Escuela de Administración

Escuela de Ingeniería, Ciencia y Tecnología

Maestría en Business Analytics

Diseño de un modelo predictivo del desempeño académico de estudiantes que ingresen a la educación superior.

Presentado por:

José Luis Soto Dueñas

Sindy Yuliana Acevedo Tarazona

Bogotá, D.C. 3 de marzo de 2024

Escuela de Administración

Escuela de Ingeniería, Ciencia y Tecnología

Maestría en Business Analytics

Diseño de un modelo predictivo del desempeño académico de estudiantes que ingresen a la educación superior.

Presentado por:

José Luis Soto Dueñas

Sindy Yuliana Acevedo Tarazona

Bajo la dirección de:

John Pablo Calvo López

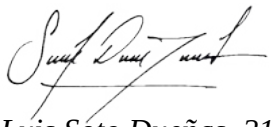
Bogotá, D.C. 3 de marzo de 2024

Tabla de contenido

Abreviaturas 6
Lista de anexos 8
Lista de figuras 9
Lista de tablas 11
Resumen ejecutivo 12
Palabras clave 13
Abstract 14
Keywords 15
1 Introducción 16
2 Objetivos 18
2.1 Objetivo general 18
2.2 Objetivos específicos 18
3 Alcance del proyecto aplicado 19
4 Estrategia para lo solución 20
4.1 Comprensión del negocio 23
4.2 Comprensión de los datos 28
4.2.1 Base de datos 28
4.2.2 Estadísticas descriptivas de variables numéricas 30
4.2.3 Distribución de las variables numéricas 32
4.2.4 Correlación entre variables numéricas y la variable respuesta 33
4.2.5 Distribución de la variable respuesta 34
4.2.6 Análisis de correlación entre variable numéricas 37
4.2.7 Reducción de variables numéricas 38
4.2.8 Estadísticas de variables cualitativas 40
4.2.9 Distribución de variables cualitativas 42
4.2.10 Correlación entre variables cualitativas y la variable respuesta 43
4.2.11 Análisis de correlación entre variables cualitativas 47
4.2.12 Reducción de variables cualitativas 48
4.3 Preparación de los datos 49
4.3.1 División del conjunto de datos en subconjuntos de entrenamiento y prueba 51

4.3.2 Identificación de valores ausentes 55
4.3.3 Estandarización de variables numéricas 55
4.4 Modelaje 57
4.4.1 Entrenamiento 57
4.4.2 Diagnóstico de residuos de la validación cruzada 61
4.4.3 Regresión Lineal (Ridge) 64
4.4.4 K-Nearest Neighbor (KNN) 68
4.4.5 Random Forest 71
4.4.6 Gradient Boosting Trees 74
4.4.7 Stacking 78
4.5 Evaluación 80
4.5.1 Análisis de la relevancia en el modelo analítico seleccionado 83
4.6 Despliegue 85
4.6.1 Pronóstico del éxito académico 86
5 Cronograma 88
6 Conclusiones 89
7 Recomendaciones 91
8 Referencias bibliográficas 92
9 Anexos 94

a. *Declaración de autonomía:* Declaro(amos) bajo gravedad de juramento, que he(mos) escrito la presente tesis de maestría por mi(nuestra) propia cuenta, y que, por lo tanto, su contenido es original. Declaro(amos) que he(mos) indicado clara y precisamente todas las fuentes directas e indirectas de información, y que esta tesis de maestría no ha sido entregada a ninguna otra institución con fines de calificación o publicación.

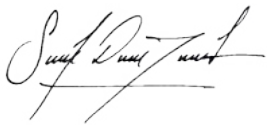


José Luis Soto Dueñas, 21 de marzo de 2024

Sindy Yuliana Acevedo Tarazona, 21 de marzo de 2024

John Pablo Calvo López, 21 de marzo de 2024

b. *Declaración de exoneración de responsabilidad:* Declaro(amos) que la responsabilidad intelectual del presente trabajo es exclusivamente de su(s) autor(es). La Universidad del Rosario no se hace responsable de contenidos, opiniones o ideologías expresadas total o parcialmente en él.



José Luis Soto Dueñas, 21 de marzo de 2024

Sindy Yuliana Acevedo Tarazona, 21 de marzo de 2024

John Pablo Calvo López, 21 de marzo de 2024

Abreviaturas

ICFES: Instituto Colombiano para la Evaluación de la Educación.

Saber 11: Examen de estado de la educación media saber 11.

Saber Pro: Examen de estado de calidad de la educación superior.

SNIES: Sistema Nacional de información de la Educación Superior.

CRIPS-DM: Cross-Industry Standard Process for Data Mining

NSE: Nivel socioeconómico

Lista de anexos

Anexo A Datos brutos pruebas saber 94

Anexo B Script construcción modelos 94

Anexo C Ejecución scripts construcción modelos 94

Lista de figuras

Figura 1. *Ciclo de vida de minería de datos metodología CRISP-DM* 22

Figura 2. *Actividades para desarrollar dentro de la metodología CRISP-DM* 23

Figura 3. *Distribución de las variables numéricas* 32

Figura 4. *Correlación entre variables numéricas y la variable respuesta* 34

Figura 5. *Distribución de la variable respuesta* 35

Figura 6. *Análisis de correlación entre variable numéricas* 37

Figura 7. *Reducción de variables numéricas* 40

Figura 8. *Distribución de variables cualitativas* 43

Figura 9. *Correlación entre variables cualitativas y la variable respuesta* 46

Figura 10. *Análisis de correlación entre variables cualitativas* 48

Figura 11. *Reducción de variables cualitativas* 49

Figura 12. *Reducción de variables cualitativas* 52

Figura 13. *Distribución de los datos de entrenamiento por área del conocimiento* 54

Figura 14. *Distribución de los datos de pruebas por área del conocimiento* 54

Figura 15. *Distribución del error de la validación cruzada* 61

Figura 16. *Diagnóstico de residuos de la validación cruzada* 62

Figura 17. *Distribución de la objetivo y datos predichos* 64

Figura 18. *Comparación del desempeño de los modelos* 82

Figura 19. Relevancia de las variables del modelo 84

Figura 20. Cronograma del proyecto 88

Lista de tablas

Tabla 1. Diccionario de datos 30

Tabla 2. Estadísticas descriptivas de variables numéricas 31

Tabla 3. Evaluación de la variable respuesta con diversas distribuciones 36

Tabla 4. Estadísticas de variables cualitativas 41

Tabla 5. Evaluación del modelo de Regresión Lineal (Ridge) 68

Tabla 6. Evaluación del modelo K-Nearest Neighbor (KNN) 71

Tabla 7. Evaluación del modelo Random Forest 74

Tabla 8. Evaluación del modelo Gradient Boosting Trees 78

Tabla 9. Evaluación del modelo Stacking 80

Tabla 10. Comparación del desempeño de los modelos 81

Tabla 11. Datos de estudio de caso 86

Tabla 12. Predicción del modelo estudio de caso 87

Resumen ejecutivo

Este proyecto se enfoca en el desarrollo de modelos predictivos para evaluar el desempeño de los estudiantes en las pruebas Saber Pro, con énfasis en áreas específicas del conocimiento en la educación superior, que incluyen Agronomía, Veterinaria, Bellas Artes, Ciencias de la Educación, Ciencias de la Salud, Ciencias Sociales y Humanas, Economía, Administración, Contaduría, Ingeniería, Arquitectura, Urbanismo, Matemáticas y Ciencias Naturales. Estos modelos tienen como objetivo identificar y jerarquizar las áreas de estudio con mayor influencia en el rendimiento en las pruebas, proporcionando orientación para la toma de decisiones relacionadas con el ingreso a la educación superior.

Los modelos se basan en los resultados obtenidos en las pruebas Saber 11 y Saber Pro en Colombia, que evalúan las competencias generales y específicas de los estudiantes de bachillerato y universitarios. Se centran en la identificación de variables socioeconómicas, educativas y de desempeño académico relevantes para anticipar el éxito de los estudiantes en la educación superior.

El desarrollo de estos modelos predictivos seguirá la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), que consta de seis fases: comprensión del problema, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. La aplicación de esta metodología garantizará un análisis de los datos de las pruebas Saber, asegurando la calidad del modelo predictivo y evaluando su efectividad mediante indicadores de precisión.

En última instancia, el modelo propuesto brindará a los estudiantes que cuenten con resultados de la prueba Saber 11 una estimación del puntaje de desempeño en las pruebas Saber Pro para cada área del conocimiento en la educación superior. Este enfoque permitirá que los estudiantes tomen decisiones informadas y basadas en datos al seleccionar su área de estudio, lo que optimizará sus posibilidades de éxito en la educación superior.

Palabras clave

Saber 11, Saber Pro, resultados, área del conocimiento, modelo analítico predictivo, metodología CRISP-DM.

