Trabajo Práctico Final
Minería de datos aplicada a bases educativas APRENDER

Violi Pablo Ezequiel

Tutor: Pablo Pytel

Licenciatura en Sistemas
Departamento de Desarrollo Productivo y Tecnológico

2021

Agradecimientos

Página dedicada a agradecimientos.

Abstract

La minería de datos puede definirse como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, a partir de grandes volúmenes de datos. Hoy en día, las distintas fuentes de información disponibles en un mismo negocio, generan la necesidad de buscar nuevos procesos para poder transformar dichos datos en información útil para las organizaciones. El rápido crecimiento en la capacidad para almacenar datos que están experimentando los procesos de recolección de información sobre la población, proporciona nuevas posibilidades para analizar su comportamiento. En el siguiente trabajo aplicamos dichas técnicas de minería de datos, en la encuesta Aprender realizada por el Ministerio de Educación Anualmente en los Colegios de la nación argentina. Nuestro objetivo, es detectar tempranamente aquellos alumnos que performarán regular, y sobresaliente en los exámenes de Matemáticas, poder agruparlos y describir sus características más influyentes. Los resultados de este trabajo pueden aplicarse para generar un espacio de ayuda y apoyo en las primeras etapas del Alumno, para acompañarlos y ayudarlos en la enseñanza. Por otro lado, este trabajo puede detectar aquellos Alumnos con perfiles orientados a ciencias exactas, para poder darles ayuda pedagógica y mostrarle los posibles caminos universitarios que puede seguir en su futuro.

Índice

Índice de figuras

1.Introducción

El aumento del volumen y variedad de información que se encuentra almacenada en grandes bases de datos digitales y otras fuentes ha crecido enormemente en las últimas décadas. Gran parte de esta información es histórica, es decir, representa transacciones o situaciones que se han producido en un periodo determinado.

En el mundo actual en que vivimos donde cada vez es más importante tener todo informatizado y cuantificado en las bases de datos de cada empresa u organización, surge la necesidad de encontrar alguna manera de sacar conclusiones a partir de estos datos, ya que de por sí solos los datos nada más que serían registros sin significado y que no darían ningún tipo de información valiosa que se pudiera explotar y sacar provecho ellos.

Para comenzar un proceso de minería de datos, es importante partir de una base de datos sólida, almacenada en un data warehouse seguro (almacén de datos). Los datos deben estar correctamente estructurados.

Existen diversas técnicas y metodologías de minería de datos que se pueden utilizar y que pueden ser más o menos adecuadas para cada caso en concreto, pero en el presente proyecto se ha elegido la metodología CRISP-DM de la cual hablaremos detalladamente más adelante y trataremos de justificar el porqué de esta elección para el caso práctico que se nos planteó: la explotación de los datos contenidos en la encuesta de Aprender del Ministerio de Educación, referido al desempeño de los alumnos en el examen de matemática.

El trabajo está divido en cuatro capítulos, diferenciados por el objetivo de los mismos.

El primer capítulo (Marco Teórico), contiene información teórica que rodea al proyecto, para comprender la dimensión, el contexto del mismo e información referente a la actualidad

El segundo capítulo (Marco Metodológico), abarca temas referentes a la minería de datos, herramientas utilizadas durante el proyecto, y metodologías abordadas.

En el tercer capítulo (Modelo Predictivo), se aplican las metodologías previamente explicadas a las bases de encuestas Aprender, realizando el paso a paso hasta la última etapa de implementación del proyecto.

En el cuarto capítulo (Líneas finales), se incluye la conclusión del proyecto y los avances esperados a futuro.

Por último, se presenta toda la referencia bibliográfica utilizada durante el proyecto, y un anexo con la base de las encuestas Aprender

2.Marco Teórico

A continuación, se detallará los principales conceptos teóricos que se aplicarán en el presente trabajo:

2.1 Ministerio de Educación

La información que produce el Ministerio es una herramienta básica para la planificación de la educación Nacional, así como para las investigaciones y proyecciones que se realizan en los ámbitos académico y privado (Indec, s.f.).

2.2 Encuestas Aprender

La encuesta Aprender es el dispositivo nacional de evaluación de los aprendizajes de los estudiantes y de sistematización de información acerca de algunas condiciones en las que ellos se desarrollan

Aprender busca obtener y generar información oportuna y de calidad para conocer mejor los logros alcanzados y los desafíos pendientes en torno a los aprendizajes de los estudiantes para contribuir a procesos de mejora educativa continua.

Con los resultados de la encuesta, se trata de caracterizar a la población en términos de

- Reporte por escuela
- Reporte Nacional de resultados
- Reportes jurisdiccionales y regionales
- Sistema Abierto de Consulta
- Presentación Interactiva de Datos

La evaluación fue desarrollada por el Ministerio de Educación, Cultura, Ciencia y Tecnología, a través de la Secretaría de Evaluación Educativa, en acuerdo con el Consejo Federal de Educación y con la participación del Cuerpo Colegiado Federal de docentes y especialistas de todo el país.Por otro lado, Gentile (2015) afirma:

El cuestionario complementario indaga en información que permite analizar los logros de aprendizaje en clave de contexto. De esta forma, Aprender brinda información sobre clima escolar, autopercepción del estudiante, prácticas educativas y uso de tecnología, entre otras informaciones.

La encuesta se realiza anualmente desde el 2016 en adelante, para nivel primario y secundario en todos los colegios del país.

2.3 Explotación de información.

Según Mitra & Acharya (2003), la revolución digital ha hecho posible que la información digitalizada sea fácil de capturar, procesar, almacenar, distribuir, y transmitir Con el importante progreso en informática y en las tecnologías relacionadas y la expansión de su uso en diferentes aspectos de la vida, se continúa recogiendo y almacenando en bases de datos gran cantidad de información. Descubrir conocimiento de este enorme volumen de datos es un reto en sí mismo. La minería de datos es un intento de buscarle sentido a la explosión de información que actualmente puede ser almacenada

Las tareas propias de la fase de minería de datos pueden ser descriptivas, o predictivas. En otras palabras, es un campo interdisciplinar con el objetivo general de predecir las salidas y revelar relaciones en los datos (Mitra & Acharya, 2003). Para ello se utilizan herramientas automáticas que (a) emplean algoritmos sofisticados para descubrir principalmente patrones ocultos, asociaciones, anomalías, y/o estructuras de la gran cantidad de datos almacenados en los data warehouses u otros repositorios de información, y (b) filtran la información necesaria de las grandes bases de datos (Riquelme, Ruiz, & Gilbert, 2006)

La explotación de información es la sub-disciplina informática que aporta a la inteligencia de negocio (Negash & Gray, 2008) las herramientas para la transformación de información en conocimiento (Langseth & Vivatrat, 2003)Se ha definido como la búsqueda de patrones interesantes y de regularidades importantes en grandes masas de información (Grigori, y otros, 2004). Al hablar de explotación de información basada en sistemas inteligentes (Michalski, Bratko, & Kubat, 1998) se refiere específicamente a la aplicación de métodos de sistemas inteligentes, para descubrir y enumerar patrones presentes en la información. Los métodos basados en sistemas inteligentes (Kononenko & Cestnik, 1986) permiten obtener resultados de análisis de la masa de

información que los métodos convencionales (Michalski R. , 1983) no logran tales como: los algoritmos de clasificación, los algoritmos de agrupación, y los de ponderación

2.4 Evolución Minería de datos

La idea de Minería de Datos no es nueva. Ya desde los años sesenta los estadísticos manejaban términos como Data Fishing, Data Mining (DM) o Data Archaeology con la idea de encontrar correlaciones sin una hipótesis previa en bases de datos con ruido.

A principios de los años ochenta, Rakesh Agrawal, GioWiederhold, Robert Blum y Gregory Piatetsky-Shapiro entre otros, empezaron a consolidar los términos de Minería de Datos y KDD.

Esta tecnología ha sido un buen punto de encuentro entre personas pertenecientes al ámbito académico y al de los negocios.

La evolución de sus herramientas en el transcurso del tiempo puede dividirse en cuatro etapas principales:

- Colección de Datos (1960).
- Acceso de Datos (1980).
- Almacén de Datos y Apoyo a las Decisiones (principios de la década de 1990).
- Minería de Datos Inteligente. (finales de la década de 1990).

Las técnicas de Data Science o Data Analytics, que tanto interés despiertan hoy en día, en realidad surgieron en la década de los 90, cuando se usaba el término KDD (Knowledge Discovery in Databases) para referirse al (amplio) concepto de hallar conocimiento en los datos

Algunas de las tareas importantes de la minería de datos incluyen la identificación de aplicaciones para las técnicas existentes, y desarrollar nuevas técnicas para dominios tradicionales o de nueva aplicación, como el comercio electrónico y la bioinformática. Existen numerosas áreas donde la minería de datos se puede aplicar, siendo un gran referente en las siguientes especializaciones: (Riquelme, Ruiz, & Gilbert, 2006):

- Comercio y banca: segmentación de clientes, previsión de ventas, análisis de riesgo.

- Medicina y Farmacia: diagnóstico de enfermedades y la efectividad de los tratamientos.
- Seguridad y detección de fraude: reconocimiento facial, identificaciones biométricas, accesos a redes no permitidos, etc.
- Recuperación de información no numérica: minería de texto, minería web, búsqueda e identificación de imagen, video, voz y texto de bases de datos multimedia.
- Astronomía: identificación de nuevas estrellas y galaxias.
- Geología, minería, agricultura y pesca: identificación de áreas de uso para distintos cultivos o de pesca o de explotación minera en bases de datos de imágenes de satélites
- Ciencias Ambientales: identificación de modelos de funcionamiento de ecosistemas naturales y/o artificiales (p.e. plantas depuradoras de aguas residuales) para mejorar su observación, gestión y/o control.
- Ciencias Sociales: Estudio de los flujos de la opinión pública. Planificación de ciudades: identificar barrios con conflicto en función de valores sociodemográficos.

En la actualidad se puede afirmar que la MD ha demostrado la validez de una primera generación de algoritmos mediante diferentes aplicaciones al mundo real. Sin embargo, estas técnicas todavía están limitadas por bases de datos simples, donde los datos se describen mediante atributos numéricos o simbólicos, no conteniendo atributos de tipo texto o imágenes, y los datos se preparan con una tarea concreta en mente. Sobrepasar este límite será un reto a conseguir (Riquelme, Ruiz, & Gilbert, 2006):

Existen cientos de productos de minería de datos y de compañías de consultoría. KDNuggets (kdnuggets.com) tiene una lista de estas compañías y sus productos en el campo de la minería de datos. Pueden resaltarse por su mayor expansión las siguientes: SAS con SAS Script y SAS Enterprise Miner; SPSS y el paquete de minería Clementine; IBM con Intelligent Miner; Microsoft incluye características de minería de datos en las bases de datos relacionales; otras compañías son Oracle, Angoss y Kxen. En la línea del software libre Weka (Witten & Frank, 2005) es un producto con mayor orientación a las técnicas provenientes de la IA, pero de fuerte impacto.

2.5 Surgimiento CRISP-DM & SEMMA

Como se mencionó anteriormente, la minería de datos surgió a principio de los años 70, y fue evolucionando y volviéndose más grande. En un intento de normalización de este proceso de descubrimiento de conocimiento, de forma similar a como se hace en ingeniería software para normalizar el proceso de desarrollo software, surgieron a finales de los 90 dos metodologías principales: CRISP-DM (Cross Industry Standard Process for Data Mining) y SEMMA (Sample, Explore, Modify, Model, and Assess). Ambas especifican las tareas a realizar en cada fase descrita por el proceso, asignando tareas concretas y definiendo lo que es deseable obtener tras cada fase.

Azevedo y Santos (2008) comparan ambas implementaciones, llegando a la siguiente conclusión:

> "Llegamos a la conclusión de que SEMMA y CRISP-DM pueden verse como una implementación del proceso KDD descrito por (Fayyad, 1996). A primera vista, podemos llegar a la conclusión de que CRISP-DM es más completo que SEMMA. Sin embargo, analizándolo más a fondo, podemos integrar el desarrollo de una comprensión del dominio de la aplicación, el conocimiento previo relevante y los objetivos del usuario final, en la etapa de Muestra de SEMMA, porque los datos no pueden muestrearse a menos que exista un Verdadera comprensión de todos los aspectos presentados. Con respecto a la consolidación al incorporar este conocimiento al sistema, podemos suponer que está presente, porque es realmente la razón para hacerlo. Esto lleva al hecho de que se han alcanzado los estándares, en relación con el proceso general: SEMMA y CRISP-DM guían a las personas a saber cómo se puede aplicar la DM en la práctica en sistemas reales. En el futuro, pretendemos analizar otros aspectos relacionados con los estándares de DM, a saber, lenguajes basados en SQL para DM, así como lenguajes basados en XML para DM. Como complemento, pretendemos investigar la existencia de otros estándares para DM"

2.6 Herramientas

En esta sección, se enumeran y describen brevemente las herramientas utilizadas durante el proyecto

2.6.1 Python

Python es un lenguaje de escritura rápido, escalable, robusta y de código abierto, ventajas que hacen de Python un aliado perfecto para la Inteligencia Artificial.

Permite plasmar ideas complejas con unas pocas líneas de código, lo que no es posible con otros lenguajes. Existen bibliotecas como “Keras” y “TensorFlow”, que contienen mucha información sobre las funcionalidades del aprendizaje automático. Además, existen bibliotecas proporcionadas por Python, que se usan mucho en los algoritmos AI como Scikitl, una biblioteca gratuita de aprendizaje automático que presenta varios algoritmos de regresión, clasificación y agrupamiento.

