algoritmos de MD (fase de minería de datos). La técnica utilizada es la de clasificación, basada en un modelo destinado a predecir la categoría de instancias en función de una serie de atributos de entrada, a partir del cual el clasificador aprende un esquema de clasificación de los datos. El primer algoritmo utilizado es C4.5, que genera un árbol de decisión a partir de las variables disponibles, mediante particiones realizadas recursivamente, según la estrategia de primero en profundidad, su implementación en WEKA se denomina J48. [3]

El segundo algoritmo empleado se denomina Naïve Bayes aumentado a árbol (Tree Augmented Network (TAN)), como todos los clasificadores Bayesianos, se basan en el teorema de Bayes, conocido como la fórmula de la probabilidad de las causas. Naïve Bayes (NB) es una simplificación que ha demostrado una alta exactitud y velocidad cuando se ha aplicado a grandes volúmenes datos. El modelo TAN de manera general obtiene mejores resultados que NB, manteniendo la simplicidad computacional y la robustez; el conjunto de padres del atributo a clasificar C, es vacío, mientras que el conjunto de variables padres de cada uno de los atributos predictores Xi, contiene necesariamente al atributo a clasificar, y como mucho otro atributo. La implementación del algoritmo TAN en WEKA se denomina BayesNet. [3]

Como último algoritmo a evaluar, se escogió a OneR, el cuál es uno de los algoritmos clasificadores más sencillos y rápidos; dado que simplemente identifica el atributo que mejor explica la clase de salida. Si hay atributos numéricos, busca los umbrales para hacer reglas con mejor tasa de aciertos. Al utilizar un clasificador, su precisión y fiabilidad depende principalmente de los casos clasificados correctamente a partir del número total de elementos (fase de evaluación e interpretación de resultados). [3]

La herramienta de MD utilizada para la investigación es WEKA, la cual se caracteriza por utilizarse bajo licencia GNU, y además se diseñó específicamente para ser utilizada en investigación y con fines educativos. El paquete WEKA contiene una colección de herramientas de visualización, algoritmos para el análisis de datos, modelado predictivo y descriptivo, unido a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. [3]

Algoritmo Clasificador C4.5 - Árbol de Decisión

Los resultados de la herramienta se presentaron en forma de esquema y gráficamente. A partir de los atributos de entrada, se obtiene como resultado una serie de condiciones representadas de forma escrita mediante un conjunto de reglas, condiciones del tipo si-sino (if-else) y gráfica mediante un árbol de decisión. En la clasificación del Algoritmo J48 (C4.5) para predecir casos de deserción y permanencia se puede apreciar el conjunto de reglas

generado para clasificar los casos de deserción (“Des”) y permanencia (“NoDes”). El nodo o condición inicial representa la cantidad de exámenes finales aprobados correspondientes al primer año de la carrera (“1°Apro”), donde se dividen en dos sub- clasificaciones, una para cantidades de materias aprobadas menores o iguales a siete y para las mayores a siete. [3]

Algoritmo Clasificador Naïve Bayes Aumentado a Árbol (TAN)

En los resultados se observó un grafo obtenido para la predicción de la condición de deserción de los estudiantes, donde todos los atributos involucrados se relacionan al nodo padre atributo/nodo objeto a clasificar. Por cada uno de los atributos se pueden visualizar la relación probabilística que posee en relación al atributo objeto (condición de deserción). Para algunos atributos el algoritmo no se detectó relación probabilística con la condición de deserción, esto ocurrió para en número de fracasos en el cursado (“FracC1”), el promedio general (“PromG1°”) y de materias aprobadas (“PromA1°”) en el primer año y la edad de ingreso del estudiante. [3]

Algoritmo Clasificador Reglas OneR

Para la predicción de la condición de deserción, se identifica como atributo condicionante a la cantidad de materias aprobadas del primer año de la carrera (“1°Apro”), con punto de corte en el valor 7, indicando que por debajo de éste número, se clasifica como casos de deserción, así como ocurre con los casos donde no aprobaron ninguna materia (“?”: indica que el campo está vacío) y para las instancias con 7 o más materias aprobadas, como casos de permanencia (esto ocurrió en el 76,6 % de los casos (655 de 855)). En la Tabla 2 se muestra la evaluación de resultados de los algoritmos descritos anteriormente. [3]

Tabla 2. Evaluaciones de Resultados de los Algoritmos de Clasificación [3]
Fuente: https://www.redalyc.org/articulo.oa?id=373544192002

Algoritmos	ICC	Deserción	VP	FP	Precisión
J48 (C4.5)	80.234%	Des	79.1%	18.7%	79.7%
		NoDes	81.3%	20.9%	80.8%
BayesNet (TAN)	78.129%	Des	81.0%	24.5%	75.3%
		NoDes	75.5%	19.0%	81.1%
OneR	76.608%	Des	83.7%	30.0%	72.1%
		NoDes	70.0%	16.3%	82.3%

Las herramientas de MD brindan resultados que deben ser interpretados y traducidos a diagnósticos y consecuencias del ámbito real (en este caso la universidad). Esto implica que los resultados de la aplicación de las técnicas se han utilizado para explicar parte del comportamiento de la situación en cuanto a la permanencia y su determinación a partir del desempeño académico de los estudiantes. Las posibles consecuencias y acciones tendientes a la toma de decisiones específicas, está sujeta a consideraciones de otros integrantes del cuerpo académico de la institución educativa.

En el proceso KDD, la preparación y acondicionamiento de los datos es la etapa más extensa y a la vez fundamental porque en gran medida los resultados posteriores dependen de ésta. En las etapas intermedias, es crítico llevar a cabo análisis e interpretaciones de resultados parciales, ya que a partir de éstos se retoma el proceso y continúa la depuración y refinamiento del conocimiento extraído.

Mediante la aplicación de algoritmos de minería de datos llevada a cabo en el presente trabajo se pudo identificar que durante el primer año de la carrera es donde adquieren mayor importancia las acciones de contención, apoyo, tutoría y todas aquellas actividades que mejoren la situación académica del alumno al ingreso en la universidad.

Se detectaron atributos que al procesarlos y asociarlos a criterios específicos, se relacionan fuertemente con la deserción y permanencia, el principal de ellos es cantidad de asignaturas aprobadas del primer año, debido a que marca una tendencia notable sobre el resto de la carrera; otros que se destacan son: el número de asignaturas cursadas, los casos donde el estudiante no regulariza la materia al cursarlas, la edad de ingreso, la procedencia; la combinación de estos criterios obtuvo porcentajes de aciertos, de entre un 76% y un 80% de los casos clasificados correctamente.

En el análisis de los resultados obtenidos de los algoritmos de clasificación C4.5 (J48), TAN (BayesNet) y OneR, se pudo observar porcentajes de aciertos similares, sin embargo, no identifican exactamente los mismos atributos. Incluso, dentro de un mismo modelo (por ejemplo, el árbol de decisión C4.5), no todos los atributos tienen la misma importancia,

existiendo algunos que el método no lo considera significativo y que podría suponerse importantes (por ejemplo, el establecimiento educativo previo de los estudiantes).

CAPÍTULO II: METODOLOGÍA Y DESARROLLO

2.1 FUNDAMENTOS TEÓRICOS

2.1.1 ¿Qué es el Machine Learning?

El Machine Learning o Aprendizaje Automático es un subcampo de las ciencias de la computación y una rama de la inteligencia artificial, se define como la aplicación de técnicas y algoritmos capaces de aprender a partir de distintas y nuevas fuentes de información, construyendo algoritmos que mejoren de forma autónoma con la experiencia. Esto permite disponer de métodos capaces de detectar automáticamente patrones en los datos, y usarlos para predecir sobre datos futuros en un entorno de incertidumbre. [4]

2.1.2 Terminología clave de aprendizaje automático (AA)

Para comprender el presente trabajo es importante conocer la terminología asociada al objeto de estudio, por ello definiremos conceptos que son de vital importancia.

- Tipos de algoritmos de AA

  Se clasifican en supervisado, no supervisado y reforzado.

  - Supervisado
    - Son algoritmos que se basan en datos que previamente han sido proporcionados por un experto. Normalmente este tipo de algoritmos son ampliamente usados cuando se tienen datos que permiten que el algoritmo aprenda por medio de ejemplos, previamente etiquetados. Este tipo de algoritmos funciona entrenando un modelo de aprendizaje basándose en datos históricos. Los nuevos casos forman parte de los datos de prueba que son suministrados al modelo para lograr una prediccion sobre ellos (Figura 2). [5]

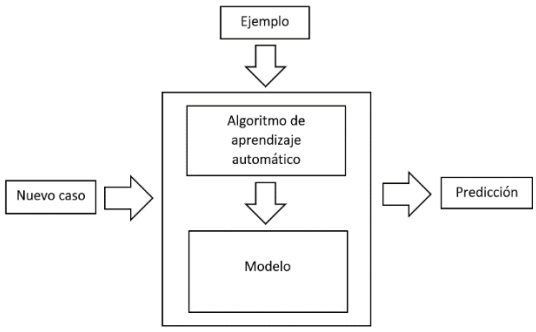


Figura 1. Aprendizaje automático supervisado
Fuente: Elaboración propia.

- No Supervisado
  - En este tipo de algoritmos el modelo de aprendizaje es ajustado en base a las observaciones. Se diferencia del aprendizaje supervisado en que no existen un conocimiento previo. Los datos proporcionados en los ejemplos son tratados como un conjunto de variables aleatorias, permitiendo crear un modelo de densidad para el conjunto de datos. A continuación, se presenta una proyección de datos con el algoritmo k-means, el cual crea un modelo de aprendizaje no supervisado (Figura 2). [5]

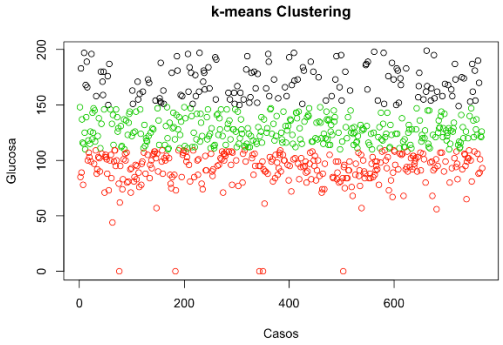


Figura 2. Clustering con el algoritmo k-means
Fuente: C. Machine Learning, Google Inc., 2019

- Por refuerzo
  - Los algoritmos de aprendizaje por refuerzo aprenden por premios, es decir por ensayo y error. Los datos de entrada se toman a través de un feedback o retroalimentación del entorno. El algoritmo aprende de

forma inteligente cuando toma decisiones acertadas, mejorando considerablemente los procesos de decisión (Figura 3). [5]

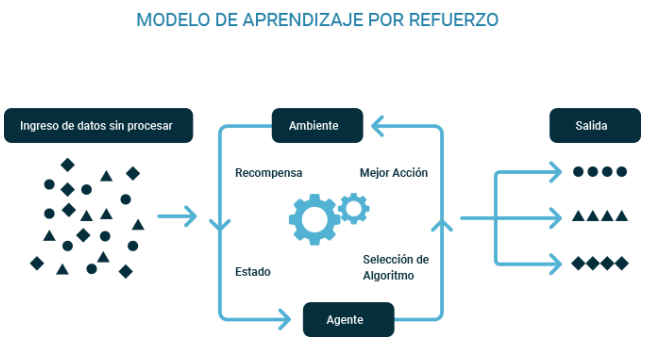


Figura 3. Aprendizaje automático por refuerzo
Fuente: C. Machine Learning, Google Inc., 2019

- Etiquetas
    - Son el valor que estamos prediciendo. En la regresión lineal simple, por ejemplo, “Estimar precio futuro del trigo”.
    - Una etiqueta también puede estar asociada a una imagen, video, audio o cualquier cosa.

- Atributos
    - Variable de entrada, es decir, la variable x en la regresión lineal simple. Un ejemplo claro de lo que son los atributos se lleva a cabo en la detección de SPAM (Correo no solicitado):
        - Palabras en el texto del correo electrónico
        - Dirección del remitente
        - Hora del día a la que se envió
        - Presencia de la frase "un truco increíble" en el correo electrónico

- Ejemplos
    - Instancia de datos en particular, **x**. (La **x** se coloca en negrita para indicar que es un vector), existen dos tipos de ejemplos:

- Etiquetados, incluyen tanto atributos como etiqueta, este tipo se usa para entrenar el modelo.
    - Sintaxis: labeled examples: {features, label}: (x, y)
    - Volviendo al caso del detector de SPAM los ejemplos etiquetados serian:
        - Los correos electrónicos individuales que los usuarios marcaron explícitamente cómo “es spam” o “no es spam”.
- Sin etiqueta, los ejemplos sin etiqueta contienen atributos, pero sin etiqueta. Una vez que el modelo se entrena con ejemplos etiquetados, ese modelo se usa para predecir la etiqueta en ejemplos sin etiqueta.
    - Su sintaxis es: unlabeled examples: {features, ?}: (x, ?)
    - En el ejemplo del detector de SPAM los ejemplos sin etiqueta son correos electrónicos nuevos que las personas todavía no han etiquetado.