Abstract

This project focuses on the development of predictive models to assess the performance of students in the Saber Pro tests, with an emphasis on specific knowledge areas in higher education, including Agronomy, Veterinary Sciences, Fine Arts, Education Sciences, Health Sciences, Social and Human Sciences, Economics, Management, Accounting, Engineering, Architecture, Urbanism, Mathematics, and Natural Sciences. These models aim to identify and prioritize the areas of study that have the most significant influence on performance in the tests, providing guidance for decisions related to entering higher education.

The models are based on the results obtained in the Saber 11 and Saber Pro tests in Colombia, which evaluate the general and specific competencies of high school and university students. They focus on identifying socio-economic, educational, and academic performance variables that are relevant in predicting students' success in higher education.

The development of these predictive models will follow the CRISP-DM methodology (Cross-Industry Standard Process for Data Mining), which comprises six phases: understanding the problem, understanding the data, data preparation, modeling, evaluation, and deployment. The application of this methodology will ensure a rigorous analysis of the Saber test data, ensuring the quality of the predictive model and evaluating its effectiveness through precision indicators.

Ultimately, the proposed model will provide students with Saber 11 results with an estimation of their performance score in the Saber Pro tests for each knowledge area in higher education. This approach will enable students to make informed, data-driven decisions when selecting their field of study, thereby optimizing their chances of success in higher education.

Keywords

Saber 11, Saber TyT, Saber Pro, results, knowledge area, predictive analytical model and CRISP-DM methodology.

1 Introducción

La educación superior se concibe como una de las principales rutas para acceder a oportunidades laborales y elevar la calidad de vida. No obstante, el recorrido de los estudiantes en programas académicos superiores no siempre culmina con éxito, pues las estadísticas revelan que únicamente el 50% de quienes se aventuran en esta etapa logran obtener su título (Banco Mundial, 2017).

Frente a esta problemática, se plantea la concepción de un proyecto empresarial que tiene por objetivo anticipar y mejorar las tasas de éxito de los estudiantes en la educación superior, empleando modelos predictivos como herramientas. Estos modelos no solo se proponen identificar las áreas específicas del conocimiento en la educación superior que tienen un historial de altos desempeños académicos, sino que también buscan jerarquizarlas. Para este propósito, se analizarán áreas que abarcan desde Agronomía, Veterinaria, Bellas Artes, Ciencias de la Educación, Ciencias de la Salud, Ciencias Sociales y Humanas, Economía, Administración, Contaduría, Ingeniería, Arquitectura, Urbanismo, hasta Matemáticas y Ciencias Naturales.

Estos modelos se cimentarán en los resultados de las pruebas Saber 11 y Saber Pro en Colombia, pruebas estandarizadas que evalúan tanto competencias generales como específicas de estudiantes de bachillerato y universitarios. El énfasis recae en la identificación de variables socioeconómicas, educativas y de desempeño académico que tienen un peso significativo en la predicción del éxito académico de los estudiantes que ingresan a la educación superior.

El desarrollo de estos modelos predictivos seguirá una metodología bien establecida, el CRISP-DM (Cross-Industry Standard Process for Data Mining), que consta de seis fases interconectadas: comprensión del problema, análisis de los datos, preparación de los datos, modelado, evaluación y puesta en práctica. La aplicación de esta metodología garantizará una evaluación de los datos obtenidos de las pruebas Saber, asegurando la calidad del modelo predictivo y permitiendo evaluar su eficacia mediante indicadores de precisión. Esta metodología, ampliamente reconocida en la

industria de la minería de datos, aportará un valor significativo al proyecto como una herramienta sólida para el análisis y la toma de decisiones.

En última instancia, el modelo propuesto ofrecerá a todos los estudiantes que cuentan con resultados de la prueba Saber 11 una estimación de su rendimiento académico en las pruebas Saber Pro para cada área del conocimiento en la educación superior. Este enfoque permitirá que los estudiantes tomen decisiones informadas y basadas en datos al seleccionar su área de estudio, lo que, a su vez, maximizará sus oportunidades de éxito en la educación superior.

2 Objetivos

2.1 Objetivo general

Desarrollar un modelo analítico para anticipar el desempeño académico de los estudiantes en su educación superior, con un enfoque específico en las áreas del conocimiento a las que aspiran ingresar.

2.2 Objetivos específicos

- Identificar las variables socioeconómicas, educativas y de desempeño académico que sean más relevantes en la predicción del rendimiento académico de los estudiantes en las diversas áreas del conocimiento de la educación superior.
- Aplicar una metodología coherente y lógica en la creación de modelos analíticos de predicción, garantizando su validez y fiabilidad.
- Evaluar la eficacia de los modelos predictivos mediante la utilización de indicadores de precisión y seleccionar el más adecuado para su implementación.

3 Alcance del proyecto aplicado

El alcance de este proyecto tiene como propósito principal el desarrollo, validación y elección de un modelo predictivo destinado a estimar los posibles resultados del desempeño académico de áreas de estudio de la educación superior para estudiantes de bachillerato que deseen ingresar a cursar programas universitarios, brindándoles información predictiva de qué áreas son las que mejor le conviene en su transitar universitario. Los resultados esperados para el proyecto empresarial se consolidan de la siguiente forma.

- Un modelo analítico completo que permita anticipar el desempeño académico de los estudiantes en su educación superior, con un enfoque específico en las áreas del conocimiento a las que aspiran ingresar.
- Una lista de las variables socioeconómicas, educativas y de desempeño académico más relevantes para la predicción del rendimiento académico de los estudiantes en las diversas áreas del conocimiento de la educación superior.
- Modelos analíticos de predicción desarrollados con una metodología coherente y lógica, garantizando su validez y fiabilidad.
- Un informe de evaluación que incluye la eficacia de los modelos predictivos, utilizando indicadores de precisión, y una recomendación sobre el modelo más adecuado para su implementación.

4 Estrategia para lo solución

El proyecto empresarial adoptó la metodología CRISP-DM, considerada general para la minería de datos y adaptable a diferentes entornos de trabajo sin prescribir un conjunto específico de técnicas o herramientas. Los analistas de datos pueden utilizar diferentes técnicas y herramientas según el proyecto específico, siempre y cuando sigan el proceso general de la metodología CRISP-DM (Chapman et al., 2000). Por tanto, la metodología se alineó perfectamente al proyecto empresarial, que se centró en desarrollar un modelo analítico para anticipar el desempeño académico de los estudiantes en su educación superior, con un enfoque específico en las áreas del conocimiento a las que aspiran ingresar. La metodología permitió llevar a cabo las actividades de alistamiento y limpieza de los datos necesarias, así como la elaboración de varios modelos y la elección de uno de ellos que pudiera estimar el puntaje de desempeño académico de la prueba de educación superior saber Pro.

Es importante mencionar que la metodología CRISP-DM se enfoca en la iteración y retroalimentación continua en el proceso de minería de datos, lo que permite la adaptabilidad y flexibilidad en su aplicación (Wu et al., 2013). De esta manera, se pueden realizar ajustes en la metodología a medida que se obtienen nuevos conocimientos o se identifican problemas en el proceso. Además, la metodología CRISP-DM destaca la importancia de la participación de expertos en el dominio del problema en cada etapa del proceso para garantizar que los resultados sean relevantes y útiles para el negocio o la industria en la que se aplican (Kohavi et al., 2002). En consecuencia, En consecuencia, se llevaron a cabo entrevistas con personal experto del área de investigación del ICFES, entidad que compartió los datos para comprender los procesos y particularidades de la evaluación de la educación en Colombia, y para garantizar que el modelo considerara las variables más representativas de la evaluación realizada por el Instituto Colombiano para la Evaluación de la Educación – ICFES.

La metodología CRISP-DM se divide en seis fases principales, las cuales son: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue (Chapman et al., 2000). La primera fase, comprensión del negocio, implica definir los objetivos del proyecto, establecer los criterios de éxito y evaluar los recursos necesarios para llevar a cabo el proyecto. La comprensión del negocio también implica la identificación de los interesados y sus necesidades, así como la definición de los problemas que se deben resolver. En esta fase, es crucial contar con la participación de expertos del dominio para garantizar que los resultados sean relevantes y útiles (Kohavi et al., 2002).

La segunda fase, comprensión de los datos, se centra en la recopilación y exploración de los datos disponibles. En esta fase, se busca entender la calidad y la naturaleza de los datos, así como identificar posibles problemas en los mismos. Además, se pueden realizar análisis exploratorios y visualizaciones para identificar patrones o relaciones entre las variables (Wu et al., 2013). La fase de preparación de los datos se enfoca en la limpieza, la integración y la transformación de los datos para prepararlos para el modelado. Esta fase es crucial para garantizar que los datos sean consistentes, precisos y relevantes para el modelo (Kohavi et al., 2002).

La fase de modelado es donde se construyen los modelos predictivos. Esta fase implica la selección de las técnicas y algoritmos apropiados, la construcción del modelo y la validación de este (Chapman et al., 2000). En la fase de evaluación, se examina el desempeño del modelo y se realizan pruebas para medir su precisión y robustez. Además, se pueden realizar comparaciones con otros modelos para determinar cuál es el más adecuado (Wu et al., 2013).