Pero, sobre todo, Python es un lenguaje gratuito de código abierto con una gran comunidad en activo, que proporciona soporte a cualquier programador. Todas estas razones combinadas, hacen que aprender Python sea una opción fácil sobre otros lenguajes para aplicaciones de inteligencia artificial (Soloaga, 2018).

Dentro del lenguaje, se utilizaron las siguientes librerías:

- Pandas: es una de las librerías de python más útiles para los científicos de datos. Las estructuras de datos principales en pandas son Series para datos en una dimensión y DataFrame para datos en dos dimensiones. Estas son las estructuras de datos más usadas en muchos campos tales como finanzas, estadística, ciencias sociales y muchas áreas de ingeniería. Pandas destaca por lo fácil y flexible que hace la manipulación de datos y el análisis de datos.

- NumPy: proporciona una estructura de datos universal que posibilita el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos. Además, esta librería proporciona funciones matemáticas de alto nivel que operan en estas estructuras de datosSeaborn

- Matplotlib: es una librería de gráficos, desde histogramas, hasta gráficos de líneas o mapas de calor. También se pueden usar comandos de Latex para agregar expresiones matemáticas a tu gráfica.

- Seaborn: es una librería gráfica basada en matplotlib, especializada en la visualización de datos estadísticos. Se caracteriza por ofrecer un interfaz de alto nivel para crear gráficos estadísticos visualmente atractivos e informativos. Seaborn considera la visualización como un aspecto fundamental a la hora de explorar y entender los datos. Se integra muy bien con la librería de manipulación de datos pandas.

- Sklearn: para machine learning: Construida sobre NumPy, SciPy y matplotlib, esta librería contiene un gran número de eficientes herramientas para machine learning y modelado estadístico, como por ejemplo, algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad.

2.6.2 Jupyter

Jupyter Notebook es una interfaz web de código abierto que permite la inclusión de texto, vídeo, audio, imágenes, así como la ejecución de código a través del navegador en múltiples lenguajes. Esta ejecución se realiza mediante la comunicación con un núcleo (Kernel) de cálculo. Aunque en principio, el equipo de desarrollo de Jupyter Notebook incluye por defecto únicamente el núcleo de cálculo Python, el carácter abierto del proyecto ha permitido aumentar el número de núcleos disponibles [1], incluyendo, por ejemplo Octave, Julia, R, Haskell, Ruby, C/C++, Fortran, Java, SageMath, Scala, o también Matlab y Mathematica. Esta interfaz, agnóstica del lenguaje (de ahí su nombre al unir 3 de los lenguajes de programación de código abierto más utilizados en el ámbito científico: Ju-lia, Py-thon y R), puede suponer por tanto una estandarización para mostrar el contenido de cursos científicos, sin encontrarse limitado a la adopción de un único lenguaje. (Granado)

3.Marco Metodológico

En el siguiente capítulo, se verá en detalle las metodologías que luego serán aplicadas en el proyecto.

3.1 Procesos de Extracción y Manipulación de los Datos

Para realizar descubrimientos de conocimientos en volúmenes de datos es necesario contar con un marco de trabajo que permita planificar y guiar el proceso de desarrollo del proyecto. Actualmente las metodologías más conocidas para el desarrollo de proyectos de minería de datos son KDD (Knowledge Discovery in Databases), CRISP-DM (Cross Industry Standard Process for Data Mining) y SEMMA (Sample, Explore, Modify, Model, Assess).

A continuación se va a detallar los pasos de cada uno de estos procesos.

3.1.1 Knowledge Discovery in databases (KDD)

Un término común en la minería de datos es el descubrimiento de conocimiento en bases de datos (KDD), que viene a ser un proceso iterativo significativo que consta de una serie de fases para la generación de conocimiento y la toma de decisiones. Las fases de KDD son:

1. Integración y recopilación. Consiste en establecer un entendimiento del dominio de la aplicación y de los conocimientos previos relevantes. En esta fase se determina también la selección de un conjunto de datos que pueden ser obtenidos de diferentes fuentes, sobre los cuales se realiza el descubrimiento.

2. Selección, limpieza y transformación. En esta etapa se seleccionan y preparan los datos que se van a minar. Sin embargo, existen factores como el ruido o valores atípicos que afectan la calidad de los datos, por lo que ante esta situación la limpieza es una de las tareas más importantes, puesto que permite la selección de la técnica que más se ajuste al problema a resolver.

3. Minería de datos. Es la fase más representativa, se determina qué tipo de tarea es la más apropiada, ya sea agrupamiento, reglas de asociación, correlación, clasificación, regresión, entre

otras. Los resultados obtenidos dependen de fases anteriores, por lo que existe la posibilidad de regresar a los pasos previos para requerir nuevos datos o para redefinir la solución al problema planteado.

4. Evaluación e interpretación. Los patrones descubiertos deben cumplir con tres propiedades: precisión, comprensibles e interesantes. En esta fase se evalúan e interpretan los patrones obtenidos. Algunas validaciones pueden ser a través de índices de evaluación, validación cruzada, matrices de confusión, entre otras.

5. Difusión y uso. Como última fase, el conocimiento descubierto debe de ser incorporado en algún sistema o simplemente documentarlo para su difusión a las partes interesadas. Este proceso incluye también la revisión y resolución de posibles conflictos con los conocimientos que anteriormente se tenía.

3.1.2 CRISP-DM

La metodología de CRISP-DM está descrita en términos de un modelo de proceso jerárquico. Chapman (2000) afirma:

> Consistente en un conjunto de tareas descritas en cuatro niveles de abstracción (de lo general a lo específico): fase, tarea genérica, tarea especializada, e instancia de procesos. (Ver la figura 1.)

En el nivel superior, el proceso de minería de datos es organizado en un número de fases; cada fase consiste de varias tareas genéricas de segundo nivel. Este segundo nivel lo llaman genérico porque está destinado a ser bastante general para cubrir todas las situaciones posibles de minería de datos. Las tareas genéricas están destinadas a ser tan completas y estables como sea posible. Completo significa que cubre tanto al proceso entero de minería de datos y todas las aplicaciones de minería de datos posibles. Estable significa que el modelo debería ser válido para acontecimientos normales y aún para desarrollos imprevistos como técnicas de modelado nuevo.

El tercer nivel, el nivel de tarea especializado, es el lugar para describir como las acciones en las tareas genéricas deberían ser realizadas en ciertas situaciones específicas. Por ejemplo, en el segundo nivel podría haber una tarea genérica llamada limpieza de datos. El tercer nivel describe

como esta tarea se diferencia en situaciones diferentes, como la limpieza de valores numéricos contra la limpieza de valores categóricos, o si el tipo de problema es agrupamiento o el modelado predictivo.

La descripción de fases y tareas como pasos discretos realizados en un orden específico representa una secuencia idealizada de eventos.

En la práctica, muchas de las tareas pueden ser realizadas en una orden diferente, y esto a menudo será necesario volver a hacer tareas anteriores repetidamente y repetir ciertas acciones. Nuestro modelo de proceso no intenta capturar todas estas posibles rutas del proceso de la minería de datos porque esto requeriría un modelo de proceso demasiado complejo.

El cuarto nivel, la instancia de proceso, es un registro de las acciones, decisiones, y de los resultados de una minería de datos real contratada.

Una instancia de proceso está organizado según las tareas definidas en los niveles más altos, pero representa lo que en realidad pasó en un contrato particular más bien que lo que pasa en general.

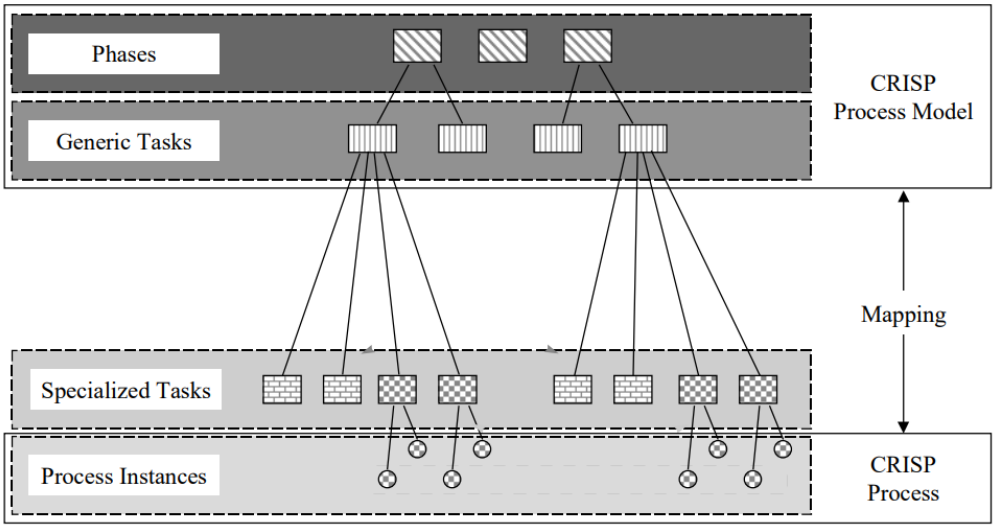

Figura 1 Four level breakdown of the CRISP-DM methodology (Chapman, y otros, 2000)

Por otro lado, el modelo de procesos de CRISP-DM nos trae los siguientes beneficios, afirma Chapman (2000):

Nos proporciona una descripción del ciclo de vida del proyecto de minería de datos. Este contiene las fases de un proyecto, sus tareas respectivas, y las relaciones entre estas tareas. En este nivel de descripción, no es posible identificar todas las relaciones. Las relaciones podrían existir entre cualquier tarea de minería de datos según los objetivos, el contexto, y –lo más importante- el interés del usuario sobre los datos.

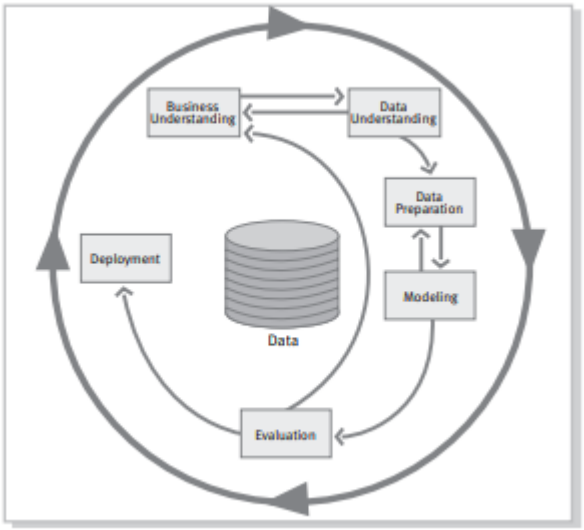

Figura 2 Phases of the CRISP-DM reference model (Chapman, y otros, 2000)

Respecto al ciclo de vida del modelo CRISP-DM, se divide en seis fases, mostradas en la figura 2. La secuencia de las fases no debe ser necesariamente rígida. Se debe avanzar y retroceder entre fases. El resultado de cada fase determina que la fase, o la tarea particular de una fase, tienen que ser realizados después. Las flechas indican las más importantes y frecuentes dependencias entre fases (Chapman, y otros, 2000)

El círculo externo en la Figura 2 simboliza la naturaleza cíclica de la minería de datos. La minería de datos no se termina una vez que la solución es desplegada. Las informaciones ocultas (lecciones cultas) durante el proceso y la solución desplegada pueden provocar nuevas, a menudo más - preguntas enfocadas en el negocio. Los procesos de minería subsecuentes se beneficiarán de las experiencias previas (Chapman, y otros, 2000)

A continuación, se perfilan las 6 fases previamente mencionadas, según Chapman (2000):

Comprensión del negocio. Esta fase inicial se enfoca en la comprensión de los objetivos de proyecto y exigencias desde una perspectiva de negocio, luego convirtiendo este conocimiento de los datos en la definición de un problema de minería de datos y en un plan preliminar diseñado para alcanzar los objetivos.

Comprensión de los datos: La fase de entendimiento de datos comienza con la colección de datos inicial y continua con las actividades que le permiten familiarizar primero con los datos, identificar los problemas de calidad de datos, descubrir los primeros conocimientos en los datos, y/o descubrir subconjuntos interesantes para formar hipótesis en cuanto a la información oculta.

Preparación de datos: La fase de preparación de datos cubre todas las actividades necesarias para construir el conjunto de datos final [los datos que serán provistos en las herramientas de modelado] de los datos en brutos iniciales. Las tareas de preparación de datos probablemente van a ser realizadas muchas veces y no en cualquier orden prescripto. Las tareas incluyen la selección de tablas, registros, y atributos, así como la transformación y la limpieza de datos para las herramientas que modelan.

Modelado: En esta fase, varias técnicas de modelado son seleccionadas y aplicadas, y sus parámetros son calibrados a valores óptimos. Típicamente hay varias técnicas para el mismo tipo de problema de minería de datos. Algunas técnicas tienen requerimientos específicos sobre la forma de datos. Por lo tanto, volver a la fase de preparación de datos es a menudo necesario.

Evaluación: En esta etapa en el proyecto, usted ha construido un modelo (o modelos) que parece tener la alta calidad de una perspectiva de análisis de datos.

Antes del proceder al despliegue final del modelo, es importante evaluar a fondo ello y la revisión de los pasos ejecutados para crearlo, para comparar el modelo correctamente obtenido con los objetivos de negocio. Un objetivo clave es determinar si hay alguna cuestión importante de negocio que no ha sido suficientemente considerada. En el final de esta fase, una decisión en el uso de los resultados de minería de datos debería ser obtenida.