- Modelos
    - Un modelo define la relación entre atributos y etiqueta.
    - Ejemplo: Un modelo de detección de spam podría asociar de manera muy definida determinados atributos con "es spam".

Tabla 3. Fases en el ciclo de un modelo
Fuente: Terminología clave de AA
https://developers.google.com/machine-learning/crash-course/framing/ml-terminology?hl=es-419

1. Entrenamiento	Crear o aprender el modelo	Es decir, le muestras ejemplos etiquetados al modelo y permites que este aprenda gradualmente las relaciones entre los atributos y la etiqueta.

2. Inferencia	Aplicar el modelo entrenado a ejemplos sin etiqueta. Es decir, usas el modelo entrenado para realizar predicciones útiles (y').	Ejemplo: Durante la inferencia, puedes predecir medianHouseValue para nuevos ejemplos sin etiqueta.

- Regresión frente a clasificación
    - Un modelo de regresión predice valores continuos
        - Ejemplo: Los modelos de regresión hacen predicciones que responden a preguntas como las siguientes: ¿Cuál es el valor de una casa en California? ¿Cuál es la probabilidad de que un usuario haga clic en este anuncio?
    - Modelo de clasificación
        - Responde a preguntas como las siguientes: ¿Un mensaje de correo electrónico determinado ¿Es spam o no es spam? ¿Esta imagen es de un perro, un gato o un hámster?

Hasta el momento se han definido términos clave para la comprensión básica de lo que es el aprendizaje automático, esto ayudara a poder interpretar el proceso de desarrollo de la presente investigación. No obstante, conforme se valla presentando nuevos términos se irán aclarando a detalle.

Es importante mencionar que para diseñar nuestro modelo de aprendizaje fue necesario unificar datos provenientes desde diferentes orígenes y formatos, la mayoría de estos fueron instrumentos de diagnóstico sobre un determinado factor en el alumno o docente implicado.

Dando por hecho que los ingredientes de aprendizaje automático son:

Datos + Modelos + Algoritmos.

Así mismo, se prepararon los datos considerando lo siguiente:

- Asegurar que los campos tuvieran el tipo adecuado
- Asegurar que los valores pueden ser interpretados correctamente

2.1.3 Ingeniería de atributos (Feature Engineering)

La actividad que más tiempo ocupa es la aplicación de la ingeniería de atributos, que consiste en hacer representación de cada elemento individual de modo adecuado para la tarea de aprendizaje, es decir, como vector de atributos. [6]

A diferencia de la programación tradicional, que se centra en el código, los proyectos de aprendizaje automático se enfocan en la representación. Es decir, una forma para perfeccionar los modelos es que los desarrolladores agreguen atributos y los mejoren.

La ingeniería de atributos es la transformación de datos sin procesar en un vector de atributos. Es necesario tener en cuenta que la ingeniería de atributos implica una gran cantidad de tiempo. [5]

2.1.4 ¿Cómo se aplica la ingeniería de atributos?

Cada campo en nuestro dataset es un atributo que puede ser útil o no, en el aprendizaje.

Predictor

- Campo útil para predecir,
- Podemos añadir al dataset posibles predictores basándonos en otros campos existentes, también podemos modificar un campo que ya tengamos. Por ejemplo, transformando la edad de un campo categórico que agrupe los alumnos por rangos de edades.

- Hay multitud de transformaciones posibles y escoger la más adecuada requiere conocimiento del ámbito del problema a solucionar y cierta experiencia. Aunque se está avanzando en la selección automática de dichas transformaciones.

En el análisis estadístico del dataset se proporciona la siguiente información:

- Distribución de valores en campos
- Errores
- Datos ausentes

Esto nos permite usar la ingeniería de atributos para:

- Transformar datos
- Generar nuevos predictores

2.1.5 Algoritmos de aprendizaje automático supervisado: Clasificadores multiclase

Para dar solución al trabajo de investigación se utilizó árboles de decisión y diferentes clasificadores:

- **Dummy Classifier**, clasificador que realiza predicciones usando reglas simples.
- **Regresión Logística** configurado a las siguientes penalizaciones y solucionadores.
    - L1 - LibLinear
    - ElasticNet – Saga
- **Random Forest** (Bosques aleatorios)
- **XGBoost**, el cual pertenece a la familia de los boosted regressors como lo es SGD.

Lo que hace semejante y aplicable estos algoritmos a nuestro caso de estudio es que todos ellos permiten clasificar un conjunto de atributos en más de dos variables objetivo. A continuación, se presenta una definición breve de cada uno de ellos y algunos otros que también pueden ser aplicables en una problemática semejante.

2.1.6 Dummy Classifier

Este clasificador es útil como una línea de base simple para comparar con otros clasificadores (reales). No debe usarse para problemas reales. [7]

Implementa varias estrategias simples de clasificación: [8]

- *stratified* genera predicciones aleatorias respetando la distribución de clases del conjunto de entrenamiento.
- *most_frequent* siempre predice la etiqueta más frecuente en el conjunto de entrenamiento.
- *Prior* siempre predice la clase que maximiza la clase anterior (*like most_frequent*) y *predict_proba* devuelve la clase anterior.
- *uniform* genera predicciones uniformemente al azar.
- *constant* siempre predice una etiqueta constante que proporciona el usuario.
  Una de las principales motivaciones de este método es la calificación F1, cuando la clase positiva es minoritaria.

2.1.7 Regresión logística

La regresión logística (RL) forma parte del conjunto de métodos estadísticos que caen bajo tal denominación y es la variante que corresponde al caso en que se valora la contribución de diferentes factores en la ocurrencia de un evento simple. [9]

En general, la regresión logística es adecuada cuando la variable de respuesta Y es politómica (admite varias categorías de respuesta, tales como mejora mucho, empeora, se mantiene, mejora, mejora mucho), pero es especialmente útil en particular cuando solo hay dos posibles respuestas (cuando la variable de respuesta es dicotómica), que es el caso más común. [9]

- **LibLinear**

  Es una biblioteca de código abierto para la clasificación lineal a gran escala. Admite regresión logística y máquinas de vectores de soporte lineal. Ofrece herramientas de línea de comandos fáciles de usar y llamadas a la biblioteca para usuarios y desarrolladores. Hay documentos completos disponibles tanto para principiantes como para usuarios avanzados. Los experimentos demuestran que “LibLinear” es muy eficiente en grandes conjuntos de datos dispersos. [10]

- **ElasticNet**

  Pertenece a la familia de Elastic Net, el cual es un modelo de regresión lineal que normaliza el vector de coeficientes con las normas L1 y L2. Esto permite generar un modelo en el que solo algunos de los coeficientes sean no nulos, manteniendo las propiedades de regularización de Ridge. La función de coste es equivalente a la ecuación 1:

$$RSS_{elastic\ net} = \sum_{i=1}^{n}(y_i - f(x_i))^2 + \alpha\left(\lambda\sum_{j=1}^{p}\beta^2 + (1-\lambda)\sum_{j=1}^{p}|\beta_j|\right)$$

Ecuación 1. Función de coste de Elastic Net.
Fuente: I. Chaos, 2020

  El parámetro λ regula el peso dado a la regularización impuesta por Ridge y por Lasso. Desde este punto de vista Elastic Net es un superconjunto de ambos modelos.
  En el caso de que exista cierta colinealidad entre varias características predictivas, Elastic Net tenderá a escoger una o todas (aun con coeficientes menores) en función de cómo haya sido parametrizado. [11]

2.1.8 Random Forest (Bosques aleatorios)

Los bosques aleatorios (RF) se utilizan con frecuencia en muchas aplicaciones de visión artificial y aprendizaje automático. Su popularidad se debe principalmente a su alta eficiencia computacional durante el entrenamiento y la evaluación, al tiempo que se obtienen resultados de vanguardia. Sin embargo, en la mayoría de las aplicaciones, los RF se usan fuera de línea. Esto limita su usabilidad para muchos problemas prácticos, por ejemplo, cuando los datos de entrenamiento llegan secuencialmente o la distribución subyacente cambia continuamente. [12]

2.1.9 XGBoost

El refuerzo de árboles es un método de aprendizaje automático altamente efectivo y ampliamente utilizado. El sistema escalable de impulso de árbol de extremo a extremo

llamado XGBoost, es ampliamente utilizado por los científicos de datos para lograr resultados de vanguardia en muchos desafíos de aprendizaje automático. [13]

2.1.10 Máquina de Soporte Vectorial

Una Máquina de Soporte Vectorial (SVM) aprende la superficie decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un kernel Gaussiano u otro tipo de kernel a un espacio de características en un espacio dimensional más alto, donde se busca la máxima separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada, puede separar los datos en todas las clases distintas, cada una formando un agrupamiento. [14]

2.1.11 Naive bayes

Es un algoritmo de los más simples, se le denomina un clasificador bayesiano ingenuo porque supone que todas las variables en el conjunto de datos son ingenuas, es decir, que no están correlacionadas entre sí.

Naive Bayes es un algoritmo de aprendizaje simple que utiliza la regla de Bayes junto con una fuerte suposición de que los atributos son condicionalmente independientes, dada la clase. Si bien esta suposición de independencia a menudo se viola en la práctica, los ingenuos Bayes a menudo ofrecen una precisión de clasificación competitiva. Junto con su eficiencia computacional y muchas otras características deseables, esto lleva a que Bayes ingenuo se aplique ampliamente en la práctica. [15]

Proporciona un mecanismo para utilizar la información en datos de muestra para estimar la probabilidad posterior $P (y | x)$ de cada clase y , dado un objeto x . Una vez que tengamos dichos estimados, podemos usarlos para la clasificación u otras aplicaciones de soporte de decisiones. [15]

2.1.12 Evaluación de la calidad

Como ya vimos, se emplearán varios tipos de algoritmos de clasificación a la solución de nuestro problema, sin embargo, es muy importante conocer la calidad que estos nos ofrecen a través de mecanismos de evaluación aplicables a los mismos. Para dar soporte a ello, se presentan las siguientes métricas de evaluación de la calidad:

- Exactitud (Accuracy)
    - La exactitud es una métrica para evaluar modelos de clasificación. Informalmente, la exactitud es la fracción de predicciones que el modelo realizó correctamente, formalmente la exactitud tiene la siguiente definición (Ecuación 2).

$$Exactitud = \frac{Número\ de\ predicciones\ correctas}{Número\ de\ predicciones\ totales}$$

Ecuación 2. Fórmula para calcular la exactitud.
Fuente: C. Machine Learning, Google Inc., 2019

    - En la clasificación binaria, la exactitud también se puede calcular en términos de positivos y negativos de la siguiente manera (Ecuación 3):

$$Exactitud = \frac{VP+VN}{VP+VN+FP+FN}$$

Ecuación 3. Exactitud para clasificación binaria.
Fuente: C. Machine Learning, Google Inc., 2019

Donde VP = Verdaderos positivos, VN = Verdaderos negativos, FP = Falsos positivos y FN = Falsos negativos.

- Precisión [5]
    - La precisión intenta responder a la pregunta:
        - ¿Qué proporción de identificadores positivas fue correcta?
        - La precisión se define de la siguiente manera (Ecuación 4):

$$Precisión = \frac{VP}{VP+FP}$$

Ecuación 4. Fórmula para calcular la precisión.
Fuente: C. Machine Learning, Google Inc., 2019

Nota: Un modelo que no produce falsos positivos tiene una precisión de 1.0.

Por ejemplo:

Suponiendo que se tienen los siguientes valores:

Verdaderos positivos (VP) = 1	Falsos positivos (FP) = 1
Falsos negativos (FN) = 8	Verdaderos negativos (VN) = 90

El cálculo de la precisión es igual a (Ecuación 5):

$$Precisión = \frac{VP}{VP+FN} = \frac{1}{1+1} = 0.5$$

Ecuación 5. Ejemplo de cálculo de precisión.
Fuente: C. Machine Learning, Google Inc., 2019

Nuestro modelo tiene una precisión de 0.5, es decir, cuando predice de forma correcta, acierta el 50% de las veces.

- Exhaustividad [5]
  - La exhaustividad intenta responder a la siguiente pregunta:
    - ¿Qué proporción de positivos reales se identificó correctamente?
    - La exhaustividad se define de la siguiente manera:

$$Exhaustividad = \frac{VP}{VP+FN}$$

Ecuación 6. Fórmula para calcular la exhaustividad.
Fuente: C. Machine Learning, Google Inc., 2019

Nota: Un modelo que no produce falsos negativos tiene una recuperación de 1.0.

Por ejemplo, calculemos la exhaustividad de nuestro clasificador con los siguientes valores de ejemplo (Ecuación 7):

Verdaderos positivos (VP) = 1	Falsos positivos (FP) = 1
Falsos negativos (FN) = 8	Verdaderos negativos (VN) = 90

$$Recall = \frac{VP}{VP+FN} = \frac{1}{1+8} = 0.11$$

Ecuación 7. Ejemplo de cálculo de exhaustividad.
Fuente: C. Machine Learning, Google Inc., 2019

Nuestro modelo tiene una recuperación de 0.11. Es decir, identifica correctamente el 11% de las predicciones afirmativas.