Por último, la fase de despliegue implica la implementación del modelo en el entorno operativo y la integración con los procesos de negocio (Chapman et al., 2000). Esta fase también puede incluir la formación de los usuarios finales y la monitorización del modelo para garantizar su eficacia a lo largo del tiempo (Kohavi et al., 2002). A continuación, se presenta la figura del ciclo de la metodología CRISP-DM descrita, recalcando que en cada fase se puede ejecutar de manera

consecutiva y, si es necesario, volver a una fase anterior, además, el ciclo es iterativo y se puede aplicar a cualquier proyecto que involucre trabajo de datos.

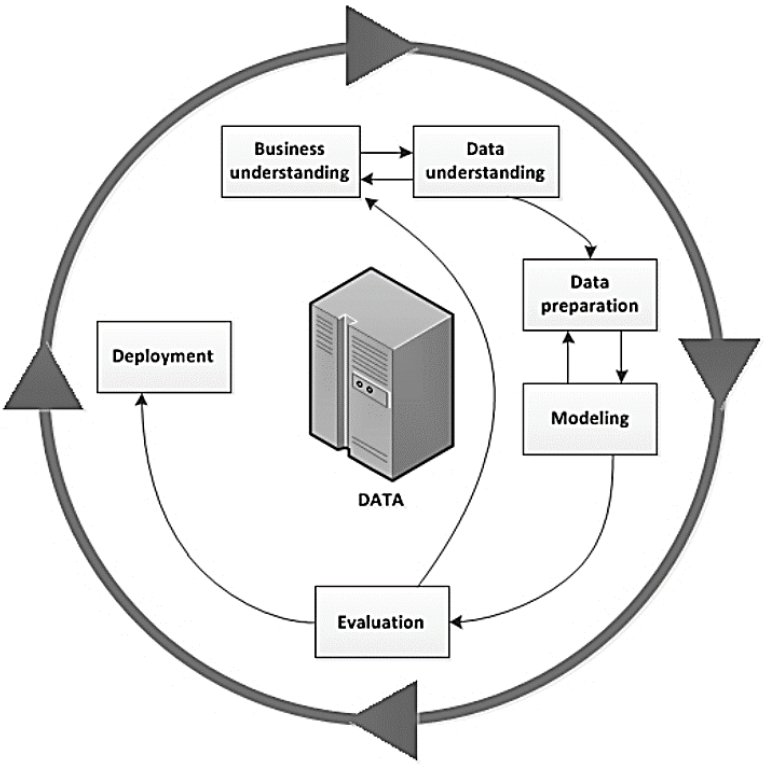


Figura 1. *Ciclo de vida de minería de datos metodología CRISP-DM*
Fuente: metodología CRISP-DM de IBM.

Tomando como referencia la metodología anterior, el proyecto se planteó desarrollas las siguientes actividades:

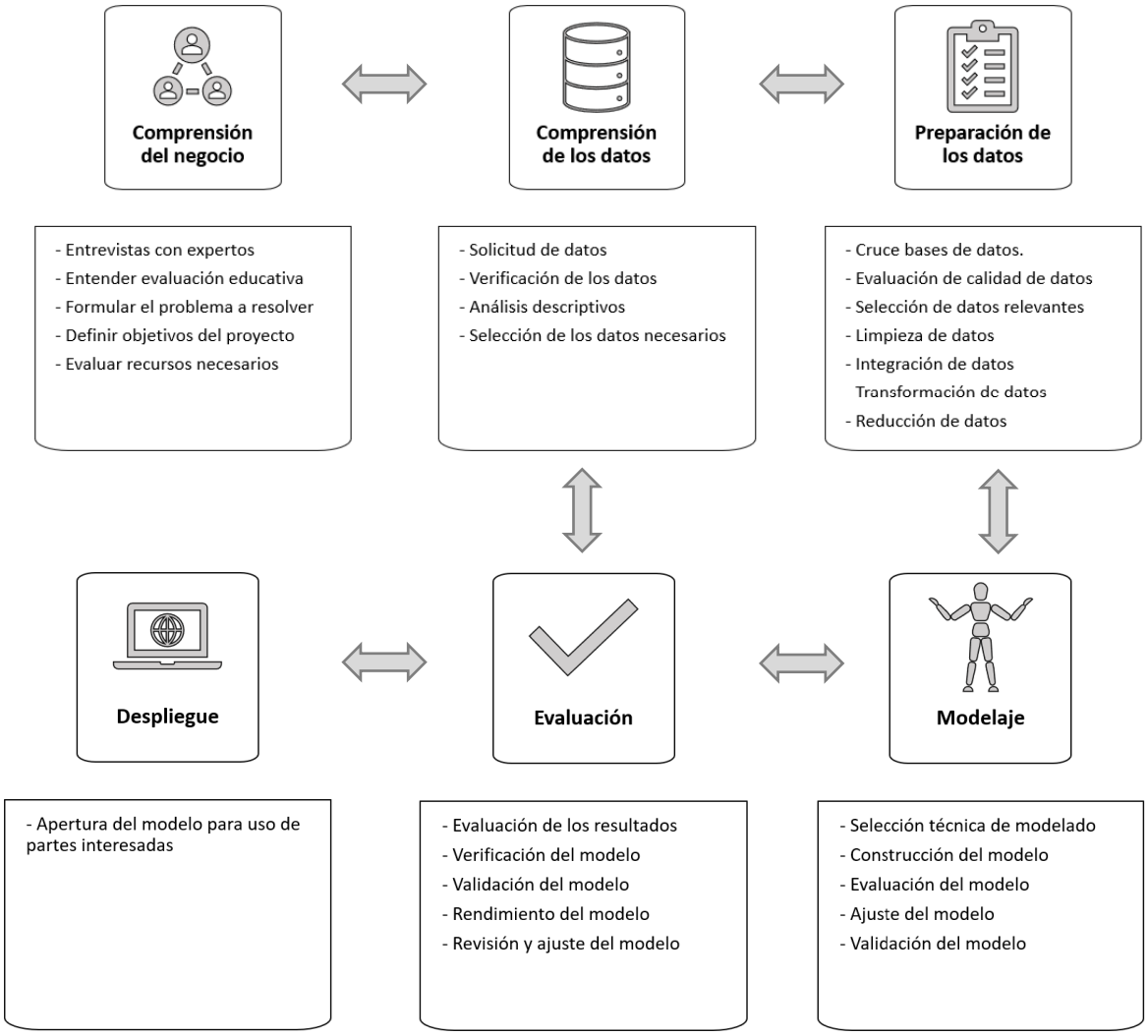


Figura 2. *Actividades para desarrollar dentro de la metodología CRISP-DM*
Fuente: Elaboración propia.

4.1 Comprensión del negocio

La fase de comprensión del negocio es fundamental para asegurar el éxito de cualquier proyecto de minería o análisis de datos. En la metodología CRISP-DM, esta fase se centra en entender los objetivos del negocio, los requisitos del proyecto, las restricciones y los recursos disponibles. En el contexto de la educación en Colombia, resulta crucial comprender cómo se evalúa a los estudiantes y cómo se monitorea la calidad de la formación de las instituciones educativas. Se debe tener en

cuenta que la evaluación educativa es un proceso complejo que implica la integración de múltiples factores, como los objetivos de aprendizaje, los contenidos, los métodos de enseñanza y las características del estudiante (Hernández et al., 2014).

Para lograr una comprensión completa del negocio, se realizaron entrevistas con expertos en medición de la evaluación y áreas técnicas de análisis de la entidad que recolecta los datos de medición, así como se examinaron documentos y publicaciones relevantes. Además, es esencial entender la metodología de evaluación de los estudiantes en Colombia. Como se mencionó previamente, el Instituto Colombiano para la Evaluación de la Educación (ICFES) es responsable de evaluar a los estudiantes en diferentes niveles educativos y monitorear la calidad de la formación de las instituciones educativas. En una reunión con un experto en el negocio, se pudo conocer que el instituto está trabajando en una nueva metodología de evaluación para comparar los resultados de los estudiantes a lo largo de su desarrollo académico en la vida escolar, de bachillerato, técnica y superior y evaluar la educación a nivel nacional. Precisamente, estos datos son los que se lograron acceder para el desarrollo del presente proyecto empresarial. La evaluación de la educación se realiza mediante la aplicación de pruebas de matemáticas, lectura crítica, ciencias naturales, ciencias sociales e inglés para medir el desempeño académico, y se caracteriza a cada estudiante en variables económicas y sociales.