Desarrollo: La creación del modelo no es generalmente el final del proyecto. Incluso si el objetivo del modelo es de aumentar el conocimiento de los datos, el conocimiento ganado tendrá que ser organizado y presentado en el modo en el que el cliente pueda usarlo. Ello a menudo implica la aplicación de modelos "vivos" dentro de un proceso de toma de decisiones de una

organización, por ejemplo, en tiempo real la personalización de página Web o la repetida obtención de bases de datos de mercadeo. Dependiendo de los requerimientos, la fase de desarrollo puede ser tan simple como la generación de un informe o tan compleja como la realización repetida de un proceso cruzado de minería de datos a través de la empresa. En muchos casos, es el cliente, no el analista de datos, quien lleva el paso de desarrollo. Sin embargo, incluso si el analista realizara el esfuerzo de despliegue, esto es importante para el cliente para entender de frente que acciones necesita para ser ejecutadas en orden para hacer uso de los modelos creados actualmente.

La figura 3 presenta un contexto de fases acompañadas por tareas genéricas y las salidas. En las secciones siguientes, describimos cada tarea genérica y sus salidas más detalladamente. Enfocamos nuestra atención en descripciones de tarea y los resúmenes de salidas.

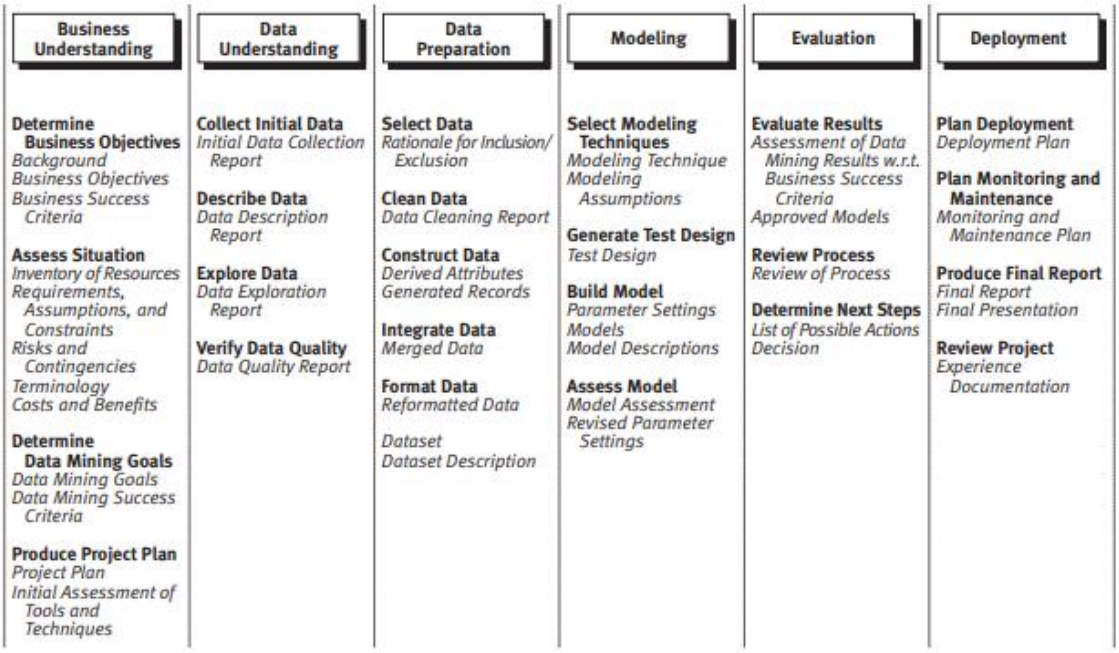

Figura 3 Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model (Chapman, y otros, 2000)

3.1.3 SEMMA

SEMMA (Sample, Explore, Modify, Model, Assess) es una metodología creada por SAS Institute (Statistical Analysis Systems), quien la define como el proceso de selección, exploración y modelado de grandes volúmenes de datos para descubrir patrones de interés (SAS, 1998). Este proceso consta de cinco etapas necesarias para guiar el desarrollo de un proyecto de minería de

datos. Sumathi y Sivananda (2006) mencionan que la metodología SEMMA permite aplicar la estadística exploratoria y técnicas de visualización de manera fácil, así como la selección y transformación de las variables más significativa, con el objetivo de crear modelos para predecir resultados y evaluarlos de manera que sirva de apoyo para la toma de decisiones.

Etapas del proceso SEMMA:

1. Muestreo. En esta etapa se toma una muestra del conjunto de datos disponible, que debe ser lo suficientemente grande para contener la información relevante, y lo suficientemente pequeña como para correr el proceso rápidamente. Esta etapa es aconsejable cuando el tamaño del conjunto de datos es demasiado extenso.

2. Exploración. Consiste en explorar los datos en búsqueda de relaciones y tendencias desconocidas. Es una etapa especial para familiarizarse con los datos, y formular nuevas hipótesis a partir de su análisis.

3. Modificación. Etapa de preparación de datos que consiste en la limpieza de los valores atípico, se realiza un tratamiento de los datos faltantes, y se seleccionan, crean y modifican las variables que servirán para la etapa del modelado.

4. Modelado. Consiste en la creación del modelo para predecir las variables, utilizando algunas de las técnicas predictivas como árboles de decisión, redes neuronales, análisis discriminante o análisis de regresión.

5. Evaluación. En esta fase se evalúa la utilidad y la exactitud de los modelos obtenidos en el proceso de minería de datos, por ejemplo analizando la capacidad predictiva de los mismos.

Una clara diferencia con respecto a otras metodologías es que en SEMMA la primera fase se inicia con el muestreo de datos. Por otra parte, SEMMA está relacionada particularmente con productos comerciales de SAS Institute

3.2 Proceso de Modelado

Los Modelos Predictivos buscan predecir el comportamiento futuro de una entidad aprendiendo de los hechos ocurridos en el pasado. Un modelo predictivo se desarrolla con datos históricos

para luego, al aplicar el algoritmo definido, intentar predecir lo que sucederá en el futuro. Por lo que el modelo contiene una variable a predecir (Variable Target) y un conjunto de variables predictoras (Variables Input).

En el caso del modelo que se está desarrollando en este trabajo se utilizarán bases de alumnos que hayan desaprobado el examen de Matemáticas (Target 1) y clientes que lo hayan aprobado (Target 0). Una vez entrenado el modelo, y que este haya aprendido, se va a poder detectar con anticipación aquellos alumnos con dificultades en Matematica, detectando los perfiles más propensos a desaprobar el examen.

El proceso de Modelado se divide en varias etapas a continuación detalladas:

- Primera etapa: Esta etapa es una de las más importantes en el proceso y consiste en el armado de la base necesaria para entrenar el modelo. Se construyen las variables consideradas necesarias para modelar del periodo de observación. El objetivo es generar la mayor cantidad de variables posibles para que el algoritmo pueda seleccionar las de mayor poder explicativo del fenómeno a predecir.

- Segunda etapa: Consta de la generación de la Variable Target en el periodo de predicción. Para esto se definen los criterios de Aprobación y desaprobación. En el caso de desaprobación la Variable Target va a tomar el valor 1 y en el caso contrario 0.

- Tercer etapa: Una vez construida la base y teniendo a cada alumno con un valor en la Variable Target comienza la fase de entrenamiento donde se comienza con un tratamiento de los datos, una selección de variables para descartar algunas variables con poco poder predictivo y luego, después de probar distintos algoritmos de predicción se selecciona el que mejor se adecua a nuestro objetivo. En este caso se van a probar los algoritmos de Regresión Logística, Redes Bayesianas, Clasificador KNN, Arboles de decisión, y Random Forest. Más adelante se detalla estos pasos para el modelo en análisis.

- Cuarta Etapa: Esta última es la llamada Etapa de Validación. Una vez elegido el modelo "Ganador" se realiza una validación atemporal, lo que significa aplicar el modelo a alumnos diferentes a los utilizados para la etapa de entrenamiento y se analiza la estabilidad del mismo.

Para dar por concluido el proceso estas pruebas tienen que tener como resultado una performance parecida a la de entrenamiento.

3.3 Algoritmos

A continuación, se introducen y explican los distintos algoritmos utilizados en el proyecto, y mencionados anteriormente.

3.3.1 Regresión Logística

La regresión logística es una técnica analítica que nos permite relacionar funcionalmente una variable dicotómica con un conjunto de variables independientes. Puede considerarse una extensión de los modelos de regresión lineal, con la particularidad de que el dominio de salida de la función está acotado al intervalo [0,1] y que el proceso de estimación, en lugar de mínimos cuadrados, utiliza el procedimiento de estimación máximo-verosímil. Es esencial el uso de programas informáticos ya que este método de estimación es iterativo.

En otras palabras, un modelo de regresión logística es aquel en donde se asume que las variables explicativas multiplicadas por sus respectivos coeficientes tiene una relación lineal, no directamente con la variable de respuesta, sino con el logaritmo natural de las probabilidades (ODDS) de que el evento va a ocurrir. Esto se define con la fórmula final de la regresión logística:

$$Ln\left(\frac{p}{1-p}\right) = \alpha + \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n$$

Donde “p” es la probabilidad de que un cliente sea malo y “1-p” es la probabilidad de que un cliente sea bueno. El término (p/(1-p)) es llamado ODDS y se entiende como la proporción del número esperado de veces que un evento ocurra y el número esperado de veces que no ocurra.

Despejando “p” de la fórmula anterior se tiene, que la probabilidad de ser un cliente malo es:

$$p = \frac{1}{1 + e^{-(\alpha+\beta_0+\beta_1 X_1+\cdots+\beta_n X_n)}}$$

Uno de los problemas fundamentales cuando intervienen diversas variables en un fenómeno es determinar cuál es la contribución de cada una de ellas. El modelo de regresión logística a veces es el elegido ya que, a diferencia del resto de los algoritmos que veremos a continuación, es de mayor facilidad para interpretar los efectos que tienen las variables predictoras o independientes sobre la variable dependiente. Esto puede llevarse a cabo con la lectura de los coeficientes de ODDS.

3.2.2 Arboles de Decisión

Una de las técnicas más comunes de minería de datos son los árboles de decisión (TDIDT) utilizados para descubrir conocimiento en formato de regla que constituye un modelo que representa el dominio de conocimiento subyacente a los ejemplos disponibles del mismo.

Un árbol de decisión es una forma gráfica y analítica de representar todos los eventos (sucesos) que pueden surgir a partir de una decisión asumida en cierto momento. Nos ayudan a tomar la decisión más "acertada", desde un punto de vista probabilístico, ante un abanico de posibles decisiones. Estos árboles permiten examinar los resultados y determinar visualmente cómo fluye el modelo. Los resultados visuales ayudan a buscar subgrupos específicos y relaciones que tal vez no encontraríamos con estadísticos más tradicionales. (Silvente, Hurtado, & Baños, 2013)

Los árboles de decisión son una técnica estadística para la segmentación, la estratificación, la predicción, la reducción de datos y el filtrado de variables, la identificación de interacciones, la fusión de categorías y la discretización de variables continuas.

3.2.3 Clasificador K-NN

El algoritmo los k-vecinos más cercanos (kNN) es un algoritmo simple y de alto rendimiento de clasificación supervisada. kNN pertenece al paradigma perezoso de aprendizaje (lazy learning), donde el trabajo se retrasa todo lo posible, ya que no se construye ningún modelo. El modelo son los propios datos o conjunto de entrenamiento, y se trabaja cuando llega un nuevo ejemplo a clasificar. Aunque kNN es una técnica simple, ha demostrado ser uno de los algoritmos más

efectivos en la Minería de Datos (está considerado uno de los 10 algoritmos más importantes de la Minería de Datos (Abdelmalik , Inza, & Larrañaga)

El algoritmo kNN se basa en el cálculo de distancias entre el ejemplo a clasificar y un grupo de k ejemplos del conjunto de entrenamiento. Dichos ejemplos son los que se encuentran más próximos al ejemplo que queremos clasificar, de modo que lo clasificamos según las clases a las que pertenezcan los k ejemplos obtenidos

3.2.4 Redes Bayesianas

Una red bayesiana es un gráfico acíclico dirigido en el que cada nodo representa una variable y cada arco representa una dependencia probabilística que especifica el condicional de cada variable dados sus padres; la variable a la que apunta el arco depende (causa-efecto) de la variable en el origen de este (Felgaer, 2004)

Las redes bayesianas (Pearl, 1988) son utilizadas en diversas áreas de aplicación como por ejemplo medicina (Beinlich, Suermondt, Chavez, & Cooper, 1989), ciencias (Bickmore, 1994) y economía (Ezawa & Schuermann, 1995) .Las mismas proveen una forma compacta de representar el conocimiento y métodos flexibles de razonamiento -basados en las teorías probabilísticas- capaces de predecir el valor de variables no observadas y explicar las observadas. Entre las características que poseen las redes bayesianas se puede destacar que permiten aprender sobre relaciones de dependencia y causalidad, permiten combinar conocimiento con datos (Heckerman, Chickering, & Geiger, 1995) y pueden manejar bases de datos incompletas (Heckerman D. , 1995)

Las redes bayesianas están diseñadas para hallar las relaciones de dependencia e independencia entre todas las variables que conforman el dominio de estudio. Basado en ello, se utilizan métodos de razonamiento probabilístico que permiten realizar predicciones sobre el valor de cualquier variable desconocida basados en los valores de las conocidas.

Existen muchas tareas prácticas que pueden reducirse a problemas de clasificación: diagnóstico médico y reconocimiento de patrones son sólo dos ejemplos de ellas.