2.1.13 Precisión y exhaustividad: Una lucha incesante

Para evaluar completamente la efectividad de un modelo, debemos examinar la precisión y la recuperación. Lamentablemente, con frecuencia hay tensión entre precisión y exhaustividad. Esto quiere decir que, al mejorar la precisión, generalmente se reduce la exhaustividad, y viceversa. [5]

2.1.14 Puntaje F1

Combinación lineal de los valores de ambas métricas (Precisión y Exhaustividad) en una media armónica denominada valor-F (F-score o F1). [16] Esta métrica podria ser la mejor si se necesitara un equilibrio entre precisión y exhaustividad y existiera una distribución de clase desigual (gran cantidad de negativos reales), su fórmula para calcular es la siguiente (Ecuación 8):

$$F1 = 2\frac{Precisión * Exhaustividad}{Precisión + Exhaustividad}$$

Ecuación 8. Fórmula para calcular F1.
Fuente: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

2.1.15 Matriz de confusión

En realidad, todas las métricas de evaluación basan sus cálculos en cuatro variables:

- Positivos verdaderos (True positives: TP)
  - Los positivos verdaderos son el número de casos en que el modelo predice la clase de interés (o clase positiva) y acierta.
- Negativos verdaderos (True negatives: TN)

    - Los negativos verdaderos son el número de casos en que el modelo predice las clases distintas de la de interés (o clases negativas y acierta).
- Falsos positivos (False positives: FP)
    - Los falsos positivos son los casos en que el modelo predice la clase positiva erróneamente.
- Falsos negativos (False negatives: FN)
    - Los falsos negativos son los casos en que el modelo predice las clases negativas erróneamente.

2.1.16 Clasificación: ROC y AUC

Como ya se describió, existen diferentes formas de evaluar la eficiencia en los resultados producto de un modelo de aprendizaje, unido a esto es posible lograr una presentación visual del rendimiento de los algoritmos de aprendizaje automático mencionados anteriormente.

2.1.17 Curva ROC

La curva ROC es una metodología de análisis desarrollada por ingenieros eléctricos y de radar durante la Segunda Guerra Mundial para resolver problemas prácticos en la detección de señales. El espacio de la curva ROC es un gráfico bidimensional que permite visualizar, organizar y seleccionar clasificadores basados en su efectividad, en nuestro caso se utilizará para comparar los diferentes parámetros para determinar con cuáles se obtienen mejores resultados. Mediante esta representación es posible conocer la relación entre los "verdaderos positivos" y los "falsos negativos". [17]

En la línea de investigación nos permitirá representar por medio de un gráfico el rendimiento de un modelo de clasificación en todos los umbrales de clasificación. Esta curva representa dos parámetros: Tasa de verdaderos positivos y tasa de falsos positivos.

Tasa de verdaderos positivos (TVP) es sinónimo de exhaustividad y, por lo tanto, se define de la siguiente manera:

$$TVP = \frac{VP}{VP + FN}$$

Ecuación 9. Tasa de verdaderos positivos.
Fuente: C. Machine Learning, Google Inc., 2019

Tasa de falsos positivos (TFP) se define de la siguiente manera:

$$TFP = \frac{FP}{FP + VN}$$

Ecuación 10. Tasa de falsos positivos.
Fuente: C. Machine Learning, Google Inc., 2019

Una curva ROC representa TVP frente a TFP en diferentes umbrales de clasificación. Reducir el umbral de clasificación clasifica más elementos como positivos, por lo que aumentarán tanto los falsos positivos como los verdaderos positivos. En la siguiente figura, se muestra una curva ROC típica (Figura 4). [5]

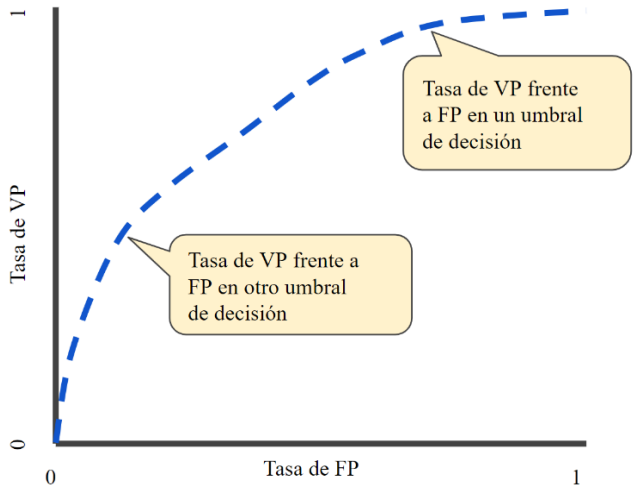


Figura 4. Tasa de VP frente a FP en diferentes umbrales de clasificación.
Fuente: C. Machine Learning, Google Inc., 2019

Para calcular los puntos en una curva ROC, podríamos evaluar un modelo de regresión logística muchas veces con diferentes umbrales de clasificación, pero esto es ineficiente. Afortunadamente, existe un algoritmo eficiente basado en ordenamiento que puede brindarnos esta información, denominado AUC.

2.1.18 AUC: Área bajo la curva ROC

AUC significa "área bajo la curva ROC". Esto significa que el AUC mide toda el área bidimensional por debajo de la curva ROC completa (piensa en un cálculo integral) de (0,0) a (1,1) (Figura 5). [5]

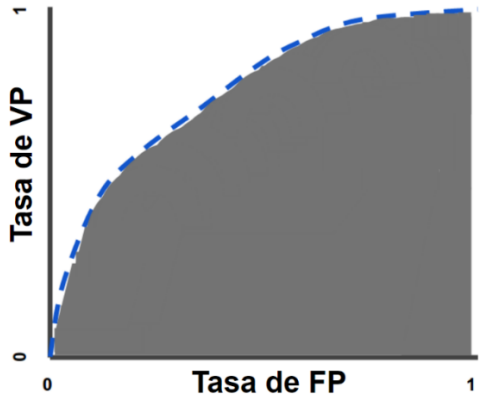


Figura 5. AUC (área bajo la curva ROC)
Fuente: C. Machine Learning, Google Inc., 2019

El AUC proporciona una medición agregada del rendimiento en todos los umbrales de clasificación posibles. Una forma de interpretar el AUC es como la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio. A continuación, se observa, a modo de ilustración, los siguientes ejemplos, que están ordenados de izquierda a derecha en orden ascendente con respecto a las predicciones de regresión logística (Figura 6):

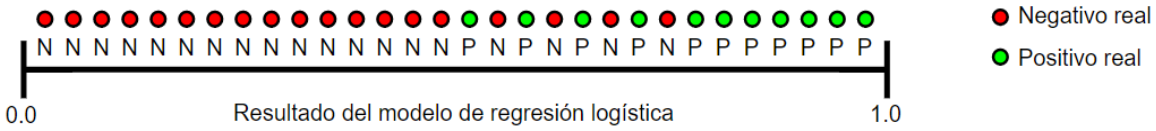


Figura 6. Predicciones en orden ascendente con respecto a la clasificación de regresión logística.
Fuente: C. Machine Learning, Google Inc., 2019

El AUC representa la probabilidad de que un ejemplo aleatorio positivo (verde) se posicione a la derecha de un ejemplo aleatorio negativo (rojo).

El AUC oscila en valor del 0 al 1. Un modelo cuyas predicciones son un 100% incorrectas tiene un AUC de 0.0; otro cuyas predicciones son un 100% correctas tiene un AUC de 1.0.

El AUC es conveniente por las dos razones siguientes: [5]

- El AUC es invariable con respecto a la escala. Mide qué tan bien se clasifican las predicciones, en lugar de sus valores absolutos.

- El AUC es invariable con respecto al umbral de clasificación. Mide la calidad de las predicciones del modelo, sin tener en cuenta qué umbral de clasificación se elige.

Sin embargo, estas dos razones tienen algunas advertencias, que pueden limitar la utilidad del AUC en determinados casos: [5]

- La invariabilidad de escala no siempre es conveniente. Por ejemplo, en algunas ocasiones, realmente necesitamos resultados de probabilidad bien calibrados, y el AUC no nos indicará eso.
- La invariabilidad del umbral de clasificación no siempre es conveniente. En los casos en que hay amplias discrepancias en las consecuencias de los falsos negativos frente a los falsos positivos, es posible que sea fundamental minimizar un tipo de error de clasificación. Por ejemplo, al realizar la detección de spam de correo electrónico, es probable que quieras priorizar la minimización de los falsos positivos (aunque eso resulte en un aumento significativo de los falsos negativos). El AUC no es una métrica útil para este tipo de optimización.

2.2 METODOLOGÍA DE LA INVESTIGACIÓN

Al inicio de nuestra investigación se formularon varios cuestionamientos que permitiera situarnos en la realidad y contexto de nuestro objeto de estudio. A continuación, se muestra una lista de preguntas que ayudaron a ir creando un plan de investigación con la implementación de una metodología apropiada y técnicas de investigación.

2.2.1 Preguntas de la investigación

Resolver con aprendizaje automático y minería de datos los siguientes cuestionamientos:

1. ¿Cuál es el sujeto del problema a resolver?
2. ¿Cuáles son las propiedades de ese sujeto que pensamos que pueden influir en la solución?
3. ¿Cómo predecir los factores que inciden en el avance y retroceso del alumno en la lectura?
4. ¿Cómo predecir los factores que dificultan el cálculo mental?

5. ¿Cómo predecir los factores que interfieren en la redacción de artículos?
6. ¿Cuáles son los factores de riesgo más recurrentes?
7. ¿De dónde obtener los datos?
8. ¿Qué tipo de datos son útiles para el Aprendizaje Automático?
9. ¿Cómo hay que estructurarlos para que se pueda aprender de ellos?
10. ¿Qué transformaciones pueden ayudar al aprendizaje?
11. ¿Estan los datos en el formato adecuado?
12. ¿Qué modelo de aprendizaje es el que mejor puede resolver nuestro problema?
13. ¿Cuál es la calidad de las predicciones del modelo aprendido?
14. ¿Cómo se integra el modelo aprendido en nuestro sistema de producción?
15. ¿Cómo pasar de la definición del problema a la estructura de los datos?
16. ¿Cuál es el impacto en términos de Coste-Beneficio de la aplicación predictiva?

Al analizar las preguntas anteriores, se llegó a la conclusión de que deberían emplearse las siguientes metodologías en la investigación:

- Cualitativa
- Cuantitativa
- Diagnóstica, descriptiva y explicativa
- Investigación documental y de campo

2.2.2 Metodología cualitativa

La investigación cualitativa se implementó en el centro escolar para conocer el proceso interno de la adquisición y el desarrollo de las habilidades básicas del pensamiento ya que el ser humano es un objeto de estudio amplio y complejo, por lo tanto, conocer como el cerebro procesa estas habilidades aunado a las emociones que enfrenta, así como a la madurez intelectual que posee, nos arroja datos distintos que se reflejan en las actitudes y aptitudes durante el proceso educativo. Razón por la que fue necesario continuar mediante esta metodología para permitir que se pueda conocer al alumno como un todo y obtener un resultado lo más cercano posible a la realidad.

2.2.3 Metodología cuantitativa

Durante la recopilación de datos se usó la metodología de investigación cuantitativa, ya que parte de diversos instrumentos prediseñados y aprobados por el consejo técnico escolar que nos brindan información relevante y sumamente importante como entrevistas, tests, cuestionarios, etc., aplicados a los alumnos, tutores, y docentes para poder concentrar y establecer una base de datos que permita situarlos en algunos parámetros y conocer las características en las que coinciden para poder predecir sus resultados educativos.

2.2.4 Metodología diagnóstica, descriptiva y explicativa

Al recopilar datos de la muestra que permitieran servir de entrenamiento y prueba, se empleó una metodología de diagnóstico que permitiera describir y explicar el estado a priori (previo a) y a postiori (posterior a) nuestra predicción y evaluación de nuestro modelo.

2.2.5 Investigación documental y de campo

Desde el inicio de nuestra investigación hasta el final de la misma, se ha empleado la investigación documental y de campo, para la formulación de ideas, conceptos, conjeturas, diseño y modelamiento de la solución de aprendizaje automático que permita predecir los factores que inciden en las habilidades básicas del pensamiento.

Es importante mencionar, que todas estas metodologías mencionadas hasta el momento, han requerido la aplicación de técnicas e instrumentos para la recolección de datos, las cuales se mencionan a continuación:

- Guías de observación
- Entrevista,
- Cuestionario,
- Bitácora o diario de campo.

2.2.6 Población y muestra

La investigación se realizó en la ciudad de Tlapacoyan, Veracruz, a alumnos y docentes de educación primaria estatal pertenecientes a la Zona 058 que la integran 48 escuelas, de la

cual se tomó una muestra de 329 alumnos y 15 docentes dando un total de la muestra de 344 personas examinadas (Tabla 4).

Tabla 4. Población y muestra examinada.
Fuente: Elaboración propia.