Para formular el problema a resolver, es necesario entender los requisitos y objetivos del proyecto. En nuestro caso, el objetivo es desarrollar un modelo analítico para anticipar el desempeño académico de los estudiantes en su educación superior, con un enfoque específico en las áreas del conocimiento a las que aspiran ingresar según las variables identificadas, con el fin de evitar la deserción y la frustración en sus proyectos de vida. Para lograr este objetivo, se evaluaron los recursos necesarios y se identificaron las limitaciones del proyecto. Se encontró que los datos históricos con los que se contó para el desarrollo del proyecto son a partir de 2014 de estudiantes que cursaron y presentaron las pruebas Saber de bachillerato y que posteriormente ingresaron a la

educación profesional y presentaron las pruebas Saber de educación superior disponibles hasta el 2021. Los recursos necesarios de tiempo de desarrollo, hardware y software estudiantil fueron proporcionados por los autores del presente trabajo, de igual forma también se acudió al desarrollo de los modelos con herramientas de código abierto como Python.

No obstante, es importante destacar que la información disponible solamente se refiere a los estudiantes que realizaron las pruebas Saber 11 y Saber Pro. Por lo tanto, no se cuentan con los resultados de aquellos que no ingresaron a la educación superior o que abandonaron en algún momento. Para determinar si estos factores influyen en la elección de un programa de educación superior por parte de los estudiantes, se requiere analizar el comportamiento de la educación en todo el territorio colombiano, incluyendo la evaluación de la situación social y económica de cada región, los resultados de las pruebas, el nivel de educación en otro idioma, la sectorización de las instituciones educativas y las principales áreas del conocimiento que se ofrecen, situación que se encuentra por fuera del alcance dado al presente proyecto. El análisis de la educación en un contexto social y económico es fundamental para comprender cómo se están formando las futuras generaciones y para identificar las fortalezas y debilidades que ayuden a tomar decisiones informadas en el ámbito educativo y político.

Además de considerar los factores mencionados anteriormente, es importante tener en cuenta que la evaluación de la educación es un tema complejo y multifacético que involucra diversos aspectos, desde la calidad de la enseñanza hasta la inclusión y la equidad educativa. En este sentido, es necesario utilizar un enfoque integral que permita abordar todas estas dimensiones y comprender la dinámica del sistema educativo en su conjunto.

Según algunos autores, como (Williamson, 2018), para realizar una evaluación integral de la educación es necesario considerar tanto los aspectos cuantitativos como los cualitativos. En cuanto a los aspectos cuantitativos, se deben analizar los resultados de las pruebas estandarizadas, como las que realiza el ICFES, para conocer el nivel de desempeño de los estudiantes en las diferentes áreas

del conocimiento y compararlos con los estándares nacionales e internacionales. También es importante analizar los indicadores de calidad de la educación, como la tasa de aprobación, la tasa de graduación y la tasa de deserción, para conocer la eficacia del sistema educativo y la capacidad de retener a los estudiantes.

Por tanto, para llevar a cabo una evaluación integral de la educación y comprender el panorama educativo en Colombia, es necesario considerar una amplia gama de factores y aspectos tanto cuantitativos como cualitativos. Al integrar estos aspectos en nuestro proyecto empresarial, se obtuvo una comprensión más completa de la educación en Colombia y desarrollar modelos predictivos que sean más precisos y relevantes para los estudiantes.

El modelo desarrollado pronostica el desempeño académico de los estudiantes en su educación superior por cada área del conocimiento a partir de información de rendimiento académico, variables sociales, económicas y educativas que presenta los individuos una vez presenta la prueba saber 11 y culmina sus estudios de bachillerato. Dado que la información al nivel de programas académicos es bastante densa, se presenta un resumen hasta el nivel del núcleo básico del conocimiento para contextualizar lo que un estudiante podría cursar en su vida universitaria. A continuación, se describe cada área del conocimiento:

- **Agronomía, veterinaria y afines**: En esta área del conocimiento, los núcleos básicos del conocimiento incluyen la agronomía, la medicina veterinaria y la zootecnia. La agronomía se enfoca en el estudio de los procesos agrícolas y cómo mejorar la producción de alimentos. La medicina veterinaria se enfoca en la prevención, diagnóstico y tratamiento de enfermedades animales, mientras que la zootecnia se enfoca en la cría y mejoramiento genético de animales de granja.
- **Bellas artes:** En las bellas artes, los núcleos básicos del conocimiento incluyen las artes plásticas, visuales y afines, el diseño, la música, las artes representativas, la publicidad y otros programas asociados a las bellas artes. Las artes plásticas y visuales se enfocan en la

producción de arte visual, mientras que el diseño se enfoca en la creación de productos y soluciones innovadoras. La música y las artes representativas involucran la interpretación y creación de piezas musicales y teatrales. La publicidad se enfoca en la creación de campañas publicitarias y la promoción de productos.

- **Ciencias de la educación:** En esta área del conocimiento, el núcleo básico del conocimiento es la educación, y se enfoca en la comprensión de los procesos educativos, el aprendizaje y la enseñanza. Los estudios de educación abarcan desde la educación básica hasta la educación superior, incluyendo la investigación sobre la calidad y efectividad de los programas educativos.
- **Ciencias de la salud:** En las ciencias de la salud, los núcleos básicos del conocimiento incluyen la medicina, la enfermería, la salud pública, la odontología, las terapias, la nutrición y dietética, la optometría, la bacteriología y la instrumentación quirúrgica. La medicina se enfoca en el diagnóstico y tratamiento de enfermedades en humanos, mientras que la enfermería se enfoca en la atención y cuidado de los pacientes. La salud pública se enfoca en la prevención de enfermedades y en la promoción de la salud en la comunidad. La odontología se enfoca en el cuidado y tratamiento de la salud dental. Las terapias abarcan la fisioterapia, terapia ocupacional y otras terapias para ayudar en la recuperación de pacientes. La nutrición y dietética se enfoca en la alimentación saludable y balanceada, mientras que la optometría se enfoca en el cuidado y tratamiento de la salud ocular. La bacteriología se enfoca en el estudio de bacterias y su relación con la salud humana, mientras que la instrumentación quirúrgica se enfoca en el manejo y cuidado de los instrumentos quirúrgicos utilizados en los procedimientos médicos.
- **Ciencias sociales y humanas:** En esta área del conocimiento, los núcleos básicos del conocimiento incluyen la antropología y artes liberales, la ciencia política y relaciones internacionales, el derecho y afines, la sociología, trabajo social y afines, la filosofía, teología y

afines, la geografía, historia, psicología, lenguas modernas, literatura, lingüística y afines, bibliotecología, otros

- **Economía, administración, contaduría y afines:** Se incluyen programas como Administración, Contaduría pública y Economía.
- **Ingeniería, arquitectura, urbanismo y afines:** se encuentran programas como Arquitectura, Ingeniería de sistemas, telemática y afines, Ingeniería agroindustrial, alimentos y afines, Ingeniería industrial y afines, Ingeniería civil y afines, Ingeniería electrónica, telecomunicaciones y afines, Ingeniería mecánica y afines, Ingeniería química y afines, Ingeniería eléctrica y afines, Ingeniería ambiental, sanitaria y afines, Ingeniería agrícola, forestal y afines, Ingeniería agronómica, pecuaria y afines, Ingeniería de minas, metalurgia y afines, Ingeniería administrativa y afines, e Ingeniería biomédica y afines.
- **Matemáticas y ciencias naturales:** Se incluyen programas como Biología, microbiología y afines, Matemáticas, estadística y afines, Geología y otros programas de ciencias naturales, Química y afines, y Física. Cabe destacar que estos programas se encuentran asociados al núcleo básico del conocimiento correspondiente.

4.2 Comprensión de los datos

En la segunda fase del proyecto empresarial, se realizó la gestión y compresión de datos obtenidos de los exámenes Saber 11 y Saber Pro a través de la página de datos abiertos del ICFES. Además, se obtuvieron los cruces de datos para identificar los estudiantes que presentaron Saber 11, examen de bachillerato, y que también hubiesen presentado la prueba Saber Pro, examen de educación superior.