Las redes bayesianas pueden realizar la tarea de clasificación -caso particular de predicción- que se caracteriza por tener una sola de las variables de la base de datos (clasificador) que se desea

predecir, mientras que todas las otras son los datos propios del caso que se desea clasificar. Pueden existir una gran cantidad de variables en la base de datos, algunas de las cuales estén directamente relacionadas con la variable clasificadora pero también otras variables que tienen una influencia directa sobre dicha clase (Felgaer, 2004)

3.2.5 Random Forest

El Random Forest es uno de los más poderosos algoritmos de aprendizaje de máquina disponibles en la actualidad. Es un algoritmo de aprendizaje automático supervisado que se puede utilizar para tareas de clasificación (predice una salida de valor discreto, es decir, una clase) y regresión (predice una salida de valor continuo). (Pramoditha)

Los dos conceptos principales detrás de los bosques aleatorios son:

- La sabiduría de la multitud: un gran grupo de personas es colectivamente más inteligente que los expertos individuales
- La diversificación - un conjunto de incorrelados árboles

Cuando entrena un bosque aleatorio para una tarea de clasificación, en realidad entrena un grupo de árboles de decisión. Luego, obtiene las predicciones de todos los árboles individuales y predice la clase que obtiene la mayor cantidad de votos. Aunque algunos árboles individuales producen predicciones incorrectas, muchos pueden producir predicciones precisas. Como grupo, pueden avanzar hacia predicciones precisas. A esto se le llama la sabiduría de la multitud.

En la siguiente figura, se puede ver gráficamente el funcionamiento de un Random Forest

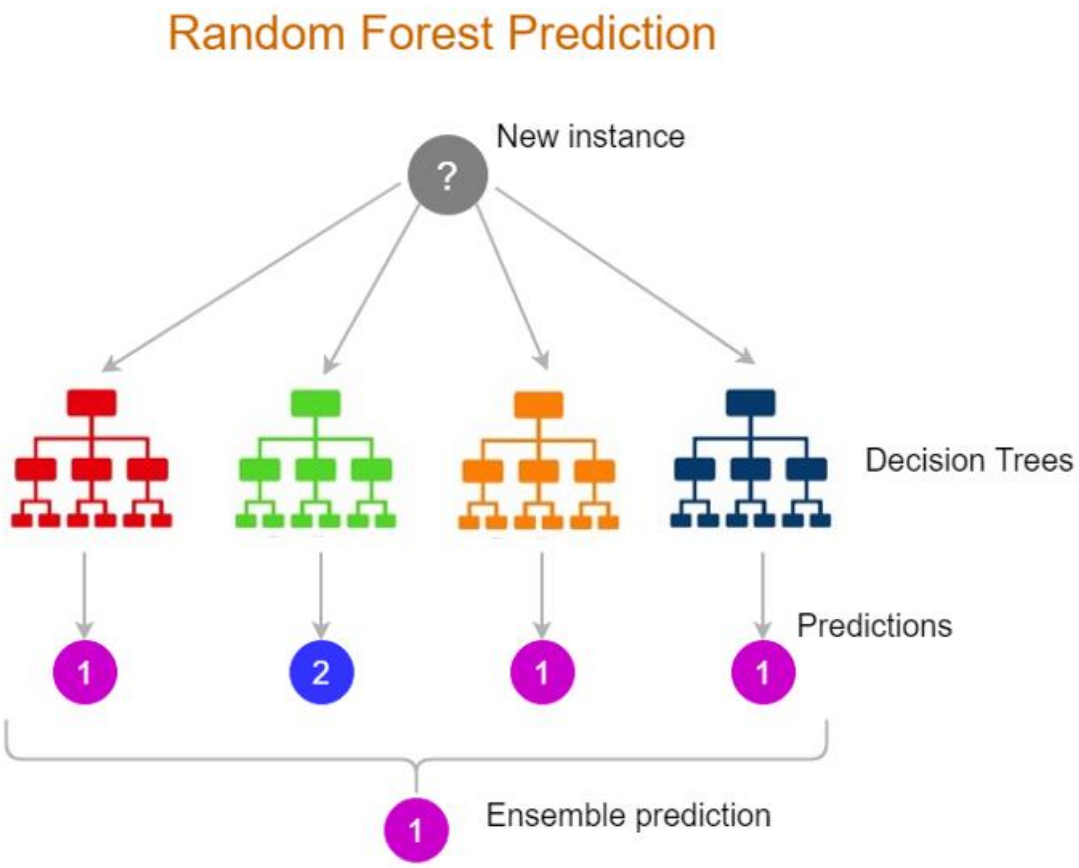

Figura 4 Diagrama del funcionamiento Random Forest (Pramoditha)

Para mantener una baja correlación (alta diversificación) entre árboles individuales, el algoritmo considera automáticamente las siguientes cosas.

- Aleatoriedad de características
- Embolsado (agregación bootstrap)

En un árbol de decisión normal, el algoritmo busca la mejor característica de todas las características cuando quiere dividir un nodo. Por el contrario, cada árbol en un bosque aleatorio busca la mejor característica de un subconjunto aleatorio de características. Esto crea una aleatoriedad adicional al hacer crecer los árboles dentro de un bosque aleatorio. Debido a la aleatoriedad de características, los árboles de decisión en un bosque aleatorio no están correlacionados.

3.2.6 Light Gradient Boosting Machine

LightGBM es un algoritmo de refuerzo (o también de potenciación) de gradientes (gradient boosting) basado en modelos de árboles de decisión, desarrollado por Microsoft. Puede ser utilizado para la categorización, clasificación y muchas otras tareas de aprendizaje automático, en las que es necesario maximizar o minimizar una función objetivo mediante la técnica de gradient boosting, que consiste en combinar clasificadores sencillos, como por ejemplo árboles de decisión de profundidad limitada (Brownlee, 2020)

Entre sus principales ventajas podemos destacar las siguientes:

- Mayor velocidad de entrenamiento y mayor eficiencia
- Menor uso de memoria
- Mayor precisión
- Soporte de aprendizaje paralelo y soporte para GPUs
- Capacidad para manejar datos a gran escala

Los experimentos de comparación en conjuntos de datos públicos muestran que LightGBM puede superar los marcos de impulso existentes tanto en eficiencia como en precisión, con un consumo de memoria significativamente menor. Además, los experimentos de aprendizaje distribuido muestran que LightGBM puede lograr una aceleración lineal mediante el uso de varias máquinas para entrenar en entornos específicos.

3.2.7 Gradient Boosting

Gradient boosting o Potenciación del gradiente, es una técnica de aprendizaje automático utilizado para el análisis de la regresión y para problemas de clasificación estadística, el cual produce un modelo predictivo en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Construye el modelo de forma escalonada como lo hacen otros métodos de boosting, y los generaliza permitiendo la optimización arbitraria de una función de pérdida diferenciable.

La idea de la potenciación del gradiente fue originada en la observación realizada por Leo Breiman en donde el Boosting puede ser interpretado como un algoritmo de optimización en una función de coste adecuada. Posteriormente Jerome H. Friedman desarrolló algoritmos de aumento de gradiente de regresión explícita, simultáneamente con la perspectiva más general de potenciación del gradiente funcional de Llew Mason, Jonathan Baxter, Peter Bartlett y Marcus Frean. En sus últimos dos trabajos presentaron la visión abstracta de los algoritmos de potenciación como algoritmos iterativos de descenso de gradientes funcionales. Es decir, algoritmos que optimizan una función de coste sobre el espacio de función mediante la elección iterativa de una función (hipótesis débil) que apunta en la dirección del gradiente negativo. Esta visión de gradiente funcional de potenciación ha llevado al desarrollo de algoritmos de potenciación en muchas áreas del aprendizaje automático y estadísticas más allá de la regresión y la clasificación.

3.3 Medidas de Ajuste de los Modelos

Frente a la multiplicidad de métodos de modelado, cada uno con sus propios indicadores estadísticos de calidad, se han tratado de encontrar métricas universales para el desempeño.

A continuación describiremos los criterios mejor desarrollados y más utilizados para medir la efectividad de los modelos y así poder compararlos entre sí.

3.3.1 Curva ROC

J. Fogarty, R. Baker, S. Hudson (2005), dicen que para entender este concepto primero es necesario introducir algunos conceptos.

- Valor de corte (π_0) es aquel valor que convierte a las predicciones en 1 cuando la probabilidad del evento estimada lo supera, y en 0 cuando la probabilidad del evento estimada es menor. El valor habitual que se suele tomar de corte es $\pi_0=0.5$

- Sensibilidad y Especificidad, la predicción de éxito cuándo es cierta se denomina sensibilidad, y la predicción de fracaso cuando es, a su vez cierta, se denomina especificidad.

Una Curva ROC (Receiver Operating Characteristic) es un gráfico en el que se representa la sensibilidad en función de (1-especificidad). Si vamos modificando los valores de corte π_0 y representamos la sensibilidad (en ordenadas) en función de (1-especificidad) (en abscisas) obtendremos la curva ROC. Ésta es una función cóncava que conecta los puntos (0,0) y (1,1). Cuando π_0 es cercano a 0, casi todas las predicciones serán 1, con lo cual la sensibilidad estará próxima a 1 y la especificidad estará cercana a 0. Así, el punto (1-especificidad, sensibilidad) tendrá coordenadas (1,1). Cuando π_0 es cercano a 1, casi todas las predicciones serán 0, con lo cual la sensibilidad estará próxima a 0 y la especificidad estará cerca de 1. Así, el punto (1-especificidad, sensibilidad) tendrá coordenadas (0,0). Cuanto mayor sea el área bajo la curva (AUC), mejores serán las predicciones. Un área igual a 0.5 representa el azar, mientras que un área igual a 1 representa al mejor modelo.

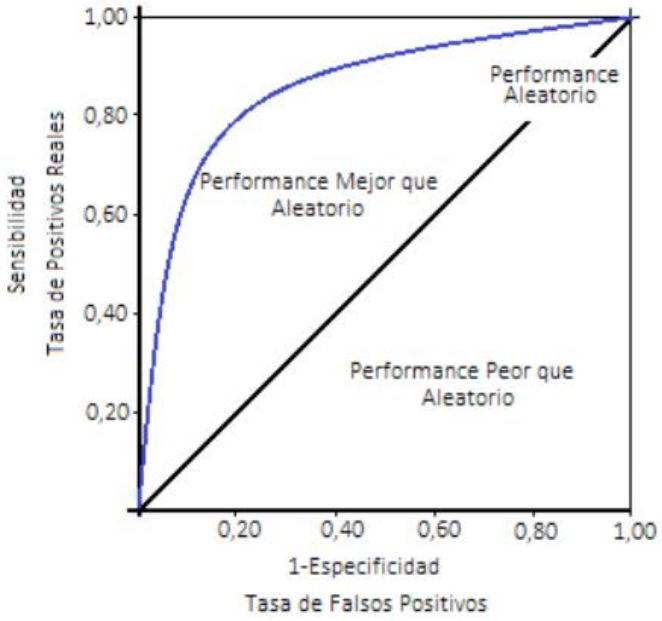

Figura 5 Curva ROC (Fogarty, Baker, & Hudson, 2005)

La curva ROC nos permite comparar modelos de diferentes tipos.

3.3.2 Kolmogorov-Smirnov (KS)

La otra medida que vamos a tener en cuenta al momento de hacer nuestra comparación de modelos para así poder quedarnos con el de mejor performance es la prueba de Kolmogorov-Smirnov (KS), y a su vez es una de las medidas más importantes para el monitoreo de los modelos, o sea, para analizar si el modelo mantiene su performance en el tiempo.

El estadístico KS puede ser usado para medir la capacidad de clasificación de un modelo, ya consiste en medir cuan distintas son las funciones de distribución de buenos y malos clientes para cada valor de puntaje de score.

Se considera que un modelo con KS menor a 20% debe ser cuestionado y mayor a 70% sea, probablemente, muy bueno para ser cierto (Kisbye, 2010)

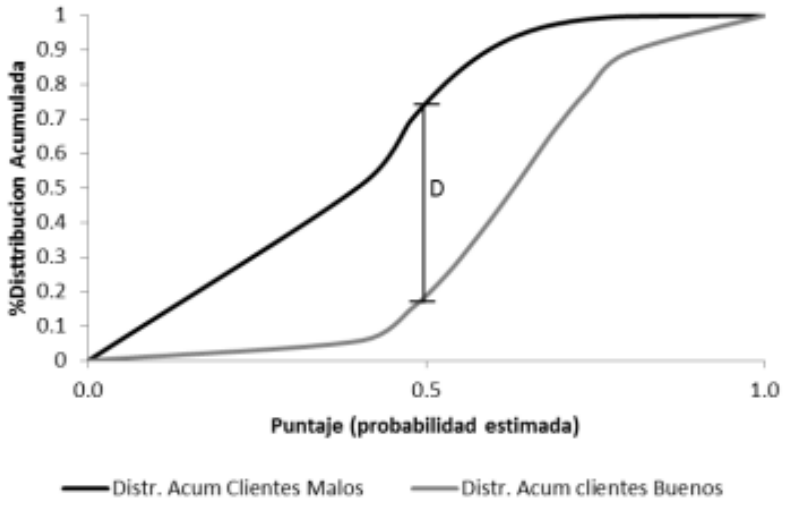

Figura 6 Test de Kolmogorov-Smirnov

4. Modelo Predictivo

En este capítulo se desarrolla el trabajo, siguiendo las fases propuestas por la metodología CRISP-DM

4.1 Comprensión del negocio

En esta fase de la metodología, se dará respuesta a cada una de las cuestiones planteadas a continuación.

4.1.1 Determinación de los objetivos de negocio

El acompañamiento pedagógico a lo largo del año escolar es algo que viene incrementando año tras año, luego de comprobar que esto genera mejores resultados y experiencias en los alumnos y los ayuda en su crecimiento personal. Hoy en día, gracias a toda la información disponible, es posible trazar planes de estudio y seguimientos personalizados para cada alumno.

Utilizando el resultado del examen de Matemática para los alumnos de secundario, en el proyecto se tratará de encontrar patrones de comportamiento en la población que tiene notas regulares o debajo del promedio, y también de alumnos con notas sobresalientes.