Grado/Grupo	Número de alumnos examinados
1a	28
1b	29
2a	27
2b	24
3a	21
3b	23
3c	22
4a	26
4b	26
5a	30
5b	29
6a	23
6b	21

Total de alumnos	329
Total docentes	15

Total de la muestra	344

2.3 METODOLOGÍA DE DESARROLLO

El levantamiento de requerimientos, la organización de un equipo de trabajo, las métricas para medir la productividad y dar un seguimiento al producto de software y personal implicado. Requieren de la implementación de una metodología de gestión de proyectos ágil acorde con las características de los alcances del proyecto en la observación al modelo de dominio como objeto de estudio. Por esta razón se ha elegido la metodología de desarrollo ágil Scrum.

2.3.1 Scrum: Metodología de desarrollo ágil

Definición

SCRUM es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en un Desarrollo Ágil de Software. [18]

Fue desarrolla por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar el desarrollo de proyectos de las principales empresas tecnológicas: Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M y Hewlett-Packard.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan y que los desafíos imprescindibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y concentrándose en maximizar la capacidad del equipo para entregar rápidamente sus desarrollos y responder a requisitos emergentes. [19]

Scrum descompone la organización en pequeños equipos auto-organizados. Cada equipo desarrolla los proyectos en base a entregas parciales «sprints», con el objetivo de alinear expectativas con el cliente y aumentar el valor que se ofrece a los mismos.

SCRUM asume que el proceso de desarrollo de sistemas es un proceso impredecible y complicado que solo puede describirse aproximadamente como una progresión general. SCRUM define el proceso de desarrollo de sistemas como un conjunto suelto de actividades que combina herramientas y técnicas conocidas y viables con lo mejor que un equipo de desarrollo puede diseñar para construir sistemas. Dado que estas actividades son flojas, se utilizan controles para gestionar el proceso y el riesgo inherente. SCRUM es una mejora del ciclo de desarrollo orientado a objetos iterativo/incremental comúnmente utilizado (Figura 7). [20]

Esquema

Figura 7. Marco de trabajo en la metodología SCRUM
Fuente: Mentor day, 2019.

Funcionamiento

1. El cliente/sponsor o “Product Owner” define los requisitos del sistema a desarrollar «Product Backlog», siempre bajo la figura de un asistente de supervisión o “Scrum Master”.
2. Se descomponen estos requisitos en varios paquetes de trabajo más manejables “Sprint Backlog”, que puede ir de 2 a 4 semanas de trabajo por paquete, esta descomposición se realiza en una reunión o “Sprint planning meeting” que puede durar hasta 8 horas y donde se define (el alcance) el “qué” y el “cómo” se va a elaborar el trabajo.
3. El equipo de trabajo auto organizado tiene una reunión diariamente “Daily Scrum” durante unos 15 minutos, en esta reunión cada uno expone que hizo, que va a hacer y que problemas se ha encontrado y se debate entre todos como como realizar las tareas.
4. Cuando termina un sprint se realiza una reunión o «Sprint Review” donde se presenta el producto resultante del “Sprint Backlog”, también puede realizarse una reunión retrospectiva «Sprint Retrospective» de hasta 3 horas, en la que se evalúan las técnicas y habilidades empleadas para valorar si pueden mejorarse y aplicarse para los siguientes Sprint.

5. Repitiéndolo para cada “Sprint Backlog” obtendríamos el producto final como una sucesión de pequeños incrementos.

Reuniones prescritas

Sprint planning meeting: Reunión de Planificación de Sprint.
Daily Scrum: Reunión de seguimiento diaria.
Sprint Review: Reunión de revisión.
Sprint Retrospective: Reunión de retrospectiva.

Roles

1. Product Owner: cliente o sponsor
2. ScrumMaster: supervisor que asiste todo el proceso.
3. Miembros del equipo de desarrollo.

2.3.2 Establecimiento de Stakeholders

Descripción de cliente(s)

Cliente

Nombre:	Secretaria de Educación de Veracruz		
Puesto:	Coordinar la política educativa del Estado y organizar el Sistema Educativo Estatal en todos sus niveles y modalidades.		
Giro:	Educación		
Usuario:	Si	**Tipo:**	Propietario del software
Actividades:	Diseña talleres y/o diplomados y autoriza su implementación		

Descripción de usuario(s)

Puesto:	Director		
Categoría profesional:	Maestra en Tecnologías Aplicadas a la Educación		
Usuario:	Si	**Tipo:**	Director
Actividades:	• Iniciar acciones y asesoría con Padres y Docentes • Favorecer la implementación de estrategias didácticas que mejoren las habilidades básicas del pensamiento		

Puesto:	Supervisor escolar		
Categoría profesional:	Maestra en Educación		
Usuario:	Si	**Tipo:**	Supervisor
Actividades:	Gestionar talleres a docentes		

Puesto:	Docentes		
Categoría profesional:	Lic. en Educación		
Usuario:	Si	**Tipo:**	Docente
Actividades:	Cambiar o implementar metodología		

Descripción de roles en Scrum

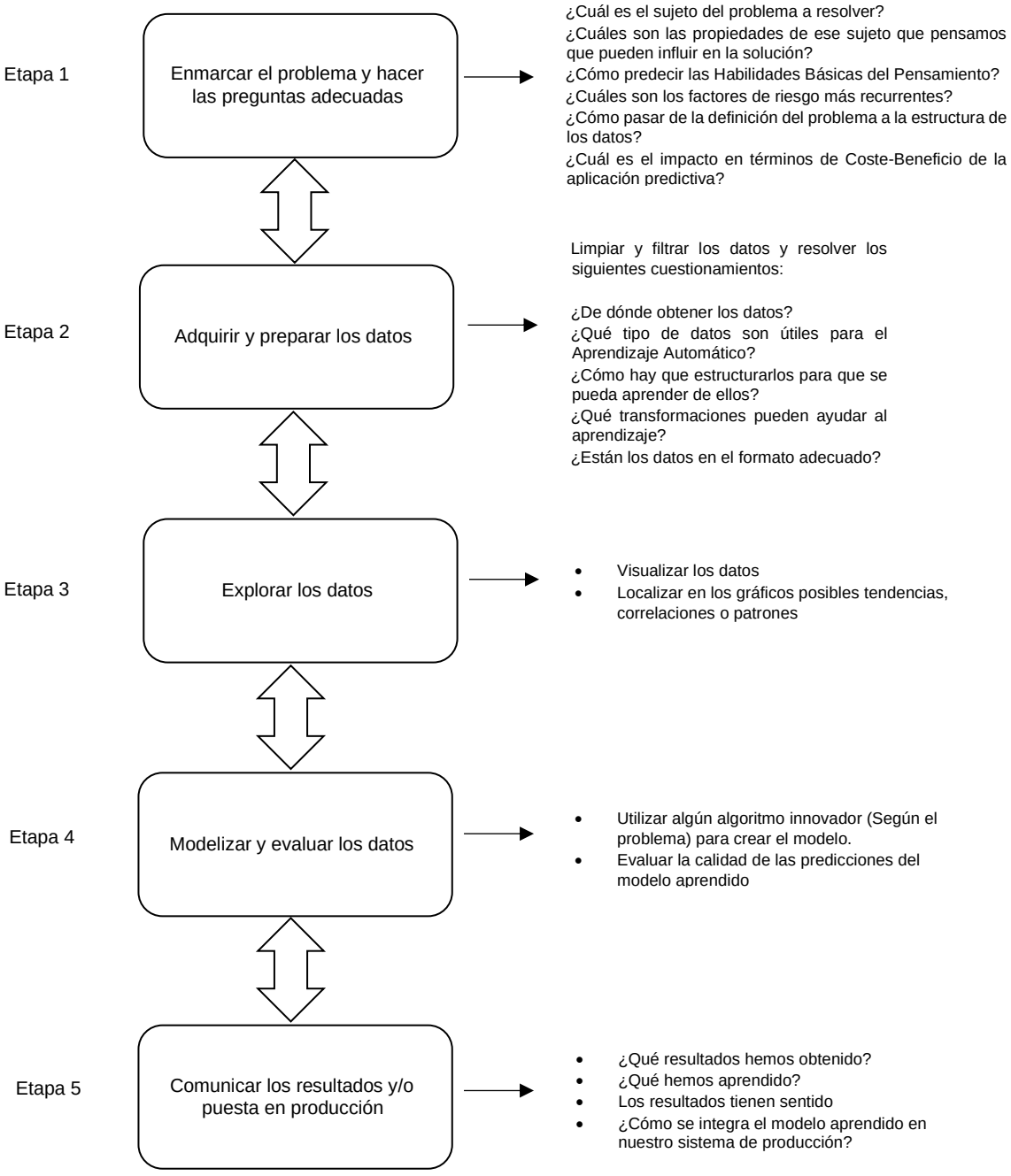
Rol:	Scrum Master
Categoría profesional:	Ingeniero en Sistemas Computacionales
Puesto:	Líder de proyecto
Actividades:	• Se encarga de que se aplique la metodología de desarrollo ágil en todo el equipo • Planifica la implantación de SCRUM junto con la organización • Ayuda a la organización a entender qué interacciones con el equipo aportan valor y cuáles no • Ayuda al Ingeniero de Requerimientos a entender la agilidad • Ayuda al Ingeniero de Requerimientos a priorizar y gestionar efectivamente las historias de usuarios • Ayuda al equipo de desarrollo a convertirse en auto-organizado y multifuncional • Soluciona posibles impedimentos que pudieran surgir • Se asegura de que en el tablero de Kanban halla una definición de DONE • Ayuda a que se lleven a cabo las mejoras tratadas en la reunión de retrospectiva • Junto con el equipo de desarrollo, actualiza el desarrollo en progreso • Realiza cursos de capacitación

Rol:	Product Owner
Categoría profesional:	Ingeniero en Sistemas Computacionales
Puesto:	Ingeniero de Requerimientos
Actividades:	• Decidir qué construir ... Y que no. • Recoger y tener claros los requisitos de software • Definir buenas historias de usuario • Fijar criterios de aceptación para cada historia de usuario • Ordenar y priorizar los items del Product Backlog • Definir el producto mínimo viable • Acordar junto al resto del equipo una definición de DONE • Definir el plan releases • Validar entregas (Sprint Reviews) • Estar disponible y accesible para el equipo • Es el responsable de cancelar el sprint si ocurre un imprevisto extremo • Asegurarse de que todo el mundo entiende los items del Product Backlog

Rol:	Development Team
Categoría profesional:	Ingeniero en Sistemas Computacionales
Puesto	Equipo de desarrollo
Actividades:	• Desarrollo de software • Análisis • Diseño • Codificación • Pruebas

	• Validación • Mantenimiento y evolución • El equipo debe ser auto-organizado. Los propios miembros del equipo establecerán la forma de hacer su trabajo. • Tiene que ser multifuncional. • Todos los miembros deben trabajar con sus habilidades para cumplir el sprint goal. • Los equipos de desarrollo deben ser pequeños, de 3 -7 personas. • Junto con el Scrum Master, se encargan de establecer los ítems del Sprint Backlog, de planificar el sprint. • Estimar las historias de usuario y tareas. • Hacer la demo. • Implementar pruebas de aceptación y pruebas unitarias. • Trabajo de calidad y mejora continua de la calidad (refactorización, por ejemplo) • Participar en los Daily meeting, Sprint Planning Meeting, Sprint Review y Sprint Retrospective. • Estar motivados. • Saber buenas prácticas de programación: pair programming, TDD, integración continua, refactorización, malos olores, patrones de diseño, etc. • Identificar posibles obstáculos y comunicárselos al Scrum Master. • Actualizar el trabajo en progreso (burndown chart) (es responsabilidad tanto del equipo de desarrollo como del Scrum Master)

2.3.3 Metodología para Ciencia de Datos



Las flechas con doble sentido indican que en cualquier momento en el análisis de datos puede ser posible no solo ir hacia delante, sino regresar una etapa atrás y replantearnos algo que no hayamos tenido en consideración.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBAS

3.1 ANÁLISIS DE DATOS

Para iniciar con el análisis de datos es importante mencionar que serán obtenidos desde una Base de Datos almacenada en PostgreSQL.

PostgreSQL es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de base de datos de código abierto más potente del mercado. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [21]

Una vez ya conociendo donde se almacenarán nuestra base de datos relacional, la cual se encuentra normalizada, es importante destacar, que un modelo predictivo hace uso de los datos denormalizados bajo una consulta o un dataset, tomando en cuenta la propiedad de predictor que define a cada atributo.

Por normas de protección de datos y privacidad en el tratamiento de los mismos, nos basaremos solo en el análisis técnico de la muestra.

La muestra de nuestro caso de estudio esta segmentada por datos de grupos correspondientes a un grado especifico y por datos pertenecientes a un cuerpo docente.

Para dar fuerza a nuestro modelo predictivo tomaremos en cuenta las realidades de dichos segmentos basándonos en un análisis predictivo por grupos pertenecientes a un determinado grado.