4.2.1 Base de datos

Se analizó una población de 174.405 estudiantes que presentaron el examen Saber 11 desde 2014 y posteriormente ingresaron a la educación superior en Colombia (ver Anexo A Datos brutos

pruebas saber). Esta base de datos incluye 25 variables que abarcan características sociales, económicas, de educación y desempeño académico de las pruebas Saber 11 y Saber Pro. A continuación, se presenta el nombre de cada variable, si presenta campos nulos, la cantidad de registros y el tipo de dato.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174405 entries, 0 to 174404
Data columns (total 25 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   ESTU_TIPODOCUMENTO        174405 non-null  object
 1   ESTU_GENERO               174405 non-null  object
 2   FAMI_ESTRATOVIVIENDA      174405 non-null  object
 3   FAMI_PERSONASHOGAR        174405 non-null  object
 4   FAMI_EDUCACIONPADRE       174405 non-null  object
 5   FAMI_EDUCACIONMADRE       174405 non-null  object
 6   FAMI_TIENEINTERNET        174405 non-null  object
 7   FAMI_TIENECOMPUTADOR      174405 non-null  object
 8   FAMI_TIENEAUTOMOVIL       174405 non-null  object
 9   COLE_GENERO               174405 non-null  object
 10  COLE_NATURALEZA           174405 non-null  object
 11  COLE_CALENDARIO           174405 non-null  object
 12  COLE_BILINGUE             174405 non-null  object
 13  COLE_AREA_UBICACION       174405 non-null  object
 14  COLE_JORNADA              174405 non-null  object
 15  PUNT_LECTURA_CRITICA      174405 non-null  int64 
 16  PUNT_MATEMATICAS          174405 non-null  int64 
 17  PUNT_C_NATURALES          174405 non-null  int64 
 18  PUNT_SOCIALES_CIUDADANAS  174405 non-null  int64 
 19  PUNT_INGLES               174405 non-null  int64 
 20  PUNT_GLOBAL               174405 non-null  int64 
 21  ESTU_NSE_INDIVIDUAL       174405 non-null  object
 22  estu_area_conocimiento    174405 non-null  object
 23  area_conocimiento         174405 non-null  object
 24  puntajeGLob_pro_tyt       174405 non-null  int64 
dtypes: int64(7), object(18)
```

De la información anterior revela que contiene un extenso conjunto de datos con 174,405 filas y 25 columnas. Estas columnas abarcan una amplia variedad de información relacionada con estudiantes y su desempeño académico en la educación superior. Los datos incluyen detalles personales de los estudiantes, como tipo de documento y género, así como información sobre sus antecedentes familiares, como estrato de vivienda, nivel educativo de los padres y la disponibilidad de recursos tecnológicos en el hogar. Además, se incluye información sobre las instituciones educativas a las que asisten los estudiantes, como la naturaleza de la institución y su ubicación. Lo más destacado son las columnas que almacenan puntajes en áreas específicas de desempeño académico y un puntaje global. Estos puntajes se almacenan como valores enteros. El conjunto de

datos también contiene datos de texto, como áreas de conocimiento y tipos de calendario escolar, que se almacenan como objetos. En la tabla siguiente se presenta la descripción de cada variable.

No	Columna	Descripción	Tipo de Dato
0	ESTU_TIPODOCUMENTO	Tipo de documento del estudiante	Objeto
1	ESTU_GENERO	Género del estudiante	Objeto
2	FAMI_ESTRATOVIVIENDA	Estrato de vivienda de la familia del estudiante	Objeto
3	FAMI_PERSONASHOGAR	Número de personas en el hogar de la familia del estudiante	Objeto
4	FAMI_EDUCACIONPADRE	Nivel educativo del padre del estudiante	Objeto
5	FAMI_EDUCACIONMADRE	Nivel educativo de la madre del estudiante	Objeto
6	FAMI_TIENEINTERNET	Familia con acceso a Internet	Objeto
7	FAMI_TIENECOMPUTADOR	Familia con computador	Objeto
8	FAMI_TIENEAUTOMOVIL	Familia con automóvil	Objeto
9	COLE_GENERO	Género de la institución educativa	Objeto
10	COLE_NATURALEZA	Naturaleza de la institución educativa	Objeto
11	COLE_CALENDARIO	Calendario de la institución educativa	Objeto
12	COLE_BILINGUE	Institución bilingüe	Objeto
13	COLE_AREA_UBICACION	Área de ubicación de la institución educativa	Objeto
14	COLE_JORNADA	Jornada de la institución educativa	Objeto
15	PUNT_LECTURA_CRITICA	Puntaje en la prueba de Lectura Crítica	Entero
16	PUNT_MATEMATICAS	Puntaje en la prueba de Matemáticas	Entero
17	PUNT_C_NATURALES	Puntaje en la prueba de Ciencias Naturales	Entero
18	PUNT_SOCIALES_CIUDADANAS	Puntaje en la prueba de Ciencias Sociales y Ciudadanas	Entero
19	PUNT_INGLES	Puntaje en la prueba de Inglés	Entero
20	PUNT_GLOBAL	Puntaje global del estudiante	Entero
21	ESTU_NSE_INDIVIDUAL	Nivel socioeconómico individual del estudiante	Objeto
22	estu_area_conocimiento	Área de conocimiento del estudiante	Objeto
24	puntajeGLob_pro_tyt	Puntaje global promedio de las pruebas Saber Pro	Entero