Esto conlleva a preguntas como ¿Importa el acceso a internet para la performance del alumno? ¿El resultado del examen varía según los ingresos económicos del hogar? ¿Importa el nivel de educación de los padres para que el alumno incorpore correctamente los conceptos de Matemática?

El criterio del éxito del trabajo, se basa en encontrar que atributos son los más ponderantes en las notas de los alumnos, y generar un modelo de predicción para detectar a principio de año aquellos alumnos que tendrán mayores dificultades en matemática, y aquellos que destacarán.

4.1.2 Evaluación de la situación

Para llevar a cabo los objetivos del negocio, se procesará la información en una computadora doméstica, con un procesador AMD Ryzen 5, y 8 Gb de memoria Ram.

Las fuentes de información que se utilizarán para el proyecto, serán la encuesta Aprender 2019 y el diccionario de datos de la encuesta mencionada Se cuenta con información referente a cada encuesta mediante los sitios oficiales del ministerio de educación, así como también expertos en el área a disposición de la población.

En cuanto a herramientas de procesamiento de datos, se utiliza el lenguaje Python, a través de Jupyter Lab para la preparación, modelado y graficado de la información, y Excel para la representación gráfica de algunos atributos.

En cuanto a presupuesto, se dispone de los recursos disponibles por el alumno y la universidad.

Como beneficio de este proyecto, se intenta dar una herramienta para ayudar a detectar el desempeño de los alumnos en matemática de manera temprana.

4.1.3 Determinación de los objetivos de la minería de datos

El objetivo de la minería en el proyecto será generar un algoritmo de predicción para determinar los alumnos con más riesgo de tener notas bajas en matemática. Para esto, se utilizar algoritmos de clasificación, como regresión logística, Naive Bayes, KNN, Arboles de Decisión y Random Forest. Así mismo, se dará un ranking de variables más ponderantes, y un análisis descriptivo de cada una

Se evaluarán distintos modelos en base a pruebas estadísticas como R2, Accuracy, entre otros. Para cada algoritmo se probará con distintos hiperparametros, mediante la utilización de GridSearch. El objetivo final es obtener un modelo que tenga un Accuracy de al menos 0,7, y un área bajo la curva ROC aceptable.

4.1.4 Producir el plan del proyecto

En el proyecto se contemplan las siguientes etapas generales:

I. Recolección de datos desde el Ministerio de Educación: Se descargan desde el sitio web las encuestas de Aprender 2019, el diccionario de datos y el detalle de dicha instancia.
II. Exploración y verificación de calidad de los datos: Se verifica los datos mediante Jupyter Lab, utilizando las librerías de Numpy, Pandas y Matplotlib
III. Preparación de los datos para el análisis: Se imputan los datos que no se encuentren completos, se generar variables discretas a partir de las variables de texto, y por último se generan variables Dummies a partir de estas. Por otro lado, se normalizan las variables

numéricas, y se genera la variable binaria target. También se divide el dataset en dos partes, una para entrenamiento y otra para evaluación.

IV. Modelado: En esta etapa se utilizan las técnicas de minería de datos mencionadas en la etapa de objetivo de minería de datos, mediante la librería Sklearn en Python

V. Evaluación de los resultados: Se evalúan los resultados de los modelos realizados, mediante análisis de R2, Acuracy, Matriz de confusión, ROC, KS, etc y se valida la efectividad y precisión mediante la aplicación del modelo a un set de prueba.

4.2 Comprensión de los datos

La fase de comprensión de los datos incluye la recolección, descripción, exploración y verificación de la calidad de los mismos

4.2.1 Recolección de datos iniciales

Los datos utilizados en este proyecto son datos referentes a la encuesta Aprender:

> Es una encuesta de propósitos múltiples que releva información sobre los alumnos en torno a distintos aspectos: situación familiar, características demográficas básicas (edad, sexo, situación conyugal, etc.), características migratorias, habitacionales, educacionales e ingresos. Más allá de su gran amplitud temática, los aspectos educativos y familiares adquieren una relevancia central. Entre los conceptos principales que permiten dar cuenta de la relación familiar con el desempeño del alumno se encuentra el de condición del hogar, estudios del padre, madre, hermanos, y datos demográficos del barrio

A su vez, se utiliza la nota del examen de Matemática, el cual incluye distintos aspectos sobre Estadística, geometría, y cálculos básicos.

Los atributos específicos que serán útiles a la hora de hacer la minería de datos son:

- Identificación del individuo
- Situación Familiar
- Acceso a internet

- Nivel educativo del entorno
- Ingreso mensual del hogar
- Sexo
- Edad
- Región
- Tiempo dedicado al estudio
- Actividades extracurriculares

Para mayor detalle de las encuestas, pueden acceder a una réplica de las mismas mediante el anexo.

En cuanto a transformación de datos, se ha tenido que pasar todos los valores a discretos, y luego a variables dummies. Por último, se normalizaron los mismos para poder ingresar de manera correcta a los modelos de predicción.

4.2.2 Descripción de los datos

Los datos se encuentran almacenados en un archivo .CSV, con delimitadores ";". En la figura 7 se puede ver una vista previa del dataset.

	ID1	cod_provincia	sector	ambito	claveseccion	idalumno	ap01_01	ap01_02	ap02	ap03	...
0	120003000120003	2	2	1	02SF00003	3	10	4	1	1	...
1	120003000120003	2	2	1	02SF00003	25	7	4	2	1	...
2	120003000120003	2	2	1	02SF00002	27	3	5	1	1	...
3	120003000120003	2	2	1	02SF00002	3	11	4	2	1	...
4	120003000120003	2	2	1	02SF00003	5	11	4	2	1	...

Figura 7. Vista previa del dataset Estudiantes Secundaria Aprender 2019

Todos los atributos se encuentran en formato float, a excepción de los atributos Clave. Los valores nulos son representados, según el atributo, con valores negativos o 0.

En el anexo se deja el detalle al diccionario de variables, con la descripción por atributo y los posibles valores de las variables.

4.2.3 Exploración de datos.

Una vez que se han descrito los datos, se procede a explorarlos, esto implica aplicar pruebas estadísticas básicas que revelarán propiedades de los datos, y crear tablas de frecuencia y gráficos de distribución de los datos. Este informe sirve principalmente para determinar la consistencia y completitud de los datos.

En la figura 8, se muestra la distribución del desempeño de los alumnos en el examen de matemática, separado por cuartiles:

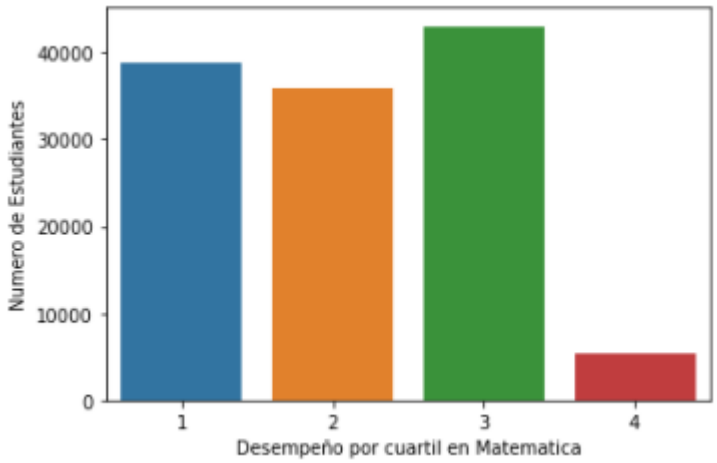

Figura 8 Desempeño por cuartil de alumnos en Matemática

Referencias para la tabla:

Valor	Descripción
1	Por debajo del nivel básico
2	Básico
3	Satisfactorio
4	Avanzado

En la figura 9, se puede ver la distribución de los estudiantes encuestados según el Sexo.

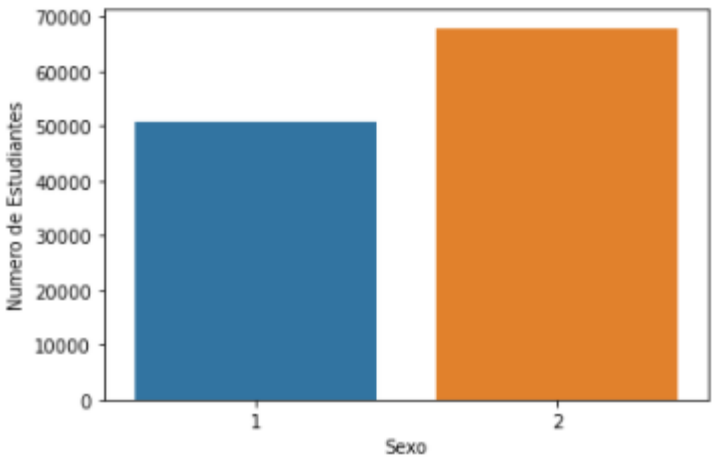

Figura 9. Distribución de Estudiantes según Sexo

Referencias para la tabla:

Valor	Descripción
1	Varón
2	Mujer

En la figura 10, se muestran las distribuciones según el Sector del colegio.

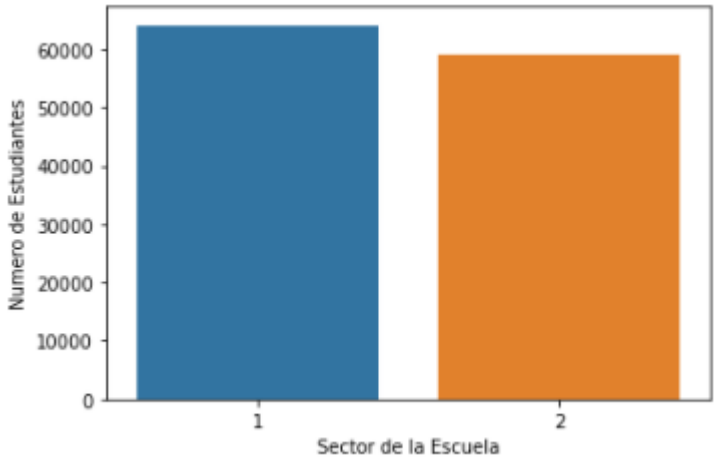

Figura 10. Distribución según el sector del Colegio

Referencias para la tabla:

Valor	Descripción
1	Estatal
2	Privado

En la figura 11, podemos ver la distribución según el ámbito del Colegio

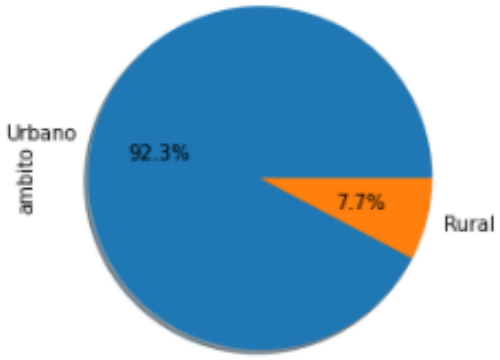

Figura 11. Distribución según Ámbito Escolar

También podemos ver la distribución según el índice Socioeconómico (basado en el cálculo de la encuesta permanente de hogares realizada por el INDEC) en la figura 12. La misma contempla las siguientes variables:

- Nivel educativo de los padres.
- Hacinamiento en el hogar (relación entre la cantidad de habitaciones de la vivienda en la que habita el estudiante y el número de miembros del hogar).
- Recepción de la Asignación Universal por Hijo (AUH) en el hogar.
- Tenencia de equipamiento informático en el hogar (Internet, consolas de videojuegos, televisión y celular).

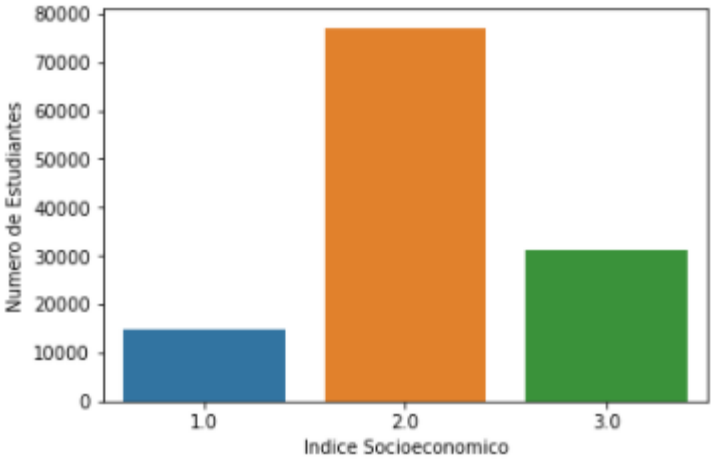

Figura 12 Distribución según índice Socioeconómico

4.2.4 Verificación de la calidad de los datos.

Luego de hacer la exploración inicial de los datos se puede afirmar que son completos, y aptos para el análisis. La cantidad de registros cubre los necesarios para aplicar los modelos. Existen datos nulos y outliers que se tendrán que trabajar en la preparación de los datos. Una posible solución para aplicar la minería de datos en los datos nulos o sin contestar, será descartarlos de la muestra, o imputarlos con la media o moda del atributo. El dataset tiene en total 343.750 registros, y 264 atributos.

4.3 Preparación de los datos

En esta fase, se trata de preparar los datos para aplicar las técnicas de minería de datos sobre ellos de manera eficaz y eficiente. A su vez se debe adaptar los datos a la necesidad, seleccionando un subset de datos con los que se trabajará posteriormente.