Conociendo la premisa anterior, los datos que analizaremos y que en adelante llamaremos “atributos”, cumplen con la propiedad de ser predictores y son los siguientes:

Para el caso del alumno:

- Estilos de aprendizaje
- Madurez intelectual
- Relaciones
- Situación emocional
- Estado de salud
- Situación económica
- Situación familiar
- Asistencia
- Participación

Para el caso del docente:

- Planeación de clase
- Desarrollo de clase
- Recursos didácticos
- Evaluación
- Cuadrante cerebral

Todos estos atributos cumplen con la propiedad de ser atributos predictores.

Nuestras variables objetivo será predecir cuándo un alumno “Requiere apoyo”, se encuentra “En desarrollo” o en el nivel “Esperado” en cada una de las Habilidades Básicas del Pensamiento, las cuales son:

- Lectura
- Escritura
- Calculo mental

3.2 SELECCIÓN DE PRUEBAS ESTADÍSTICAS

3.2.1 Exploración de datos previos al análisis

Para poder tener un contexto amplio de nuestro análisis es importante extraer los siguientes datos en Python:

```
Información del dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28 entries, 0 to 27
Data columns (total 14 columns):
ID                    28 non-null int64
Nombre                28 non-null object
Estilos_Aprendizaje   28 non-null object
Madurez_Intelectual   28 non-null object
Relaciones            28 non-null object
Situacion_Emocional   28 non-null object
Estado_Salud          28 non-null object
Situacion_Economica   28 non-null object
Situacion_Familiar    28 non-null object
Asistencia            28 non-null object
Participacion         28 non-null object
Lectura               28 non-null object
Escritura             28 non-null object
Calculo_Mental        28 non-null object
dtypes: int64(1), object(13)
memory usage: 3.1+ KB
None
```

Como podemos observar para nuestro análisis contamos con 28 instancias, esta prueba se realizó con valores categóricos danto para los datos como para las etiquetas.

El valor ID se descarta, para nuestro análisis de datos, pues no tiene ningún valor predictor.

La cantidad de memoria de uso de nuestra fuente de información es: 3.1+ Kilobytes.

Es importante mencionar que para nuestro análisis de datos omitiremos los datos de las columnas correspondiente al ID y Nombre, ya que no tienen una propiedad predictora. Dando como resultado 12 características observadas en 28 alumnos examinados, correspondientes a un grupo.

3.2.2 Principales indicadores estadísticos sobre nuestro conjunto de datos

Describir nuestro dataset es muy importante para conocer algunos indicadores importantes como son:

- Conocer la cantidad de elementos por columna examinados,
- Media,
- Desviación estándar,
- Valor máximo,
- Valor mínimo,
- Percentiles 25%, 50% y 75%.

Tabla 5. Indicadores estadísticos.
Fuente: Elaboración propia en Python.

	Estilos_Aprendizaje	Madurez_Intelectual	Relaciones	Situacion_Emocional	Estado_Salud	Situacion_Economica	Situacion_Familiar	Asistencia	Participac
count	28.000000	28.000000	28.000000	28.000000	28.000000	28.00000	28.000000	28.0	28.000
mean	2.250000	2.714286	1.035714	1.142857	0.964286	1.50000	0.821429	1.0	0.857
std	0.927961	0.712697	0.331343	0.448395	0.188982	0.57735	0.390021	0.0	0.705
min	1.000000	1.000000	0.000000	0.000000	0.000000	0.00000	0.000000	1.0	0.000
25%	1.000000	2.000000	1.000000	1.000000	1.000000	1.00000	1.000000	1.0	0.000
50%	3.000000	3.000000	1.000000	1.000000	1.000000	2.00000	1.000000	1.0	1.000
75%	3.000000	3.000000	1.000000	1.000000	1.000000	2.00000	1.000000	1.0	1.000
max	3.000000	4.000000	2.000000	2.000000	1.000000	2.00000	1.000000	1.0	2.000

.

```
DISTRIBUCIÓN DE HABILIDADES BÁSICAS DEL PENSAMIENTO (HBP)
HBP de LECTURA
Lectura
Esperado          2
Requiere apoyo    26
dtype: int64
HBP de ESCRITURA
Escritura
Esperado          2
Requiere apoyo    26
dtype: int64
HBP de CÁLCULO MENTAL
Calculo_Mental
En desarrollo     13
Esperado          2
Requiere apoyo    13
dtype: int64
```

En estos resultados observamos la distribución de las HBP de un grupo, podemos observar que para las primeras 2 correspondientes a “Lectura” y “Escritura”, no se tiene a ningún alumno en el nivel de “En desarrollo”, lo contrario a la HBP de “Calculo_Mental” que integra una distribución de alumnos en sus 3 respectivos niveles.

3.2.3 Histogramas de features

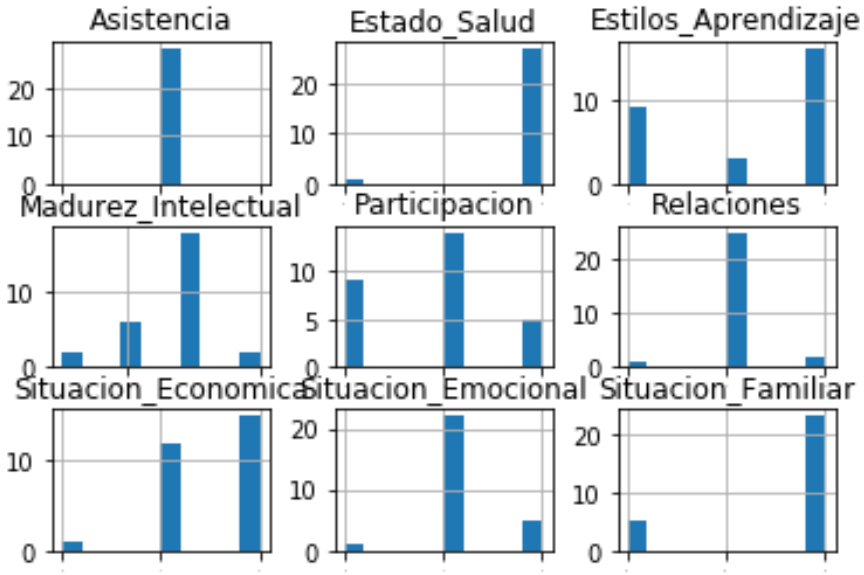


Figura 8. Histogramas de features del alumno.
Fuente: Elaboración propia en Python.

3.2.4 Exploración de datos

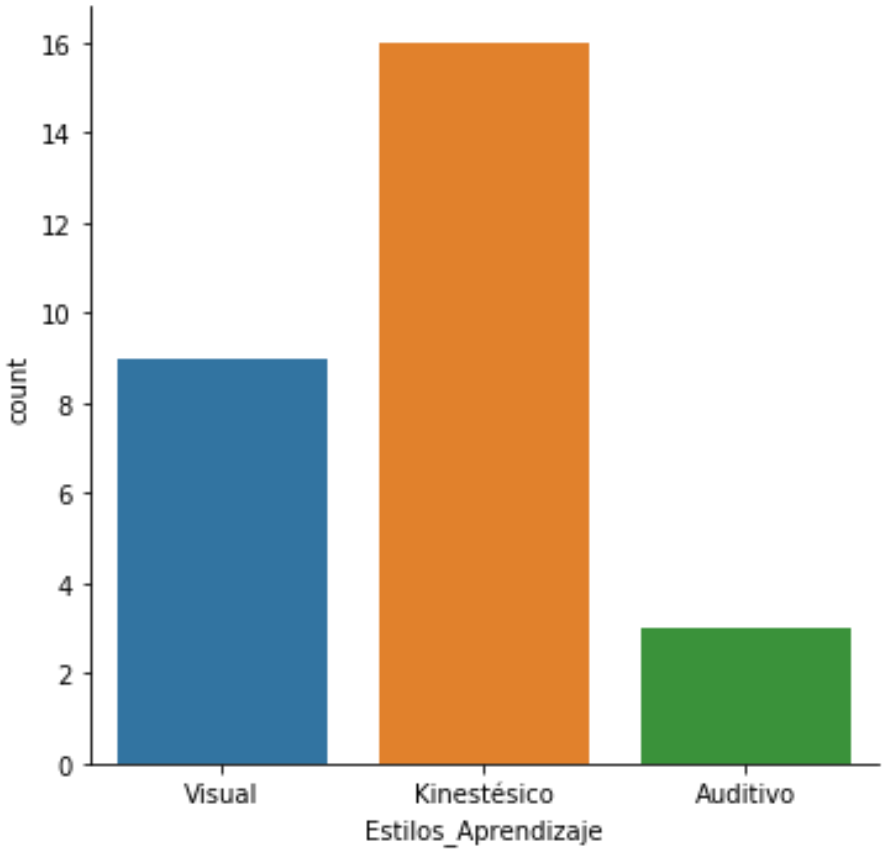


Figura 9. Estilos de aprendizaje.
Fuente: Elaboración propia en Python.

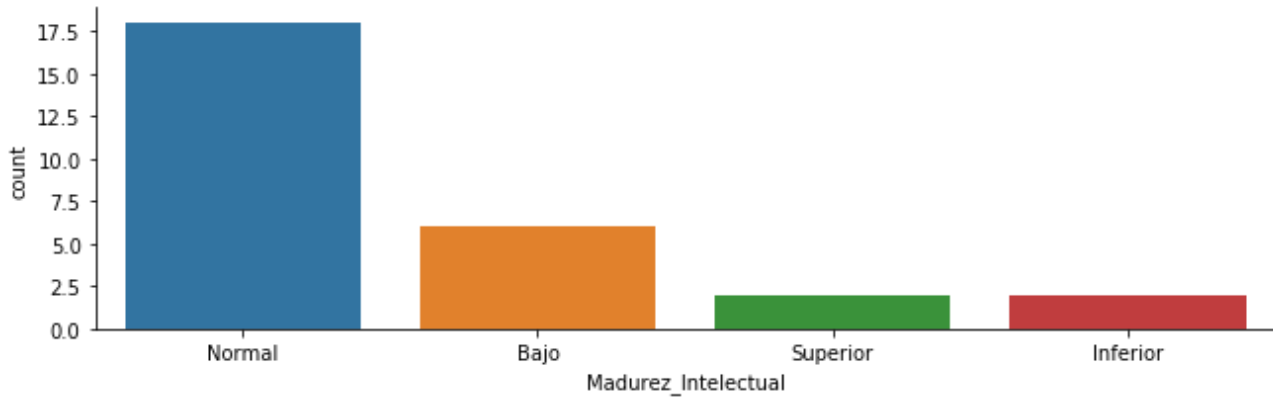


Figura 10. Madurez intelectual.
Fuente: Elaboración propia en Python.

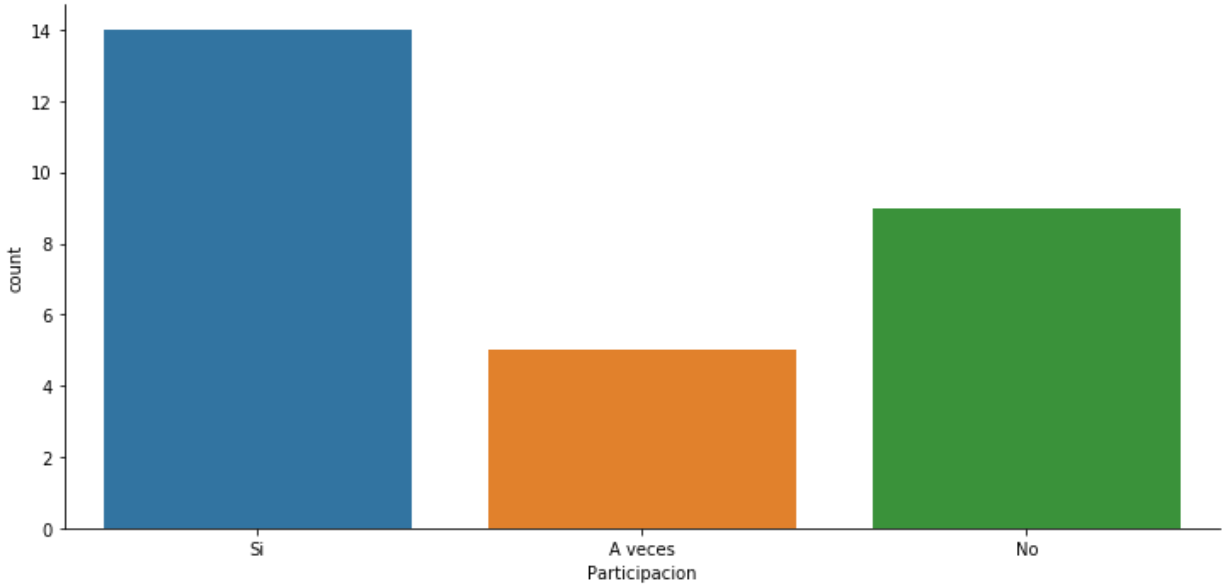


Figura 11. Participación.
Fuente: Elaboración propia en Python.