***Tabla 1**. Diccionario de datos*

Fuente: Elaboración propia.

4.2.2 Estadísticas descriptivas de variables numéricas

A continuación, se presenta un resumen de las estadísticas descriptivas de las variables numéricas de la base de datos.

Estadística	PUNT_LECTURA_CRITICA	PUNT_MATEMATICAS	PUNT_C_NATURALES	PUNT_SOCIALES_CIUDADANAS	PUNT_INGLES	PUNT_GLOBAL	puntajeGLob_pro_tyt
Count (*)	174	174	174	174	174	174	174
Media	58.07	59.31	58.68	58.45	59.50	293.48	156.12
Desviación Estándar	9.32	12.01	10.37	10.03	14.17	46.69	25.04
Mínimo	0	17	8	11	0	92	0
Percentil 25%	52	51	52	52	49	261	139
Percentil 50% (Mediana)	58	58	58	59	55	290	156
Percentil 75%	64	67	66	65	68	323	173
Máximo	100	100	100	100	100	492	269

** Datos dados en unidades de miles*

Tabla 2*. Estadísticas descriptivas de variables numéricas*
Fuente: Elaboración propia.

En primer lugar, se observa que el puntaje promedio en la prueba de Lectura Crítica es de 58.07, con una desviación estándar de 9.32, indicando que la mayoría de los estudiantes obtienen puntajes cercanos a la media. La prueba de Matemáticas presenta un promedio de 59.31 y una desviación estándar más alta de 12.01, lo que sugiere una mayor variabilidad en los puntajes. Por otro lado, en la prueba de Ciencias Naturales, el puntaje promedio es de 58.68, con una desviación estándar razonable de 10.37. Similarmente, en la prueba de Ciencias Sociales y Ciudadanas, el puntaje promedio es de 58.45, con una desviación estándar de 10.03. Finalmente, en la prueba de Inglés, el puntaje promedio es de 59.50, pero con una desviación estándar más alta de 14.17, lo que indica una variabilidad considerable en los puntajes de inglés.

Además de las pruebas individuales, se calcula un puntaje global promedio de 293.48, derivado de las pruebas anteriores, con una desviación estándar de 46.69. Este puntaje refleja el rendimiento general de los estudiantes en todas las áreas evaluadas. Por último, se presenta una variable llamada "puntajeGLob_pro_tyt," con un puntaje promedio de 156.12 y una desviación estándar de 25.04.

Esta variable es el resultado de las pruebas Saber Pro y exhibe una variabilidad razonable en los puntajes.

4.2.3 Distribución de las variables numéricas

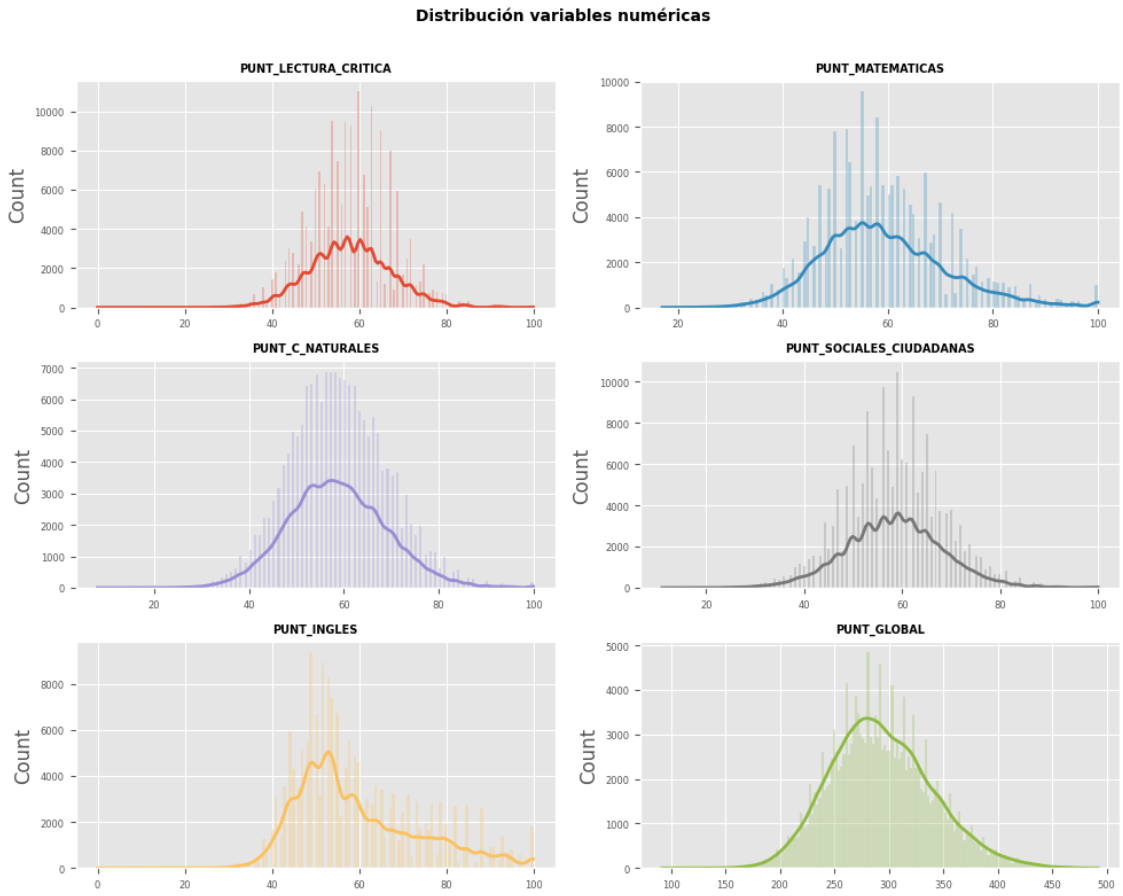


Figura 3. *Distribución de las variables numéricas*

Fuente: Elaboración propia.

Tras analizar las distribuciones de las variables numéricas en la figura anterior, se revela una tendencia predominante hacia una distribución normal en la mayoría de las variables. Esta distribución normal se caracteriza por una agrupación de los datos alrededor de un valor central con una forma simétrica de campana, indicando que la mayor parte de los puntajes se concentran cerca de la media y disminuyen a medida que nos alejamos de este valor central. Este patrón es coherente

con la expectativa de que los puntajes de los estudiantes sigan una distribución normal en ausencia de sesgos o factores externos significativos.

Sin embargo, al examinar específicamente el puntaje de la prueba de inglés, se observa una distribución asimétrica hacia la derecha. Esta particularidad señala una concentración mayor de puntajes más bajos, con algunos valores más altos que se extienden hacia la derecha de la distribución. La presencia de esta asimetría sugiere la influencia de factores adicionales que podrían estar afectando los resultados de la prueba de inglés, como la dificultad del examen, la preparación específica en ese idioma por parte de los estudiantes y la instrucción o enfoque lingüístico de los colegios a los cuales pertenecen.

Por lo tanto, aunque la mayoría de las variables numéricas exhiben una distribución normal, el puntaje en la prueba de inglés presenta una excepción notable. Este hallazgo subraya la importancia de realizar un análisis minucioso de cada variable y sus respectivas distribuciones para comprender la complejidad de los datos y orientar adecuadamente las decisiones educativas y evaluativas. Asimismo, sugiere la necesidad de investigar a fondo las posibles causas detrás de esta distribución asimétrica y considerar estrategias específicas para abordar las deficiencias identificadas en el dominio del inglés.

4.2.4 Correlación entre variables numéricas y la variable respuesta

El propósito de los gráficos de correlación realizados con la variable "puntajeGLob_pro_tyt", que es la variable respuesta de un modelo analítico de predicción, es evaluar la relación entre esta variable y otras variables numéricas del conjunto de datos. Estos gráficos de dispersión muestran visualmente si existe una correlación lineal entre "puntajeGLob_pro_tyt" y las demás variables, lo que es esencial para comprender cómo influyen las variables predictoras en la variable de interés. Al analizar estos gráficos, es posible identificar posibles patrones y determinar la fuerza y dirección de la relación entre las variables, lo que facilita la construcción y evaluación de modelos predictivos. En los gráficos se evidencia una correlación positiva entre las variables de desempeño