4.3.1 Seleccionar los Datos

En principio, se van a utilizar todos los registros de la encuesta Aprender 2019. En términos de atributos, se seleccionarán solo los que tengan mayor correlación con el atributo target (aprox. 65 atributos). Los mismos son listados a continuación:

cod_provincia	ap25_03	ap50_05
sector	ap28_01	ap51_01
ambito	ap29_01	ap51_02
ap02	ap34	ap51_03
ap03	ap35_03	ap51_04
ap04	ap39_01	ap51_05
ap05	ap39_02	ap51_06
ap06	ap39_03	ap51_07
ap07	ap39_04	ap51_08
ap09	ap40_01	ap51_09
ap11_08	ap40_02	ap51_10
ap14_02	ap40_03	ldesemp
ap16	ap40_04	mdesemp
ap17	ap40_05	isocioa
ap23_01	ap40_06	ap26_rec
ap23_02	ap42_01	trabaja_fuera_hogar
ap23_03	ap43_a	trabaja_fuera_hogar_remunerado
ap23_04	ap49_01	migración
ap23_05	ap49_02	edadA_junio2019
ap23_06	ap50_01	ap32_e
ap24	ap50_02	ap33_01_a
ap25_01	ap50_03	ap33_01_b
ap25_02	ap50_04	

Para el dataset final, por limitaciones de procesamiento, se seleccionará una muestra aleatoria de 100.000 registros para entrenar y validar los distintos modelos de predicción

4.3.2 Limpiar los Datos

La base de datos con la que se cuenta para el proyecto contiene toda la información necesaria para poder cumplir los objetivos de la minería de datos.

La tabla cuenta con valores negativos, que, según el atributo, representa falta de contestación, o una opción.

Para poder separar estos casos del resto, primero transformamos los atributos en los cuales el valor negativo nos aporta una respuesta valida:

Para esto, se remplaza en los atributos ap32_e, ap33_01_a, ap33_01_b y ap43_a el valor “-9” por “2”.

Luego, se pasan el resto de valores negativos en el dataset a NaN (Nulos en Python), y se pasan todas las columnas a formato Integer (Entero).

Por último, vemos el % de valores faltantes por atributo:

Atributo	*% Valores Faltantes*
cod_provincia	0%
sector	0%
ambito	0%
ap02	3%
ap03	2%
ap04	2%
ap05	4%
ap06	3%
ap07	4%
ap09	3%
ap11_08	6%
ap14_02	15%
ap16	3%
ap17	4%
ap23_01	4%
ap23_02	7%
ap23_03	7%
ap23_04	6%
ap23_05	8%
ap23_06	7%
ap24	3%
ap25_01	9%
ap25_02	9%
ap25_03	10%
ap28_01	4%
ap29_01	4%
ap32_e	1%
ap33_01_a	1%
ap33_01_b	1%
ap34	6%
ap35_03	6%
ap39_01	6%
ap39_02	6%
ap39_03	6%

ap39_04	6%
ap40_01	6%
ap40_02	7%
ap40_03	7%
ap40_04	7%
ap40_05	7%
ap40_06	7%
ap42_01	9%
ap43_a	1%
ap49_01	27%
ap49_02	27%
ap50_01	25%
ap50_02	26%
ap50_03	27%
ap50_04	27%
ap50_05	26%
ap51_01	26%
ap51_02	26%
ap51_03	27%
ap51_04	27%
ap51_05	27%
ap51_06	27%
ap51_07	28%
ap51_08	27%
ap51_09	27%
ap51_10	27%
ldesemp	4%
mdesemp	6%
isocioa	6%
ap26_rec	3%
trabaja_fuera_hogar	4%
trabaja_fuera_hogar_remunerado	4%
migración	2%
edadA_junio2019	5%

Dado que tenemos suficientes datos para analizar, borraremos todos los registros donde haya uno o más atributos nulos.

Nuestro dataset final queda de 123.018 registros, y 68 columnas.

4.3.3 Construir los Datos

La variable target (mdesemp) tiene de origen 4 posibles valores (1 - Por debajo del nivel básico, 2 - Básico, 3 – Satisfactorio, 4 – Avanzado). Para el modelo predictivo, vamos a transformar la variable en binaria, agrupando los valores de la siguiente manera:

Valor Final	Valor Origen
1	1- Por debajo del nivel básico
0	2- Básico
	3- Satisfactorio
	4- Avanzado

4.3.4 Integrar los Datos

No ha sido necesario generar nuevas estructuras ni la fusión de distintas tablas, dado que, al ser una única tabla de origen, todos los datos están integrados previamente.

4.3.5 Formateo de los Datos

Para poder adaptar el dataset a los modelos predictivos, fue necesario generar variables Dummies con todos los atributos discretos del dataset. Esto se realizó con la función get_dummies de Pandas. Para ahorrar recursos, se eliminó la primera columna de cada dummie generada.

Dado que el dataset final queda de 123.000 registros, y nuestra unidad de procesamiento es una computadora hogareña, tomamos una muestra aleatoria estratificada de 20.000 registros para ingresar en los modelos de predicción. De esta manera podremos avanzar con el proyecto más allá de las limitaciones de Hardware.

Por otro lado, se normalizaron los atributos continuos, para mejorar las predicciones de los algoritmos. Para esto se utilizó la función StandardScaler, de Sklearn.

Una vez extraída dicha muestra, evaluamos la distribución de nuestra variable target:

Valor de atributo Target	Cantidad
0	13.663

1	6.337

Como vemos en la tabla, la distribución no es balanceada ya que los registros con target = 1, son un 46% menos que los targets= 0.

Para mejorar la predicción de los modelos, es recomendable balancear las clases en la etapa de entrenamiento del modelo.

Siguiendo esto, lo primero que se realizó con el dataset es separarlo en Entrenamiento y validación, mediante la herramienta Model_Selection, de SckitLearn. La función se encarga de separar de manera equitativa el dataset en 2 subdatasets. Para el entrenamiento dejamos un 70% del dataset, y un 30% para la validación.

Luego, aplicamos la función de under_sampling, del módulo imblearn para balancear las clases del dataset de Entrenamiento. La misma implica la selección aleatoria de ejemplos de la clase mayoritaria para eliminarlos del conjunto de datos de entrenamiento.

Finalmente, separamos tanto el dataset de entrenamiento, como de validación, en 2 distintos para separar el atributo target de los features, quedando de la siguiente manera:

X_train = Dataset de entrenamiento con atributo Target binario.

Y_train = Dataset de entrenamiento con features formateadas.

X_test = Dataset de validación con atributo Target binario.

Y_test = Dataset de validación con features formateadas.

4.4 Modelado

El modelado consiste en la implementación de los algoritmos de minería de datos necesarios para responder las preguntas surgidas en las fases de conocimiento del negocio y de exploración de los datos.

4.4.1 Escoger la Técnica de Modelado

Dado que nuestro problema es de clasificación binaria, podemos usar múltiples algoritmos para dicha finalidad, con múltiples hiperparametros en cada uno.

Para poder realizar esta tarea, vamos a utilizar la función Pipeline del módulo Scikitlearn. Esta consiste en ensamblar distintos algoritmos, y validar cada modelo con cross validation.

Por otro lado, vamos a utilizar GridSearch, que es otra herramienta del módulo ScikitLearn, que nos permite probar distintas combinaciones de hiperparametros para cada modelo, logrando así obtener el modelo más preciso para nuestra predicción.

Finalmente se podrán medir con estadísticos cada modelo realizado, y pipelines nos mostrará cual es el mejor, y con que hiperparametros tuvo su mejor performance.

Dentro de la función Pipelines, vamos a desarrollar pruebas con los siguientes algoritmos:

- Regresiones Logísticas
- Naive Bayes
- Clasificador KNN
- Arboles de Decision
- Random Forest
- LightGBM
- XGBoosting

En total, con los distintos algoritmos y su diferente combinación de hiperparametros, estimamos probar 300 modelos distintos para ver cual tiene mejor nivel de predicción.

4.4.2 Generar el plan de Prueba

El procedimiento que se empleará para probar la calidad y validez del modelo será, por un lado, la separación del dataset inicial en entrenamiento y validación (70% entrenamiento, 30% validación). Por otro lado, se utilizará la técnica de CrossValidation en la etapa de entrenamiento de todos los modelos, con separación de 5 sub datasets.

Por otro lado, vamos a utilizar distintos métodos estadísticos para evaluar la calidad y performance de los modelos. Entre los principales métodos de evaluación, se encuentran:

- Accuracy
- Precision
- Recall
- Score F1
- AUC (Area bajo la curva ROC)

Se agregan los siguientes criterios para las tareas de regresión:

- Root Mean Squared Error
- Mean Squared Error

4.4.3 Construir el Modelo

A continuación, se procederá a ejecutar el modelo elegido sobre los datos de entrenamiento. En este apartado se describirán los ajustes de parámetros del modelo que se eligen en la herramienta de minería de datos, así como la salida de dichos modelos y su descripción.

Separaremos esta sección según el algoritmo probado.

- Regresiones Logísticas:

  Para los modelos de regresión, se aplicaron pruebas con los siguientes posibles hiperparametros:

  - Penalidad: 11 o 12
  - Regularizacion C: 0.001, 0.01, 0.1, 1, 10, 100, 1000

- Naive Bayes:

  Para Bayes, se probó con distintos valores sobre la porción de mayor variación de todas las características (smoothing), utilizando valores entre 0.00000001, 0.000000001 y 0.00000001

- Clasificador KNN:

Para KNN, se probó con distinto rango de distancia entre los vecinos. Probamos con todos los valores enteros entre 1 y 50.

- Arboles de Decisión:

  Para arboles de decisión, se probó con toda la combinación de los siguientes hiperparametros:

  - Criterio de clasificación: Gini o Entropía
  - Criterio de división: Mejor o Aleatorio
  - Máxima profundidad: None, 5 o 10
  - Mínimo de muestras para separación: 2 o 5
  - Mínimo de muestras para hoja: 1, 2 o 3

- Random Forest:

  Para este algoritmo, se utilizaron la siguiente combinación de hiperparametros:

  - Criterio de clasificación: Gini o Entropía
  - Cantidad de estimadores: 3, 5, 10 o 50
  - Máxima profundidad: None, 5 o 10
  - Mínimo de muestras para separación: 2 o 5
  - Peso de clases: None, o Balanceadas

- LightGBM & XGBoosting:

  Para el ensamble de árboles de decisión, se utilizaron la siguiente combinación de hiperparametros:

  - Detención temprana de rondas: 20
  - Métrica de evaluación: AUC
  - Evaluación de nombres: Valido
  - Verbose: 100
  - Máxima profundidad: Aleatorio entre 10 y 50
  - Numero de abandonos: Aleatorio entre 6 y 50
  - Ratio de aprendizaje: [0.1,0.01,0.001]
  - Mínimo de hijos por muestra: Aleatorio entre 100 y 500
  - Peso mínimo de hijo: [1e-5, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3, 1e4]
  - Registro Alpha: [0, 1e-1, 1, 2, 5, 7, 10, 50, 100]
  - Registro Lambda: [0, 1e-1, 1, 5, 10, 20, 50, 100]

4.4.4 Evaluar el Modelo

En esta sección, se evalúan los resultados de todos los modelos previamente mencionados.

Separaremos esta sección según el algoritmo probado.

- Regresiones Logísticas:

  El mejor modelo surge con los hiperparameros de C: 0.001 y Penalidad: 12. En los datos de evaluación nos devolvió un accuracy de 0.716, y un área bajo la curva ROC de 0.79

Clase	precisión	recall	f1-score
0	0,855	0,724	0,784
1	0,550	0,733	0,629
Promedio	0,702	0,728	0,706

Accuracy:	**0,716**

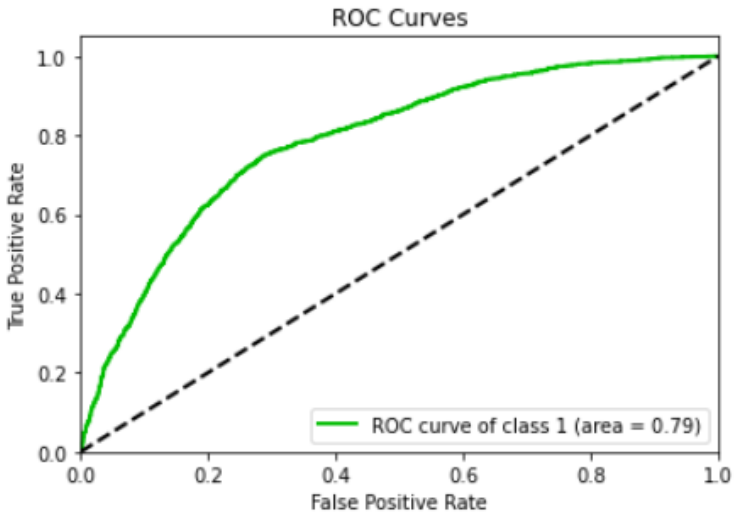

Figura 13 Curva ROC – Modelo Regresión Logistica

- Naive Bayes:

  El mejor modelo surge con Smoothing = 0.00000001. En los datos de evaluación nos devolvió un accuracy de 0.697, y un área bajo la curva ROC de 0.74

Clase	precisión	recall	f1-score
0	0,713	0,933	0,809
1	0,561	0,186	0,279
Promedio	0,637	0,559	0,544