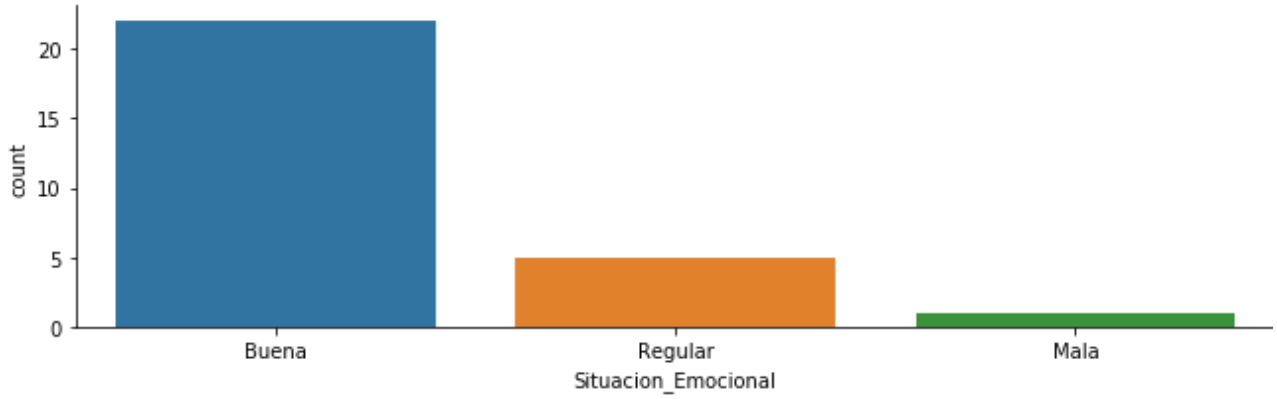


Figura 12. Situación emocional.
Fuente: Elaboración propia en Python.

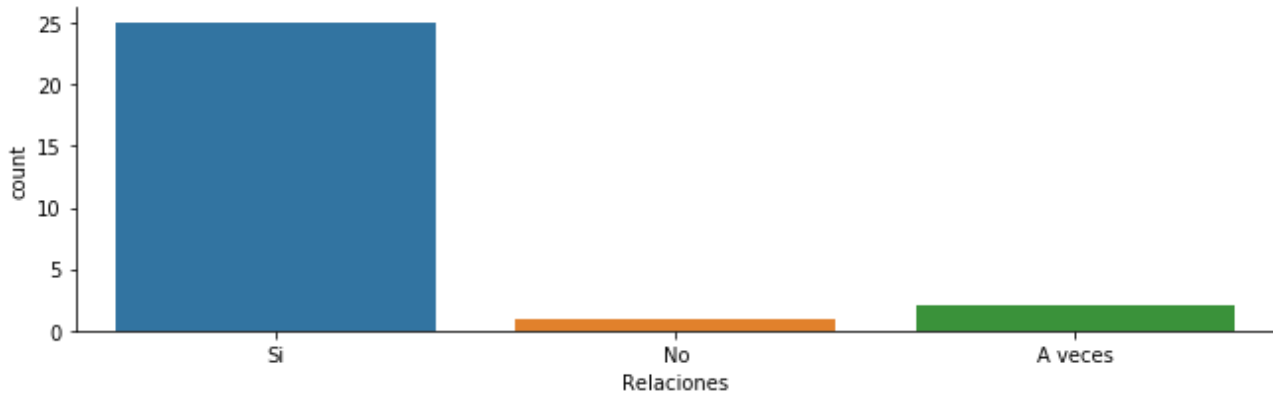


Figura 13. Relaciones.
Fuente: Elaboración propia en Python.

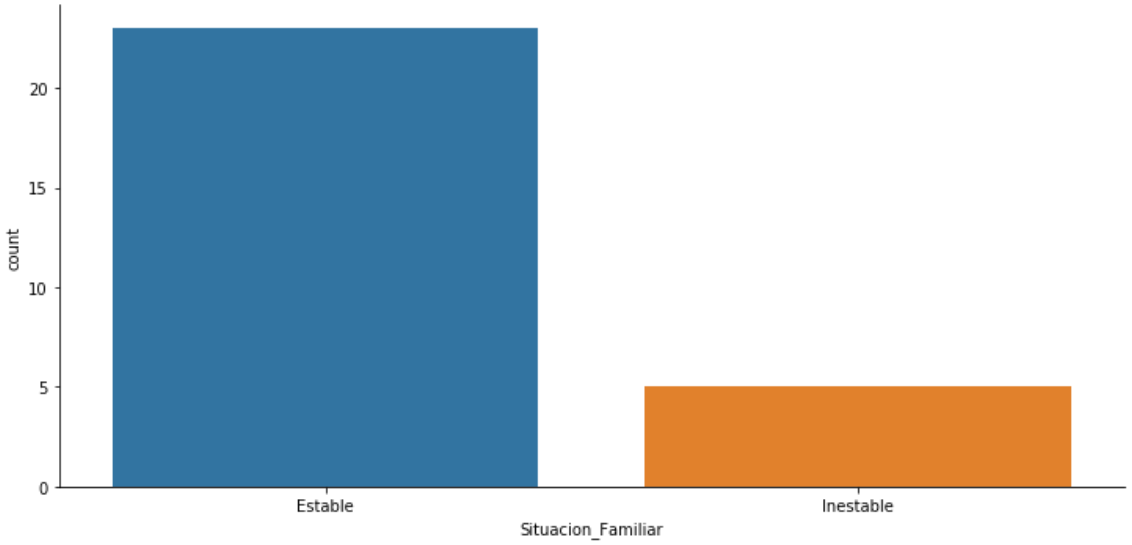


Figura 14. Situación familiar.
Fuente: Elaboración propia en Python.

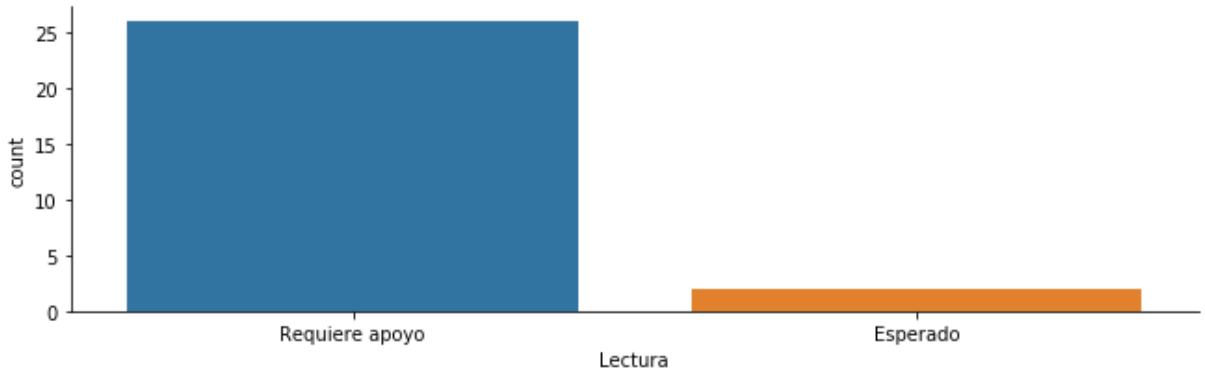


Figura 15. Lectura.
Fuente: Elaboración propia en Python.

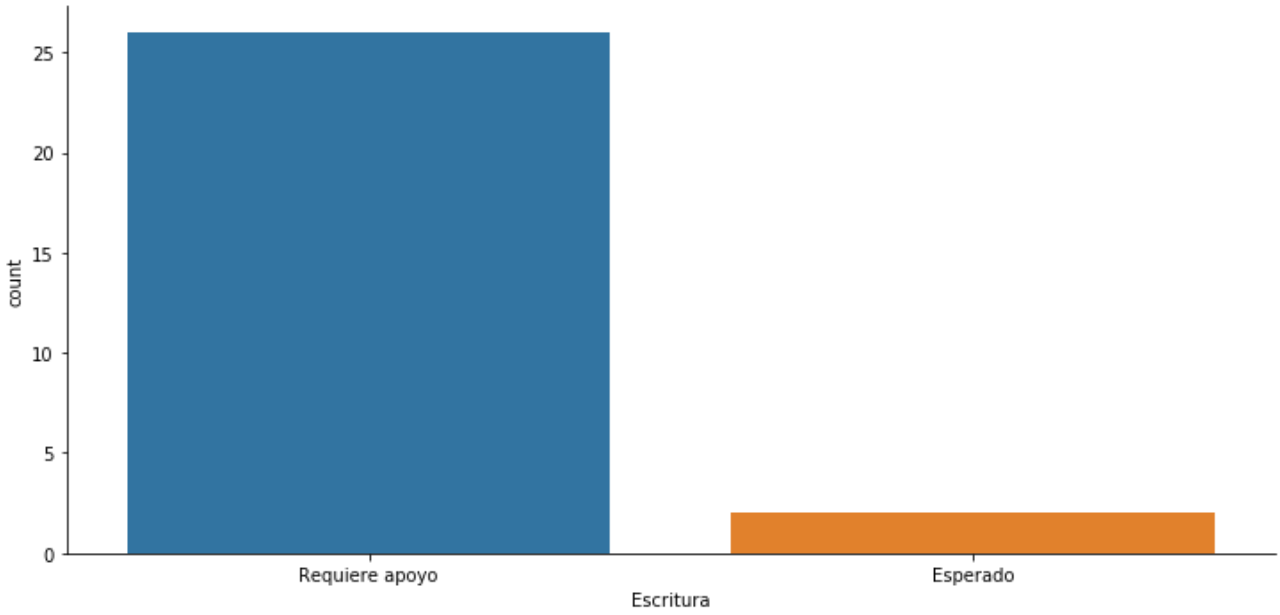


Figura 16. Escritura.
Fuente: Elaboración propia en Python.

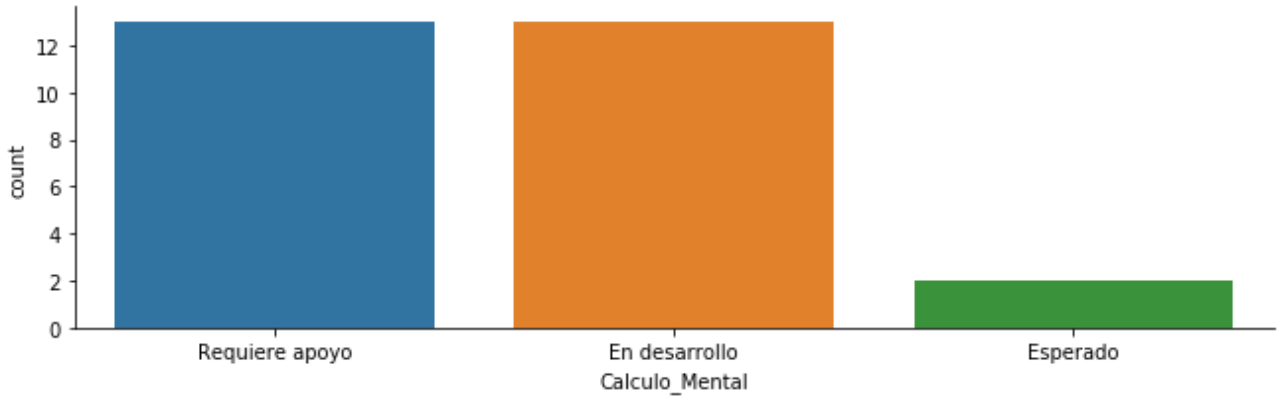


Figura 17. Cálculo mental
Fuente: Elaboración propia en Python.

3.3 REALIZACIÓN DE ANÁLISIS (INTERPRETACIÓN)

3.3.1 Modelo de aprendizaje automático supervisado: Árboles de decisión

Los árboles de decisión se aplican a dos contextos diferentes, en la regresión para la predicción sobre valores continuos y la clasificación que es la que se utilizará para nuestro modelo de predicción debido a las características de la problemática planteada al inicio de la presente investigación.

El árbol de clasificación que se configuró en Python toma en cuenta las siguientes premisas:

1. Se utilizó un archivo de datos con información de los resultados obtenidos en el examen de diagnóstico sobre las HBP en alumnos de Primero A de Educación Primaria.
2. Las variables dependientes de las HBP (Lectura, Escritura y Cálculo metal) son: Esperado, En desarrollo y Requiere apoyo. Los datos de las variables independientes son: Estilos de aprendizaje, Madurez intelectual, Relaciones, Situación emocional, Situación familiar, Asistencia, Participación.
3. Utilizaremos la Madurez Intelectual y la Situación Familiar para crear nuestro arbol de clasificación con las etiquetas Esperado, En Desarrollo y Requiere apoyo y con ello poder clasificar registros nuevos. Para ello crearemos el arbol con el conjunto de datos a los que les llamaremos conjunto de entrenamiento.
4. Nuestro árbol de clasificación se ejecutó con 28 registros de datos históricos de alumnos de los cuales 25% se utilizaron para probar el modelo y 75% para entrenarlo o crear el árbol bajo los criterios de Madurez Intelectual y Situación Familiar.