académico de la prueba saber 11 con la variable respuesta que es el puntaje global de la prueba saber pro.

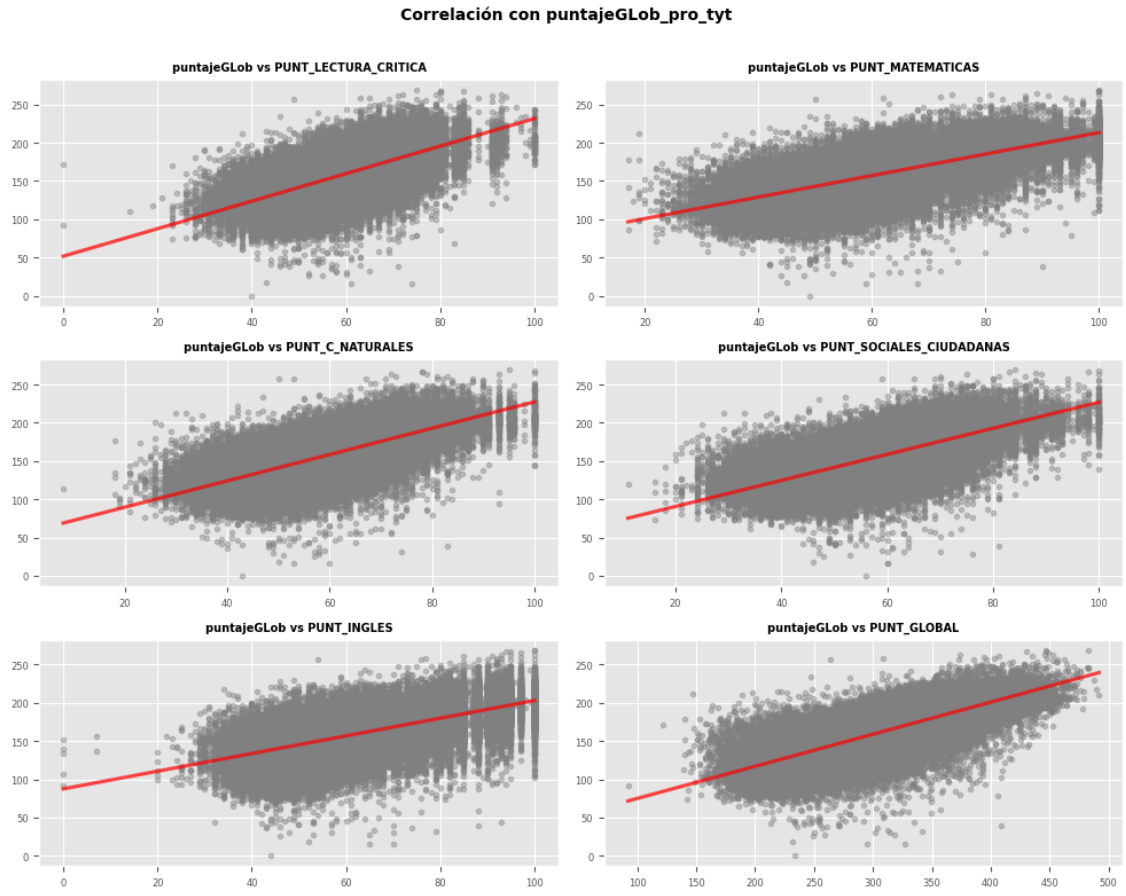


Figura 4. *Correlación entre variables numéricas y la variable respuesta*
Fuente: Elaboración propia.

4.2.5 Distribución de la variable respuesta

Cuando se desarrolla un modelo, resulta esencial analizar detenidamente la distribución de la variable de respuesta, ya que esta variable es fundamental para las predicciones. En el caso de la variable "puntajeGLob_pro_tyt," se observa que presenta una distribución simétrica. Esta simetría es un aspecto importante en estadística, ya que sugiere que los datos están equilibrados y siguen un patrón predecible. Sin embargo, en ocasiones, la aplicación de transformaciones puede ayudar a mejorar la visualización de la distribución.

Para explorar y visualizar estas transformaciones, se han creado gráficos que muestran la distribución original de la variable "puntajeGLob_pro_tyt" junto con las distribuciones resultantes después de aplicar dos transformaciones, la raíz cuadrada y el logaritmo. Estas transformaciones permiten explorar si los datos pueden ajustarse mejor a un modelo estadístico, especialmente si se busca que se asemejen más a una distribución normal. Estas visualizaciones proporcionan información útil para determinar si las transformaciones son adecuadas y cómo afectan la distribución de la variable.

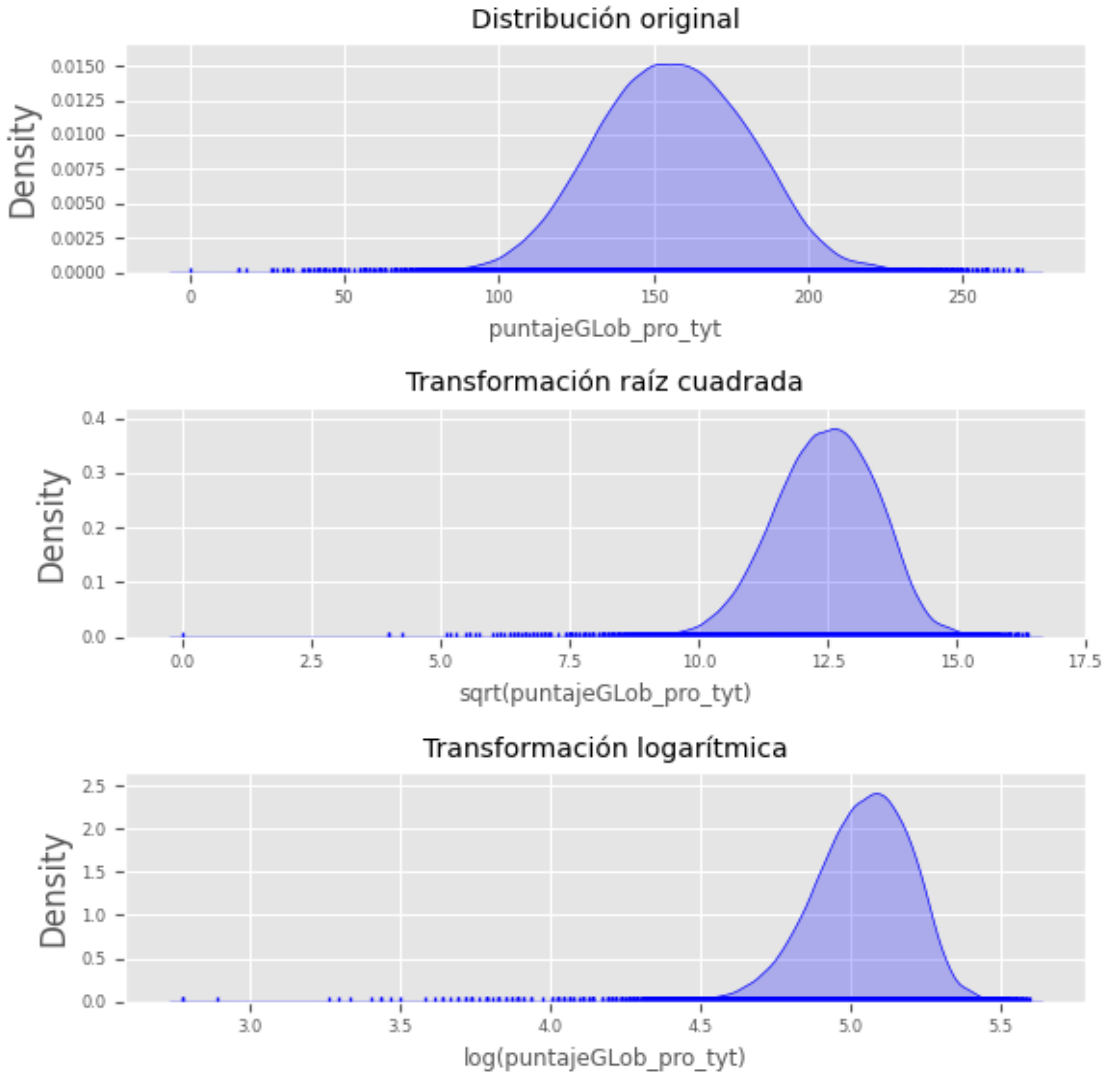


Figura 5. *Distribución de la variable respuesta*

Fuente: Elaboración propia.

Algunos algoritmos de machine learning y técnicas de aprendizaje estadístico requieren que la variable de interés siga una distribución específica. Por ejemplo, en el caso de los modelos de regresión lineal, es necesario que la variable de respuesta se distribuya de manera normal, mientras que, para los modelos lineales generalizados, se espera que siga una distribución de la familia exponencial. A continuación, se presenta una evaluación de diversos ajustes de distribuciones a la variable "puntajeGLob_pro_tyt" con respecto a varias métricas de calidad.

Distribución	sumsquare_error	aic	bic	kl_div	ks_statistic	ks_pvalue
gamma	0,00013	1.875	1.905	inf	0,01274	5,33E-25
norm	0,00013	1.868	1.888	inf	0,01301	4,54E-26
beta	0,00013	1.867	1.907	inf	0,01335	2,04E-27
chi2	0,00015	inf	inf	inf	0,02754	2,65E-115
logistic	0,00017	1.533	1.553	inf	0,02680	3,26E-109
cauchy	0,00049	1.319	1.339	inf	0,07681	0,00E+00
powerlaw	0,00280	1.166	1.196	inf	0,38455	0,00E+00
expon	0,00367	1.186	1.207	inf	0,47716	0,00E+00
exponpow	0,00541	1.657	1.688	inf	0,93089	0,00E+00

Tabla 3. Evaluación de la variable respuesta con diversas distribuciones

Fuente: Elaboración propia.

Estas métricas son indicadores de cuán bien se ajusta cada distribución a los datos observados. La prueba KS (Kolmogorov-Smirnov) se utiliza para determinar si una muestra de datos sigue una distribución específica. En este caso, se proporcionan valores de suma de cuadrados del error, criterios de información (AIC y BIC), divergencia de Kullback-Leibler (KL), estadística KS y valor p para varias distribuciones (gamma, normal, beta, chi-cuadrado, logística, Cauchy, ley de potencias, exponencial y exponencial con potencia), analizando todos los valores de las estadísticas KS son muy bajos, lo que indica que no hay una discrepancia significativa entre la distribución empírica de los datos y las distribuciones teóricas propuestas.

4.2.6 Análisis de correlación entre variable numéricas

La correlación es una medida que permite comprender cómo dos variables están relacionadas entre sí, lo que es crucial en estadística y análisis de datos. La siguiente figura, muestra la matriz de correlación de Pearson, que mide la fuerza y dirección de las relaciones lineales entre las variables. En ella se identifica las correlaciones más fuertes y destacadas es tonalidades oscuras.

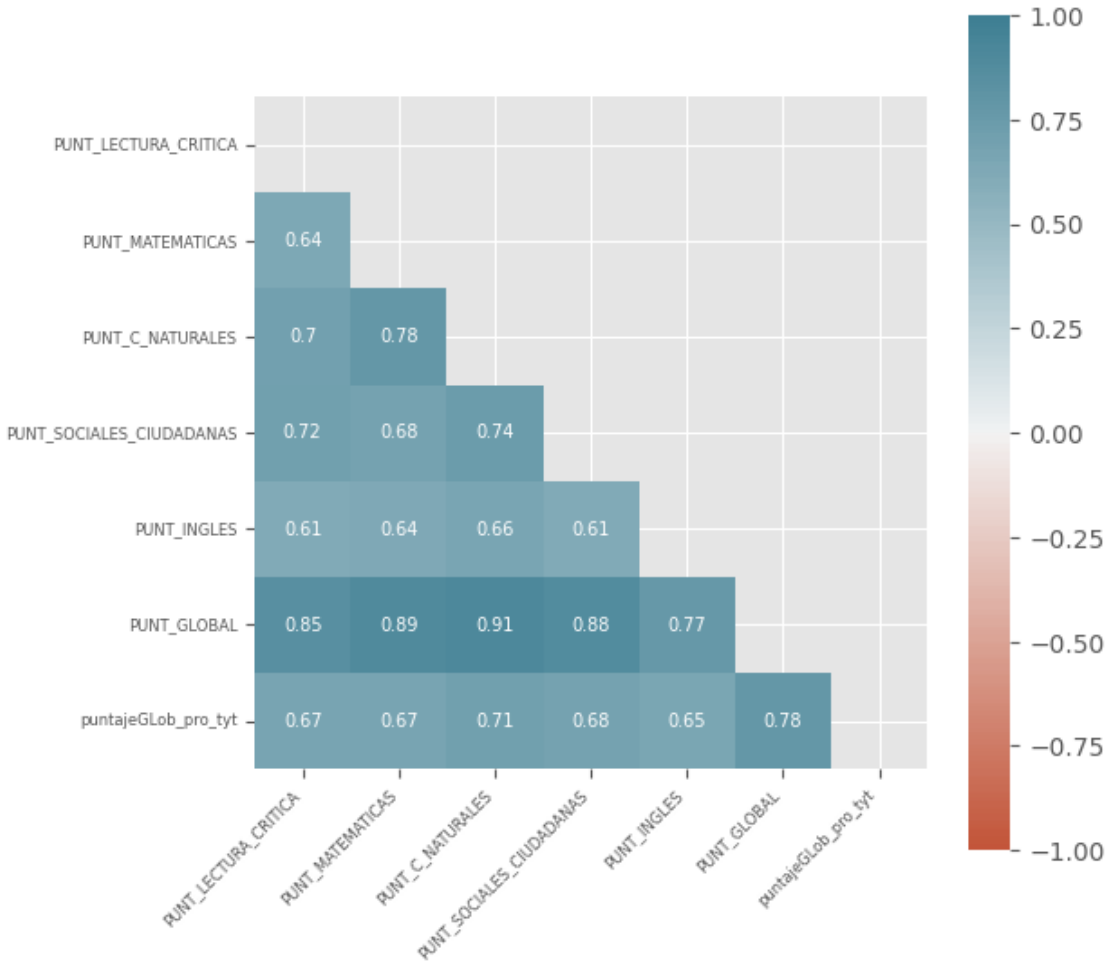


Figura 6. *Análisis de correlación entre variable numéricas*

Fuente: Elaboración propia.

Uno de los hallazgos más destacados es la fuerte correlación positiva entre el puntaje global (PUNT_GLOBAL) y el puntaje en Ciencias Naturales (PUNT_C_NATURALES) con un valor de 0.909854. Esto indica que un alto rendimiento en Ciencias Naturales tiende a estar asociado con un