Accuracy:	**0,697**

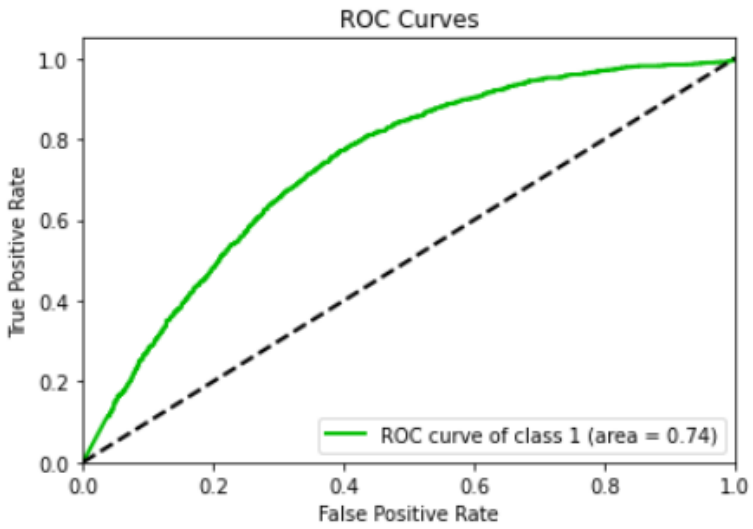

Figura 14 Curva ROC – Modelo Naive Bayes

- Clasificador KNN:

El mejor modelo surge con n_neighbors = 21. En los datos de evaluación nos devolvió un accuracy de 0.706, y un área bajo la curva ROC de 0.73

Clase	precisión	recall	f1-score
0	0,773	0,807	0,790
1	0,538	0,486	0,511
Promedio	0,656	0,647	0,650

Accuracy:	**0,706**

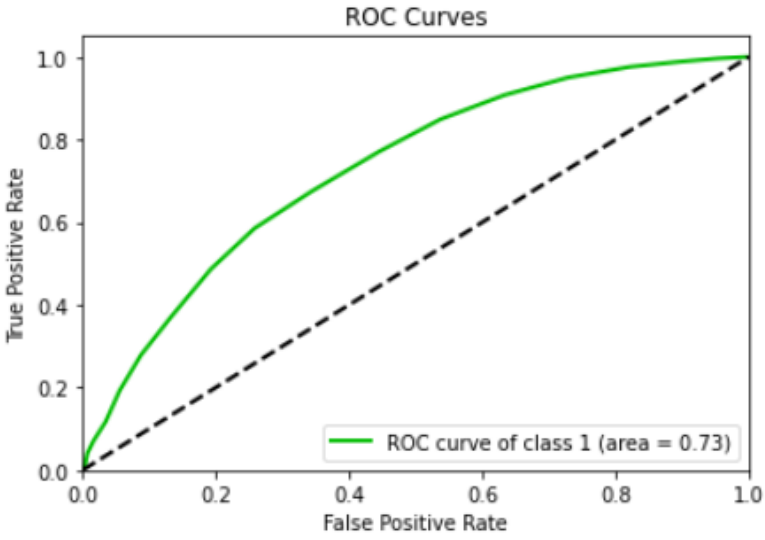

Figura 15 Curva ROC – Modelo clasificador KNN

- Árbol de decisión:

  El árbol con mejor predicción surge de combinar los siguientes hiperparametros:

  - Criterion: 'entropy'
  - Max_depth: 5
  - Min_samples_leaf: 1
  - Min_samples_split: 2
  - Splitter: 'best'

  En los datos de evaluación nos devolvió un accuracy de 0.642, y un área bajo la curva ROC de 0.72

Clase	precisión	recall	f1-score
0	0,841	0,588	0,692
1	0,459	0,758	0,572
Promedio	0,650	0,673	0,632

Accuracy:	**0,642**

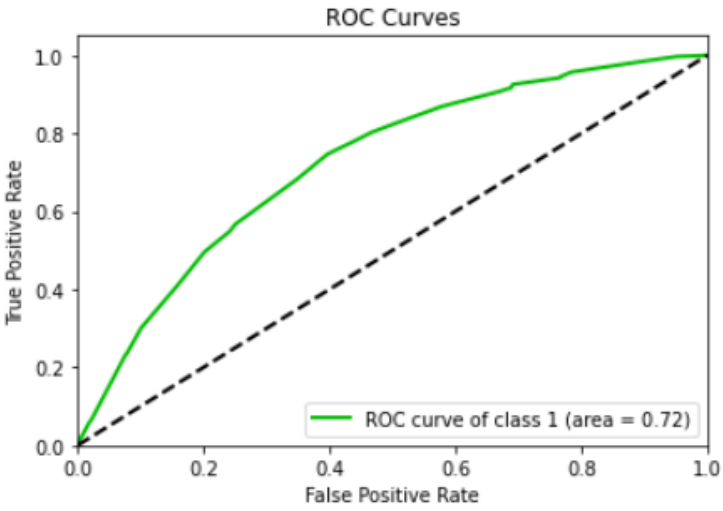

Figura 16 Curva ROC – Modelo de árbol de decisión

- Random Forest:

  El modelo con mejor predicción surge de combinar los siguientes hiperparametros:

  - Criterion: 'entropy'
  - Max_depth: None
  - Class_Weight: ‘balanced’
  - Min_samples_split: 5
  - N_estimators: 50

  En los datos de evaluación nos devolvió un accuracy de 0.69, y un área bajo la curva ROC de 0.76

Clase	precisión	recall	f1-score
0	0,838	0,678	0,750
1	0,506	0,717	0,593
Promedio	0,672	0,697	0,671

Accuracy:	**0,690**

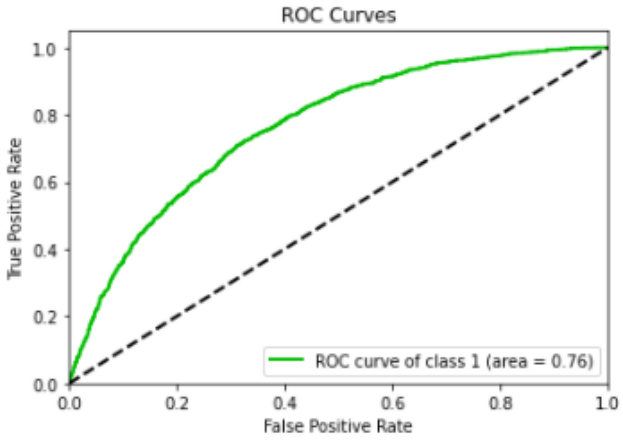

Figura 17 Curva ROC – Modelo Random Forest

- Modelo LightGBM:

  La mejor predicción surge con los hiperparametros :

  - colsample_bytree: 0.47
  - learning_rate :0.01
  - max_depth: 31
  - min_child_samples: 141
  - min_child_weight: 1
  - num_leaves: 16
  - reg_alpha: 1
  - reg_lambda: 0.1

  En los datos de evaluación nos devolvió un accuracy de 0.718, y un área bajo la curva ROC de 0.80

Clase	precisión	recall	f1-score
0	0,851	0,713	0,776
1	0,539	0,729	0,620
Promedio	0,695	0,721	0,698

Accuracy:	**0,718**

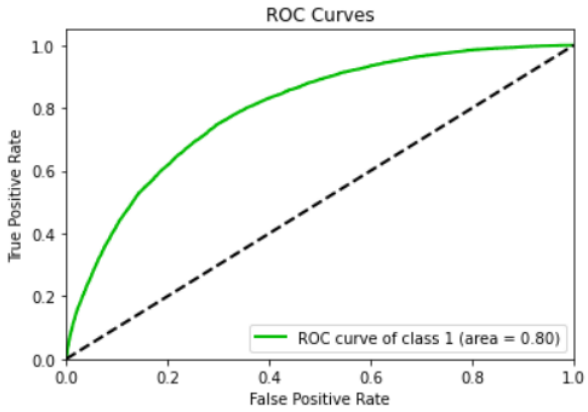

Figura 18 Curva ROC – Modelo LightGBM

- Modelo XGBoosting:

  La mejor predicción surge con los hiperparametros :

  - colsample_bytree: 0.40
  - learning_rate :0.01
  - max_depth: 22
  - min_child_samples: 430
  - min_child_weight: 0.01
  - num_leaves: 22
  - reg_alpha: 0
  - reg_lambda: 100

  En los datos de evaluación nos devolvió un accuracy de 0.707, y un área bajo la curva ROC de 0.79

Clase	precisión	recall	f1-score
0	0,852	0,692	0,764
1	0,525	0,739	0,614
Promedio	0,689	0,716	0,689

Accuracy:	**0,707**

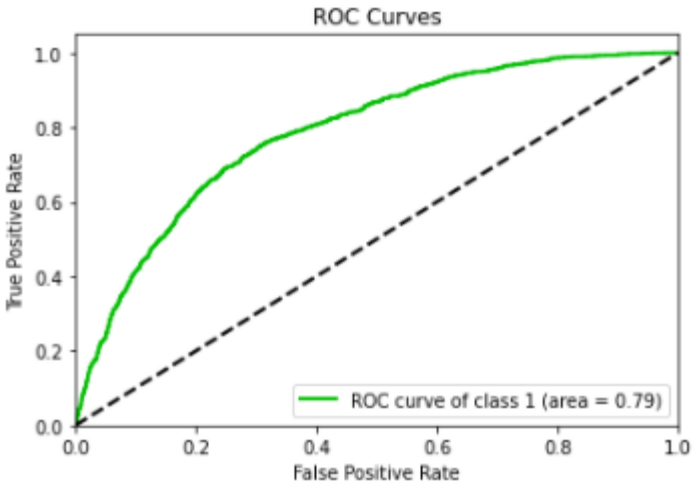

Figura 19 Curva ROC – Modelo XGBoosting

4.5 Evaluación

A continuación, se expone una evaluación de los resultados generales, así como una revisión de los aspectos por mejorar.

4.5.1 Evaluar los resultados

Luego de analizar los resultados arrojados por los distintos modelos e hiperparametros probados, concluimos que el modelo con mejores niveles de predicción es el resultante del algoritmo LightGBM, con los hiperparametros configurados de la siguiente manera:

- colsample_bytree: 0.67
- learning_rate :0.01
- max_depth: 42
- min_child_samples: 483
- min_child_weight: 100.0
- num_leaves: 37
- reg_alpha: 0
- reg_lambda: 1

En el dataset de evaluación, nos arroja un accuracy de 0,718, siendo levemente mayor al resto de los modelos probados, y alcanzo nuestro objetivo propuesto (resultados mayores a 0,7 de accuracy).

Respecto al test KS (Kolmogorov-Smirnov), nos arroja un resultado de 0,431.

A modo de entender los resultados, analizamos el peso que tienen las variables dentro del modelo mediante la librería eli5 en Python. La lista de variables por importancia nos quedo ordenada de la siguiente manera (Extrajimos las 15 más importantes):

Peso	Variable	Descripción
0.1675	sector_2	¿Es sector privado o público?
0.0404	ap40_04_4.0	¿En qué medida estás de acuerdo con la siguiente afirmación? Disfruto estudiando Matemática - Nada de Acuerdo
0.0396	ap40_01_1.0	¿Cuán de acuerdo estás con la siguiente afirmación sobre la convivencia en tu escuela? Hay un ambiente de buena convivencia - Nada de Acuerdo
0.0300	edadA_junio2019_17.0	¿Edad de 17 años?
0.0297	ap24_1.0	¿Asististe a nivel inicial?
0.0262	ap40_02_1.0	¿En qué medida estás de acuerdo con la siguiente afirmación? Me interesan las clases de Matemática en mi escuela
0.0244	ap17_6.0	Nivel educativo del padre: ¿Educación Superior universitaria?
0.0199	ap39_04_2.0	En general, ¿cómo te resultan Resolver problemas y ejercicios?
0.0172	isocioa_3.0	¿Índice socioeconómico Alto del alumno?
0.0166	ap40_01_4.0	¿En qué medida estás de acuerdo con la siguiente afirmación? Disfruto estudiando Matemática - Muy de Acuerdo
0.0166	ap40_01_3.0	¿En qué medida estás de acuerdo con la siguiente afirmación? Disfruto estudiando Matemática - De Acuerdo

0.0158	ap02_2.0	¿Sexo?
0.0158	ap16_5.0	Nivel educativo de la madre: ¿Educación Superior Técnica?
0.0158	ap25_01_1.0	¿Repetiste algún año durante tu escolaridad? Primaria
0.0157	ap34_2.0	¿Qué vas a hacer cuando termines el secundario? Seguir estudiando en educación universitaria

Si vemos la distribución de algunas de estas variables con el atributo target, vemos que existe una fuerte correlación entre ellas:

Figura 20. Distribución de desaprobados según atributo Sector

Figura 21. Distribución de desaprobados según atributo isocioa = 3

Figura 22. Distribución de desaprobados según atributo edadA_junio2019 = 17

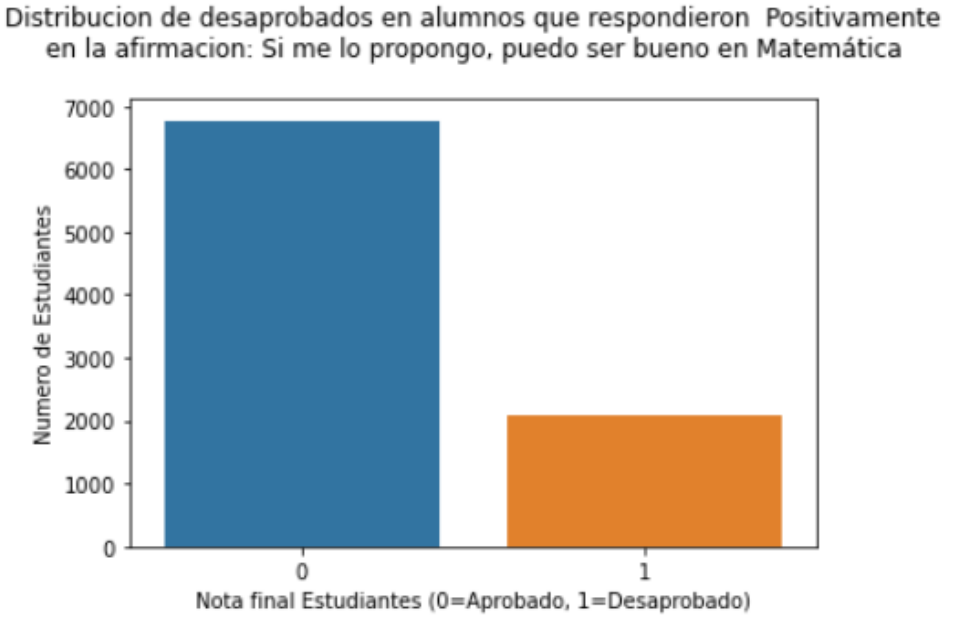

Figura 23. Distribución según atributo ap40_04 = 4

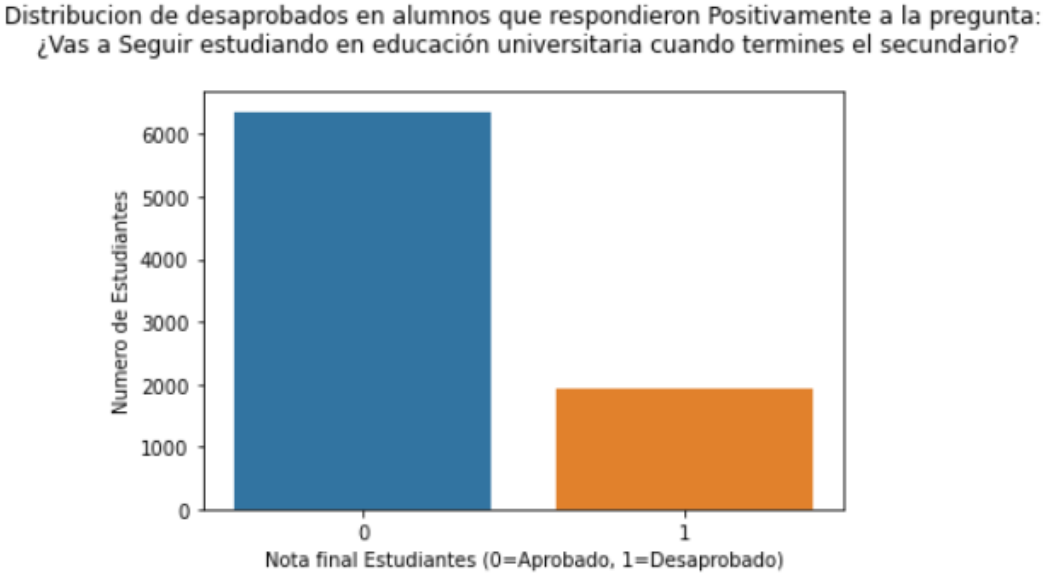

Figura 24. Distribución según atributo ap34 = 2

4.5.2 Revisión del proceso

Haciendo una revisión global del proceso de modelado, vemos que se utilizaron todos los algoritmos relevantes, y obtuvimos resultados satisfactorios con el dataset de pruebas. Si bien hubo limitaciones de procesamiento por la tecnología utilizada, esto no impidió generar resultados representativos con respecto al total de la población

4.5.3 Próximos pasos

Como próximos pasos del proyecto, esta estipulado probar la efectividad del modelo ganador en las futuras encuestas de Aprender, para ver la estabilidad del modelo a lo largo de los años posteriores

4.6 Implementación

A partir de los resultados del modelo, surgen recomendaciones de aplicación de los resultados, ejecución del modelo en otros conjuntos de datos y preguntas que podrían generar nuevas investigaciones. A continuación, se presentan dichas recomendaciones.

4.6.1 Plan de implementación

La implementación que se recomienda, es separar a los alumnos en 10 grupos diferentes, según su riesgo de desaprobación que resulta de la probabilidad que arroja nuestro modelo.

De esta manera, separamos a la población de la siguiente forma (Datos tomados a partir del dataset de evaluación):

Ranking Riesgo	Alumnos totales	Desaprobados	Tasa de desaprobación	Captura acumulada de desaprobados
1	500	343	69%	22%
2	500	304	61%	41%
3	500	248	50%	57%
4	500	202	40%	70%

5	500	138	28%	78%
6	500	123	25%	86%
7	500	94	19%	92%
8	500	65	13%	96%
9	500	45	9%	99%
10	500	15	3%	100%
Total	**5.000**	**1.577**	**32%**	**-**

Luego de separar a la población en grupos, se recomienda realizar alguna acción de corrección temprana sobre los alumnos que se encuentran en los rankings 1, 2, 3 y 4. De esta manera, estaremos dando apoyo anticipado al 70% de los alumnos que desaprueban el examen (2.000 alumnos, sobre la muestra total de 5.000).

El resto de los rankings, al no ser de riesgo, no es necesario realizar ninguna acción diferente.

4.6.2 Plan de monitoreo y mantención

El plan de monitoreo para el modelo, se establece según lo indican los pasos futuros del proyecto, aplicando el modelo de predicción en futuras encuestas de Aprender, para validar que siga siendo efectivo y con un ordenamiento estable.

4.6.3 Informe final

Como informe final del modelado, se puede establecer que alcanzamos las metas proyectadas, y pudimos resolver los conflictos de limpieza y estructura de datos de manera positiva. Como resultado final, tenemos un modelo con buenos niveles de predicción, y estabilidad, para poder utilizar con las futuras encuestas de Aprender, o para aplicar en otro ámbito similar.

4.6.4 Revisión de proyecto

En la revisión del proyecto, hacemos una validación sobre los objetivos previamente mencionados:

- ✓ Realizar un análisis de datos sobre la encuesta Aprender
- ✓ Adaptar los datos para poder ser procesados en un modelo predictivo
- ✓ Aplicar un algoritmo de predicción, con un Accuracy mayor a 0,7
- ✓ Definir variables de mayor peso con respecto a la nota del alumno
- ✓ Establecer un plan de implementación para ayudar a los alumnos más críticos

5. Líneas Finales

En el siguiente capítulo, se presentan las conclusiones del trabajo, en conjunto con información adicional referente al mismo.

5.1 Conclusión

La aplicación de técnicas de minería de datos sobre distintas fuentes de datos puede revelar información que no se conocía hasta el momento. La misma le otorga un foco de mayor detalle al análisis, y puede mejorar, improvisar, y adelantarse a problemas sobre cualquier contexto.

En nuestro estudio, vemos que, utilizando las técnicas de minería, revelamos información innovadora en el campo de la educación nacional. En base al modelo que armamos, se puede tener una ganancia real y demostrada con los planes de implementación propuestos.

El machine learning no es una herramienta que de soluciones por si sola, sino que debe estar acompañada en todo el proceso, por un grupo de personas involucradas y con distintos roles. Pero si al conocimiento del experto, se lo complementa con modelos de datos, puede dar como resultado una experiencia superior, y más eficiente.

Por otro lado, probamos soluciones con distintos algoritmos, y vimos que todos arrojan resultados positivos, aunque algunos demuestran mayor poder de predicción que otros.

El lenguaje Python, junto con sus librerías y herramientas, nos brinda una gran ayuda para encontrar soluciones a los problemas encontrados, y nos facilita el procesamiento y limpieza de la información. Por otro lado, nos permite graficar de manera analítica los distintos aspectos del dataset.

La metodología CRISP-DM nos facilitó la organización del proyecto, así como nos estableció un marco practico y metodológico apto para desarrollar y optimizar los modelos de machine learning en un entorno controlable y escalable

5.2 Líneas futuras

Todo el trabajo fue realizado con las encuestas realizadas durante el 2019 a los alumnos a nivel nacional. Por limitaciones en el hardware disponible para el procesamiento de datos, se realizaron las pruebas con una muestra aleatoria del total de la población. En el futuro, nuestra expectativa es poder probar el modelo final con datos del 2020 y 2021 sobre las encuestas de Aprender y evaluar si el nivel de predicción se mantiene estable. Por otro lado, nos queda pendiente poder volver a entrenar el modelo de machine learning en un servidor con mayor capacidad de procesamiento, para poder probar con toda la población, y no limitarnos a muestras aleatorias.

Bibliografía

Abdelmalik , M., Inza, I., & Larrañaga, P. (s.f.). *Clasificadores K-NN.* Universidad del Paıs Vasco–Euskal Herriko Unibertsitatea.

Azevedo, A., & Santos, M. (2008). KDD, semma and CRISP-DM: A parallel overview. *IADIS European Conference on Data Mining* . Amsterdam, The Netherlands.

Beinlich, I., Suermondt, H., Chavez, R., & Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *In proceedings of the 2nd European Conference on Artificial Intelligence in Medicine. .*

Bickmore, T. W. (1994). Real-Time Sensor Data Validation. *NASA Contractor Report 195295, National Aeronautics and Space Administration.*

Britos, P., & García-Martínez, R. (2009). *Propuesta de Procesos de Explotación de Información.*

Britos, P., Felgaer, P., & García-Martínez, R. (2008). Bayesian Networks Optimization Based on Induction Learning Techniques. *In Artificial Intelligence in Theory and Practice II.* Boston: Springer: M. Bramer.

Britos, P., Jiménez Rey, E., & García-Martínez, E. (2008). Work in Progress: Programming Misunderstandings Discovering Process Based On Intelligent Data Mining Tools. *Proceedings 38th ASEE/IEEE Frontiers. Education Conference.*

Brownlee, J. (2020). *Gradient Boosting with Scikit-Learn, XGBoost, LightGBM, and CatBoost.*

Chapman, P., Clinton, J., R., K., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0.*

Ezawa, K. J., & Schuermann, T. (1995). Fraud/Uncollectible Debt Detection Using a Bayesian Network Based Learning System: A Rare Binary Outcome with Mixed Data Structures. *Proceedings of the Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA,*, 157-166.

Fayyad, U. M. (1996). From data mining to knowledge discovery: an overview. *Advances in knowledge discovery and data mining*.

Felgaer, P. (2004). Optimización de Redes Bayesianas basado en Técnicas de Aprendizaje por Inducción. *Reportes Técnicos en Ingeniería del Software*, págs. 64-69.

Fogarty, J., Baker, R., & Hudson, S. (2005). Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction. *ACM International Conference Proceeding Series, Proceedings of Graphics Interface 2005*. Waterloo, Ontario, Canada.

Granado, E. C. (s.f.). *Manual de uso de Jupyter notebook para aplicaciones docentes.* Madrid: Universidad Complutense de Madrid.

Grigori, D., Casati, F., Castellanos, M., Dayal, u., Sayal, M., & Shan, M. (2004). Business Process Intelligence. *Computers in Industry 53*, págs. 321-343.

Heckerman, D. (1995). A tutorial on learning bayesian networks. *Technical report MSR-TR-95-06, Microsoft research, Redmond, WA*.

Heckerman, D., Chickering, M., & Geiger, D. (1995). Learning bayesian networks, the combination of knowledge and statistical data. *Machine learning 20*, 197-243.

Indec. (s.f.). *Quiénes somos: Instituto Nacional de Estadistica y Censos*. Obtenido de Instituto Nacional de Estadistica y Censos: https://www.indec.gob.ar/indec/web/Institucional-Indec-QuienesSomos-1

Kisbye, P. (2010). *Test de Kolmogorov-Smirnov.* FaMAF.

Kononenko, I., & Cestnik, B. (1986). *Lymphography Data Set*. Obtenido de UCI Machine Learning: http://archive.ics.uci.edu/ml/datasets/Lymphography

Langseth, J., & Vivatrat, N. (2003). Why Proactive Business Intelligence is a Hallmark of the Real-Time Enterprise: Outward Bound. *Intelligent Enterprise 5*, págs. 34-41.

Mata, M. C., & Macassi, S. (1997). *Cómo elaborar muestras para los sondeos de audiencias.* Quito.

Michalski, R. (1983). A Theory and Methodology of Inductive Learning. *Artificial Intelligence, 20*, págs. 111-161.

Michalski, R., Bratko, I., & Kubat, M. (1998). Machine Learning and Data Mining, Methods. *John Wiley & Sons*.

Mitra, S., & Acharya, T. (2003). *Data mining: multimedia, soft computing and bioinformatics.* John Wiley & Sons.

Negash, S., & Gray, P. (2008). Business Intelligence. *En Handbook on Decision Support Systems 2, ed. F. Burstein y C. Holsapple*.

Pearl, J. (1988). Probabilisic reasoning in intelligent systems. *Morgan Kaufmann, San Mateo, CA*.

Perez, J., Henriques, M. F., Pazos, R., Cruz, L., Reyes, G., Salinas, J., & Mexicano, A. (2007). La IO Aplicada a la solucion de problemas regionales. *2do Taller Latino Liberoamericano de Investigacion de Operaciones*.

Pramoditha, R. (s.f.). *blog Data Science 365.* Obtenido de https://medium.com/data-science-365

Riquelme, J., Ruiz, R., & Gilbert, K. (2006). Minería de Datos: Conceptos y Tendencias . *Revista Iberoamericana de Inteligencia Artificial.*

Silvente, Hurtado, & Baños. (2013). Cómo aplicar árboles de decisión en SPSS. *Reire*.

Soloaga, A. (2018). *Akademus.* Obtenido de www.akademus.es

Sumathi, & Sivanandam. (2006). *Introduction to Data Mining and its Applications.*

Witten, I., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques 2nd Edition.* Morgan Kaufmann.

Anexo

CUESTIONARIO DE ENCUESTAS APRENDER 2019

Los cuestionarios oficiales del Ministerio de Educación los encuentran en la ruta:

https://drive.google.com/file/d/1UwFOUxcPabof0TxA-Bzt2PbhOX4bWAuH/view

Analisis de respuestas a ítems de Matemática (Informe realizado por el Ministerio de Educación):

https://www.argentina.gob.ar/sites/default/files/analisis_de_respuestas_a_items_de_matematica.pdf