A continuación, en la Figura 8 se presenta el árbol de clasificación generado a partir de los atributos predictores de un grupo de alumnos seleccionado, del cual se recomienda analizar lo siguiente:

- Condición: Es un nodo donde se toma alguna decisión
- Entropía: Es una medida de impureza.
- Samples: Número de muestras que satisfacen las condiciones necesarias para llegar a este nodo
- Value: Cuántas muestras de cada clase llegan a este nodo
- Class: Variable objetivo de clasificación

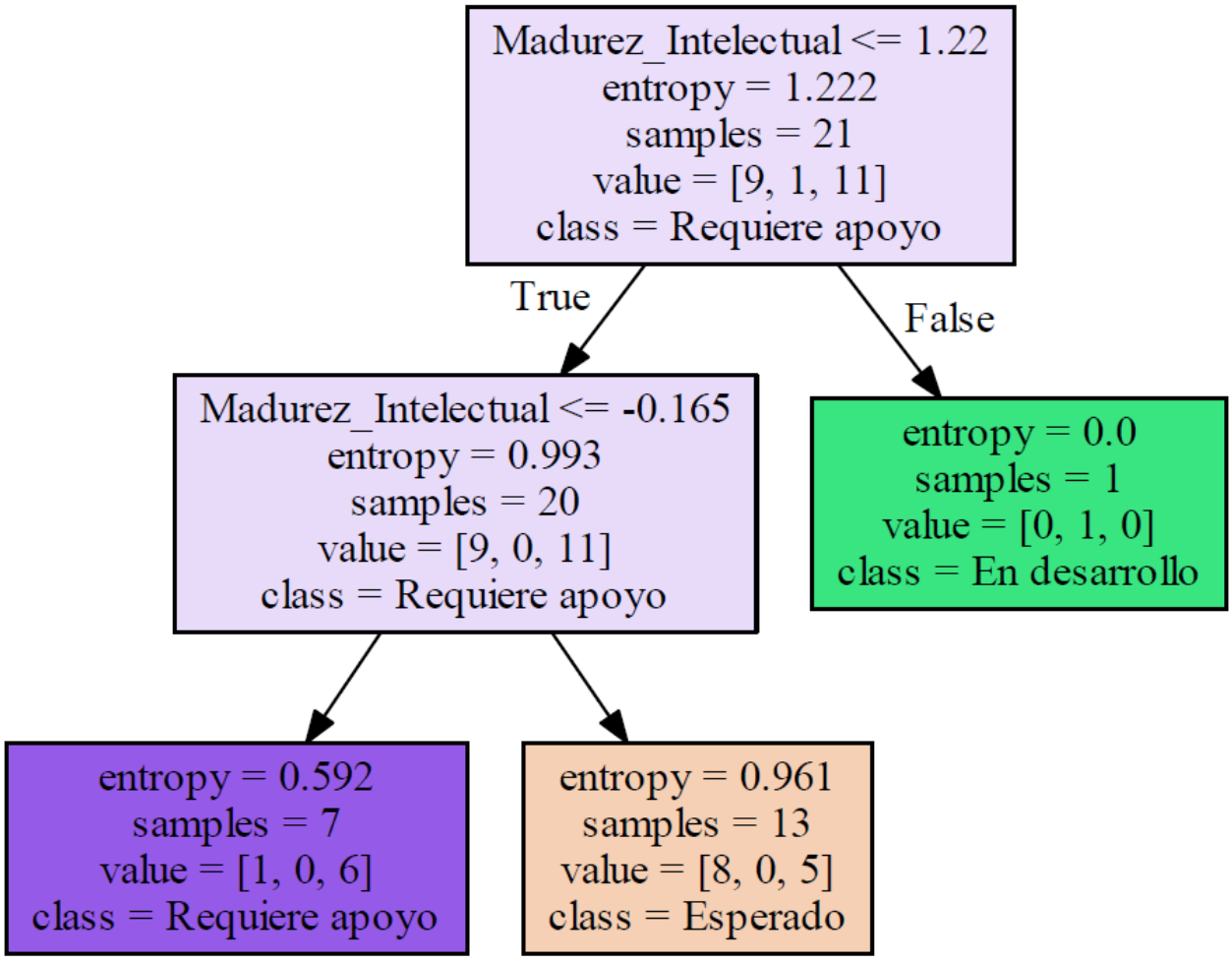


Figura 18. Representación del árbol de clasificación.
Fuente: Elaboración propia en Python, 2020.

Considerando las características únicas de nuestros datos se seleccionó evaluar con nuestro modelo de aprendizaje automático la Habilidad Básica del Pensamiento de Calculo Mental compuesta con 3 etiquetas respectivas.

La interpretación de nuestro clasificador se realizó de la siguiente forma:

En el nodo final ubicado del lado derecho en el segundo nivel, dice entropía = 0, muestras = 1 y value = [0, 1, 0] es un nodo de color verde que indica que los alumnos de Madurez Intelectual mayores a -0.165 o mayores a 1.22 tendrán una tendencia a clasificarse en el nivel "En desarrollo".

3.3.2 Matriz de confusión del árbol de decisión

Así mismo, se visualizó la matriz de confusión correspondiente (Figura 9):

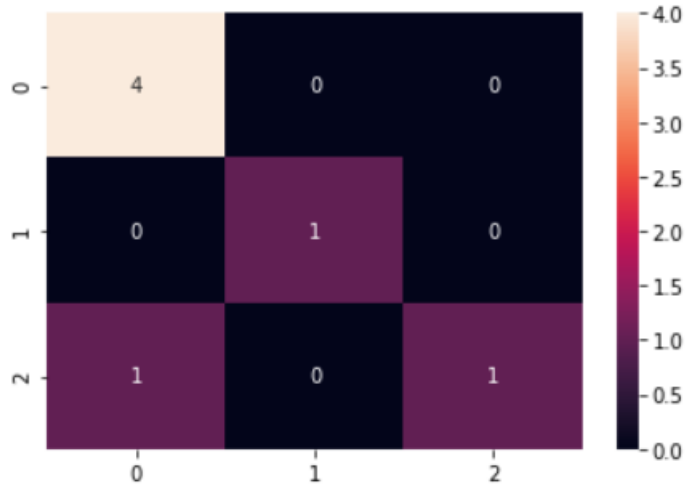
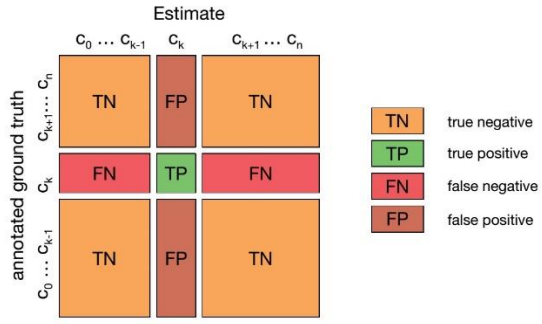


Figura 19. Matriz de confusión del árbol generado.
Fuente: Elaboración propia en Python, 2020.

Que representa lo siguiente:



Como podemos darnos cuenta tenemos 1 predicción verdadero positivo, 6 verdadero negativo

Figura 20. Descripción de una matriz de confusión.
Fuente: Krüger, 2016.

```
Puntajes de precisión para cada clase: [0.8 1.  1.]
```

Y relación total de TP / (TP + FP)

```
Precisión: 0.8571428571428571
```

3.3.3 Evaluación de la calidad de las predicciones de los modelos aprendidos

Se evaluó la calidad de entrenamiento de los clasificadores en base a un estudio en función de la variable objetivo “Cálculo mental”, no sin antes mencionar que al estar desbalanceada la representación de las clases y al haber poca data para realizar alguna técnica de oversampling, para no afectar el rendimiento de los clasificadores, se decidió agrupar las dos clases superiores en una sola y desplegar las métricas de evaluación para cada uno de los modelos aprendidos.

Las métricas aplicadas fueron:

- Puntaje de exactitud
- Precisión
- Exhaustividad
- Puntaje F1
- Soporte
- Macro promedio
- Promedio ponderado

3.3.4 Evaluación de Dummy Classifier

En el estudio se inició con la regla base que es siempre partir de lo más básico e ir complejizando. En primer lugar, se aplicó el algoritmo Dummy Classifier el cual es un clasificador que hace predicciones usando reglas simples. Es útil como una línea de base simple para comparar con otros clasificadores (reales), pero no debe usarse para problemas reales. [7] En él se aplica una técnica de estratificación:

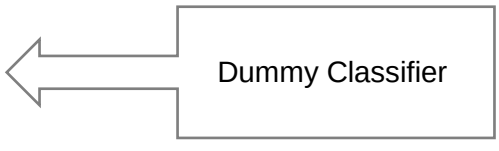
```
Model=DummyClassifier(strategy='stratified')
Accuracy Score: 0.333

----------------Classification Report-------------------------------------
               precision    recall  f1-score   support

 En desarrollo       0.00      0.00      0.00         1
Requiere apoyo       0.67      0.40      0.50         5

      accuracy                           0.33         6
     macro avg       0.33      0.20      0.25         6
  weighted avg       0.56      0.33      0.42         6

------------------------------------------------------------------------
```



3.3.5 Evaluación de Regresión de Logística

Posteriormente, se aplicó Regresión Logística penalizando el mayor uso de parámetros.

```
Model=LogisticRegression(max_iter=1000, penalty='l1', solver='liblinear')
Accuracy Score: 0.167

----------------Classification Report-------------------------------------
                precision    recall  f1-score   support

 En desarrollo       0.17      1.00      0.29         1
Requiere apoyo       0.00      0.00      0.00         5

      accuracy                           0.17         6
     macro avg       0.08      0.50      0.14         6
  weighted avg       0.03      0.17      0.05         6

--------------------------------------------------------------------------
```

Así como la Regresión Logística penalizando con ElasticNet y el solucionador Saga, balanceando el uso de más parámetros. El resultado indica que no afecta en nada y no es diferente de un Lasso Regressor, como el descrito anteriormente.

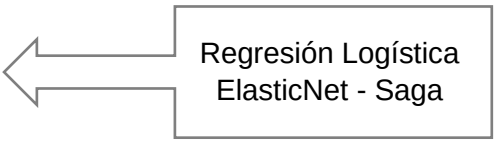
```
Model=LogisticRegression(l1_ratio=0.6, max_iter=10000, n_jobs=-1,
                   penalty='elasticnet', solver='saga')
Accuracy Score: 0.500

----------------Classification Report-------------------------------------
                precision    recall  f1-score   support

 En desarrollo       0.25      1.00      0.40         1
Requiere apoyo       1.00      0.40      0.57         5

      accuracy                           0.50         6
     macro avg       0.62      0.70      0.49         6
  weighted avg       0.88      0.50      0.54         6

--------------------------------------------------------------------------
```



3.3.6 Evaluación de Random Forest

Los bosques aleatorios (Random Forest) corresponden a una combinación de predictores de árboles de manera que cada árbol depende de los valores de un vector aleatorio muestreado de forma independiente y con el mismo la distribución para todos los árboles del bosque. [22] El resultado de la evaluación de este algoritmo se muestra en el siguiente reporte:

```
Model=RandomForestClassifier(n_jobs=-1)
Accuracy Score: 0.833

----------------Classification Report-----------------------------------
               precision    recall  f1-score   support

 En desarrollo       0.50      1.00      0.67         1
Requiere apoyo       1.00      0.80      0.89         5

      accuracy                           0.83         6
     macro avg       0.75      0.90      0.78         6
  weighted avg       0.92      0.83      0.85         6

------------------------------------------------------------------------
```

3.3.7 Evaluación de XGBoost

Por último, se implementó la métrica para XGBoost, el cual es un algoritmo innovador de los más usados en la actualidad. Su innovación radica en ser eficiente, flexible, portátil, optimizado, además de que resuelve muchos problemas de ciencia de datos de manera rápida y precisa.

```
Model=XGBClassifier(base_score=None, booster=None, colsample_bylevel=None,
              colsample_bynode=None, colsample_bytree=None, gamma=None,
              gpu_id=None, importance_type='gain', interaction_constraints=None,
              learning_rate=None, max_delta_step=None, max_depth=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=-1, num_parallel_tree=None,
              random_state=None, reg_alpha=None, reg_lambda=None,
              scale_pos_weight=None, subsample=None, tree_method=None,
              validate_parameters=None, verbosity=None)
Accuracy Score: 0.667

----------------Classification Report-----------------------------------
               precision    recall  f1-score   support

 En desarrollo       0.33      1.00      0.50         1
Requiere apoyo       1.00      0.60      0.75         5

      accuracy                           0.67         6
     macro avg       0.67      0.80      0.62         6
  weighted avg       0.89      0.67      0.71         6

------------------------------------------------------------------------
```

3.3.8 Diez variables más importantes para el modelo

```
Variable: Madurez_Intelectual_Normal Importance: 0.1
Variable: Situacion_Emocional_Buena Importance: 0.08
Variable: Situacion_Emocional_Regular Importance: 0.08
Variable: Participacion_Aveces Importance: 0.08
Variable: Participacion_Si     Importance: 0.08
Variable: Madurez_Intelectual_Bajo Importance: 0.07
Variable: Situacion_Familiar_Estable Importance: 0.07
Variable: Estilos_Aprendizaje_Kinestesico Importance: 0.06
Variable: Estilos_Aprendizaje_Visual Importance: 0.05
Variable: Madurez_Intelectual_Superior Importance: 0.05
```

3.3.9 Importancia de variables visualizadas contra su margen de error

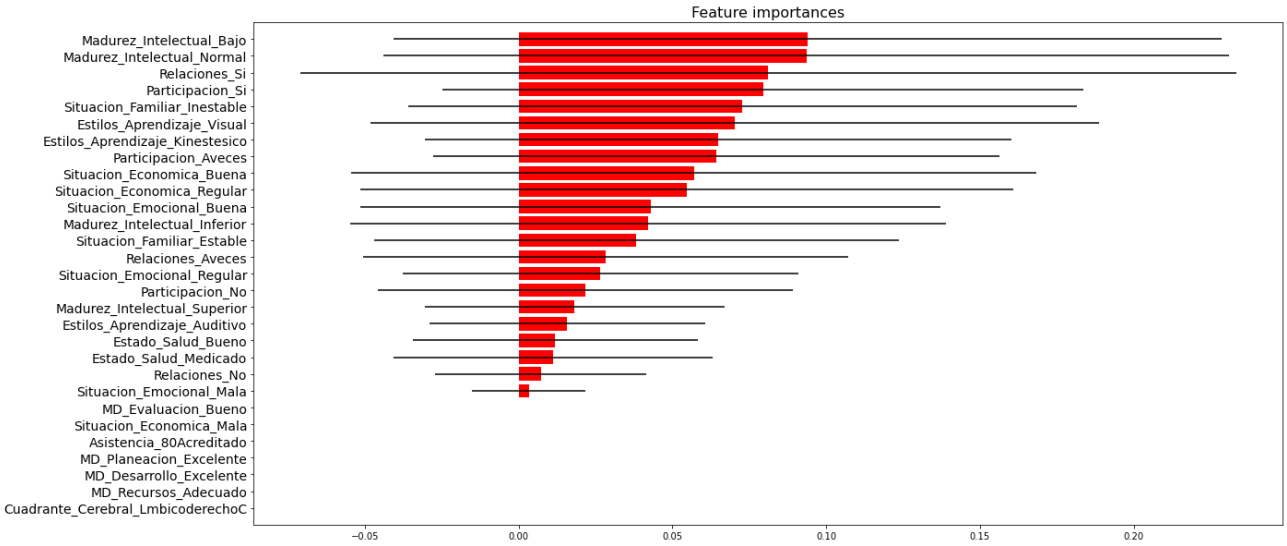


Figura 21. Atributos más importantes.
Fuente: Elaboración propia en Python, 2020.

3.3.10 Predicción de nuevos ejemplos no etiquetados

A continuación, se presenta un caso de estudio particular en la tabla 5 basándonos en un escenario donde se aplicó el diagnóstico a un alumno de nuevo ingreso:

Tabla 6. Diagnóstico de alumno de nuevo ingreso
Fuente: Elaboración propia

Estilos Aprendizaje	Madurez Intelectual	Relaciones	Situación Económica	Estado Salud	Situación Emocional	Situación Familiar	Asistencia	Participación
Visual	Bajo	No	Regular	Medicado	Regular	Estable	>= 80 Acreditado	A veces

Así mismo, se realizó un diagnóstico de la metodología y cuadrante cerebral del docente frente a grupo al cual pertenece el alumno:

Tabla 7. Diagnóstico de la metodología y cuadrante cerebral del docente frente a grupo.
Fuente: Elaboración propia

Planeación	Desarrollo	Recursos	Evaluación	Cuadrante Cerebral
Bueno	Excelente	Repetitivo	Excelente	Cortical Izquierdo [A]

Los resultados de la predicción fueron:

```
En el target Calculo_Mental, el modelo LogisticRegression diagnóstica que para el alumno: 0 -> Requiere apoyo
En el target Calculo_Mental, el modelo LogisticRegression diagnóstica que para el alumno: 0 -> Requiere apoyo
En el target Calculo_Mental, el modelo RandomForestClassifier diagnóstica que para el alumno: 0 -> Requiere apoyo
En el target Calculo_Mental, el modelo XGBClassifier diagnóstica que para el alumno: 0 -> Requiere apoyo
```

- El 0 es un ID de ejemplo, si se quisiera predecir un grupo de alumnos la representación sería de 0 hasta n, siendo n el número máximo de alumnos en el grupo.
- Es necesario destacar que se aplicó dos configuraciones de Regresión Logística (L1 LibLinear y ElasticNet Saga) aunque su representación literal mostrada anteriormente tenga una denotación ambigua al momento de visualizar la predicción.

Finalmente se predicen las probabilidades para las clases asociadas a las variables objetivo de nuestro caso de estudio.

```
En el target Calculo_Mental, el modelo LogisticRegression diagnóstica que para el alumno: 0 -> [0.36119217 0.63880783]
En el target Calculo_Mental, el modelo LogisticRegression diagnóstica que para el alumno: 0 -> [0.29148298 0.70851702]
En el target Calculo_Mental, el modelo RandomForestClassifier diagnóstica que para el alumno: 0 -> [0.14833333 0.85166667]
En el target Calculo_Mental, el modelo XGBClassifier diagnóstica que para el alumno: 0 -> [0.06938422 0.9306158 ]
```

3.4 COMPROBACIÓN DE LA HIPÓTESIS

Fue necesario aplicar estrategias para resolver desequilibrio de datos en Python con la librería imbalanced-learn, debido a que al momento de explorar los datos previamente, en el dataset de entrenamiento se tiene algunas de las clases de muestra es una clase “minoritaria”, lo cual quiere decir que tenemos pocas muestras o ejemplos. Esto provoca un desbalanceo en los datos que se utilizan para el entrenamiento del modelo de aprendizaje automático. Sin embargo, esto no afecta la eficiencia de nuestro modelo en producción debido a que a mayor número de ejemplos el desequilibrio se desvanece, esto da como resultado una mayor precisión en las predicciones, evitando así en menor medida tener problemas de sobreajuste (overfitting) y subajuste (underfitting) que pueden afectar los resultados al no generalizar correctamente los datos de entrenamiento o cualquier dato del dominio del problema. [23] Del mismo modo, es importante considerar que los árboles de decisión tienen tendencia al sobreajuste.

Con este estudio se comprobó en base al factor predictor la hipótesis planteada al inicio de esta investigación:

> “Es posible determinar el indicador de rendimiento de un alumno utilizando de forma predictiva un modelo de Aprendizaje Automático para definir las necesidades de acciones de forma anticipada.”

IV RESULTADOS Y CONCLUSIONES

Los resultados del modelo de aprendizaje automático supervisado nos llevaron a la conclusión de la importancia que el análisis de datos ofrece a la toma de decisiones estratégicas en la aplicación de metodologías apropiadas al aprendizaje y que aporten valor a la Ruta de Mejora Escolar permitiendo crear un plan de intervención anticipado que mejore el rendimiento escolar de los alumnos, elevando sus niveles de desempeño en las Habilidades Básicas del Pensamiento.

Los árboles clasificadores permitieron tomar decisiones eficaces y eficientes si son configurados adecuadamente. Es importante tener las suficientes muestras posibles para optimizar los resultados. Por otra parte, existe una gran variedad de algoritmos clasificadores, que permiten generar otros tipos de resultados que pueden ser usados para discernir entre el más óptimo para un caso de estudio determinado.

Una observación que se tuvo al implementar los algoritmos clasificadores es la dificultad para determinar un aproximador para las 3 clases ya que la clase Esperado está muy bajamente representada en la data de muestra, pero es aplicable al escalar la cantidad de data. El mínimo requerido que se sugiere es tener al menos unos 100 registros como entrada al modelo.

Una propuesta a mencionar como resultado del análisis de datos es que para la problemática tratada debido a la escasez de data es mejor usar un modelo no paramétrico (Random Forest), pero con el aumento de la misma puede ser relevante usar XGBoost.

Una muy buena observación que se puede realizar del modelo de aprendizaje automático desarrollado es la siguiente:

- Con más entrada de datos a nuestro modelo se puede mejorar el intervalo de confianza.

A raíz de lo anterior se puede deducir lo siguiente:

- El estudio realizado en la presente investigación da evidencia suficiente para establecer una hipótesis, la cual deberia validarse con una muestra más grande.
- Es importante mencionar que la desviación estándar tendrá un valor muy alto debido a la escasez de data como entramiento del modelo. Si se desea obtener mayor representatividad en el estudio es necesario generar data de mejor calidad, sin embargo, las correlaciones entregadas se pueden expandir utilizando técnicas como IV (Information Values) la cual es una metodología estadística para encontrar la importancia de features, al igual que Predictive Power Score (PPS) y SKF (Stratified K-Fold) en el cual se pueden iterar distintos “pliegues” de la data para generar más escenarios en caso de tener poca data, e incluso, usar el mismo fold de la data con baja representatividad.

Se espera que esta investigación sea un aliciente para promover la aplicación de Inteligencia Artificial en herramientas de software para la toma de decisiones en todos los sectores, pero principalmente en aquellos más vulnerables donde se potencialice y mejore la calidad de vida de las personas.

REFERENCIAS BIBLIOGRÁFICAS

[1] F. Ramírez, “Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957,” 2018. [Online]. Available: https://empresas.blogthinkbig.com/historia-de-la-ia-frank-rosenblatt-y-e/.

[2] D. Delen, “A comparative analysis of machine learning techniques for student retention management,” *Decis. Support Syst.*, vol. 49, no. 4, pp. 498–506, 2010, doi: 10.1016/j.dss.2010.06.003.

[3] K. B. Eckert and R. Suénaga, “Analysis of Attrition-Retention of College Students Using Classification Technique in Data Mining,” *Form. Univ.*, vol. 14, no. 5, pp. 3–12, 2015, doi: 10.4067/S0718-50062015000500002.

[4] J. Hernández Cáceres and W. Alejandro Rojas Calvo Director, “Clustering basado en el algoritmo K-means para la identificación de grupos de pacientes quirúrgicos Clustering technique based on k-means algorithm for the identification of clusters of surgical patients,” 2016.

[5] M. Learning, “Curso intensivo de Machine Learning,” 2019. [Online]. Available: https://developers.google.com/machine-learning/crash-course. [Accessed: 09-Oct-2019].

[6] E. Puertas Sanz and others, “Ingeniería de Atributos y Minería de Datos para la Recuperación de Información con Adversario,” 2013.

[7] Scikit-Learn, “DummyClassifier,” 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html.

[8] S. Learn, “Métricas y puntuación: cuantificando la calidad de las predicciones.” [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html#dummy-estimators.

[9] S. de la F. Hernandez, “Regresion logistica,” *Fac. Ciencias Econ. y Empres. UNAM*, 2011.

[10] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, “LIBLINEAR: A library for large linear classification,” *J. Mach. Learn. Res.*, vol. 9, no. 2008, pp. 1871–1874, 2008, doi: 10.1145/1390681.1442794.

[11] I. Chaos, “ElasticNET,” 2020. [Online]. Available:

https://www.interactivechaos.com/manual/tutorial-de-machine-learning/elastic-net.

[12] C. Leistner and H. Bischof, “On-line Random Forests,” pp. 1393–1400, 2009.

[13] A. K. Agarwal, S. Wadhwa, and S. Chandra, “XGBoost: A Scalable Tree Boosting System,” *J. Assoc. Physicians India*, vol. 42, no. 8, p. 665, 1994.

[14] A. Gustavo, “Las Máquinas de Soporte Vectorial (SVMs),” no. January 2005, 2014.

[15] 2011 Bruce, “Encyclopedia of Machine Learning,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.

[16] K. Bengoetxea, A. Atutxa, and M. Iruskieta, “Un detector de la unidad central de un texto basado en técnicas de aprendizaje automático en textos científicos para el euskera,” *Proces. Leng. Nat.*, vol. 58, pp. 37–44, 2017.

[17] N. Series, “Measuring the Accuracy of Diagnostic Systems Author(s): John A. Swets Source:,” *Science (80-.).*, vol. 240, no. 4857, pp. 1285–1293, 2015.

[18] K. Schwaber, *Agile Project Management with Scrum*, no. Cmm. 2004.

[19] R. J. Zeballos D., “Aplicando scrum,” *Rev. Investig. y Tecnol.*, vol. 1, pp. 125–132, 2012.

[20] K. Schwaber, “SCRUM Development Process,” no. February 1986, 1997.

[21] F. F. R. C. Mariuxi Paola Zea Ordóñez, Jimmy Rolando Molina Ríos, “Administración de Base de Datos en PostgreSQL,” 2019. [Online]. Available: https://play.google.com/books/reader?id=5-mkDgAAQBAJ&hl=es&pg=GBS.PA1. [Accessed: 16-Nov-2019].

[22] L. Breiman, “Random Forest,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1017/CBO9781107415324.004.

[23] W. M. P. Van Der Aalst, V. Rubin, H. M. W. Verbeek, B. F. Van Dongen, E. Kindler, and C. W. Günther, *Process mining: A two-step approach to balance between underfitting and overfitting*, vol. 9, no. 1. 2010.