rendimiento global sobresaliente. Además, el puntaje global también muestra una alta correlación con el puntaje en Matemáticas (PUNT_MATEMATICAS) y en Sociales y Ciudadanía (PUNT_SOCIALES_CIUDADANAS), destacando la influencia significativa de estas áreas en el rendimiento general.

La relación entre la capacidad de lectura crítica (PUNT_LECTURA_CRITICA) y el rendimiento global (PUNT_GLOBAL) también es notable, con una correlación de 0.849747. Esto sugiere que un buen desempeño en lectura crítica a menudo se traduce en un buen rendimiento global.

Por otro lado, la correlación más baja entre estas variables se encuentra entre PUNT_C_NATURALES y PUNT_MATEMATICAS, aunque sigue siendo positiva con un valor de 0.782410. Esto indica que estas dos áreas están relacionadas, pero su vínculo es menos pronunciado en comparación con otras correlaciones.

Este análisis proporciona una visión más clara de cómo las diversas áreas de conocimiento están interconectadas y cómo pueden impactar en el rendimiento global de los estudiantes. Estos descubrimientos son de suma importancia para la toma de decisiones y para comprender los factores que influyen en el éxito académico en el contexto de un modelo analítico de predicción. Como resultado, se ha tomado la decisión de excluir la variable de puntaje global para la construcción de los modelos analíticos de predicción, con el objetivo de reducir posibles errores en las predicciones debido a que esta variable condensa el resultado de las demás variables numéricas.

4.2.7 Reducción de variables numéricas

El resultado de la correlación de variables después de retirar la variable "puntaje global" revela relaciones significativas entre varias áreas de conocimiento. Por ejemplo, se observa una correlación entre los puntajes de Matemáticas y Ciencias Naturales (0.782), lo que sugiere que los estudiantes que obtienen altas calificaciones en una de estas áreas tienden a hacerlo también en la

otra. Lo mismo ocurre con las puntuaciones de Ciencias Sociales y Ciudadanía y Lectura Crítica, que muestran una correlación de 0.739 y 0.717, respectivamente.

También es interesante destacar que, incluso después de retirar el "puntaje global," aún se encuentra una correlación importante (0.712) entre el puntaje global de la prueba saber pro y la variable de Ciencias Naturales. Esto podría indicar que el rendimiento en las pruebas universitarias está influenciado por los resultados de Ciencias Naturales.

La figura que a continuación se presenta, sugiere que las áreas de Matemáticas, Ciencias Naturales y Ciencias Sociales y Ciudadanía están intrínsecamente relacionadas en términos de rendimiento, y que el puntaje global en la prueba saber pro mantiene una correlación significativa con Ciencias Naturales.

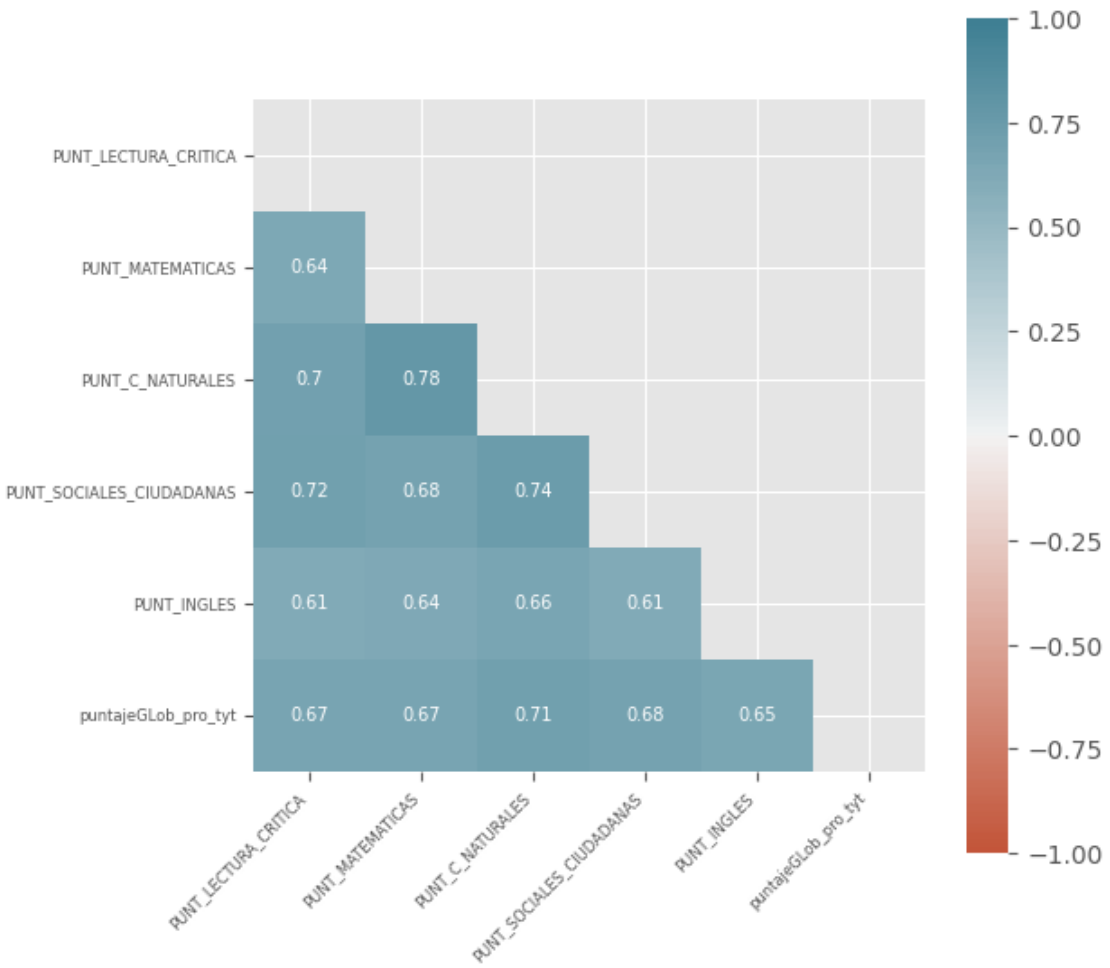


Figura 7. *Reducción de variables numéricas*

Fuente: Elaboración propia.

4.2.8 Estadísticas de variables cualitativas

A continuación, se presenta un resumen de las estadísticas descriptivas de las variables cualitativas de la base de datos.

Atributo	Cantidad de valores únicos	Valor más común	Frecuencia del valor más común	Porcentaje del valor más común
ESTU_TIPODOCUMENTO	8	TI	165	95%
ESTU_GENERO	2	F	109	63%
FAMI_ESTRATOVIVIENDA	6	Estrato 2	59	34%
FAMI_PERSONASHOGAR	5	3 a 4	99	57%

Atributo	Cantidad de valores únicos	Valor más común	Frecuencia del valor más común	Porcentaje del valor más común
FAMI_EDUCACIONPADRE	2	NO	101	58%
FAMI_EDUCACIONMADRE	2	NO	93	53%
FAMI_TIENEINTERNET	2	SI	133	76%
FAMI_TIENECOMPUTADOR	2	SI	146	84%
FAMI_TIENEAUTOMOVIL	2	NO	104	60%
COLE_GENERO	3	MIXTO	154	89%
COLE_NATURALEZA	2	OFICIAL	99	57%
COLE_CALENDARIO	2	A	165	95%
COLE_BILINGUE	2	NO	169	97%
COLE_AREA_UBICACION	2	URBANO	163	94%
COLE_JORNADA	4	UNICA	151	87%
ESTU_NSE_INDIVIDUAL	5	NSE3	80	46%
estu_area_conocimiento	10	CIENCIAS SOCIALES Y HUMANAS (CSH)	47	27%

Tabla 4. *Estadísticas de variables cualitativas*

Fuente: Elaboración propia.

El análisis de las variables cualitativas en el conjunto de datos revela información valiosa sobre las características de los estudiantes y las instituciones educativas en estudio. Entre los hallazgos más destacados, se observa que la mayoría de los estudiantes tienen como tipo de documento la Tarjeta de Identidad (TI) y son de género femenino. Además, la mayor proporción de estudiantes proviene de hogares ubicados en el Estrato 2, con un número de personas en el hogar de 3 a 4 y cuyos padres no tienen educación formal.

En cuanto a las características de las instituciones educativas, se destaca que la mayoría son de naturaleza oficial y operan bajo el calendario académico A. Además, la gran mayoría no son instituciones bilingües y están ubicadas en áreas urbanas. Las jornadas educativas varían, siendo "ÚNICA" la más frecuente.

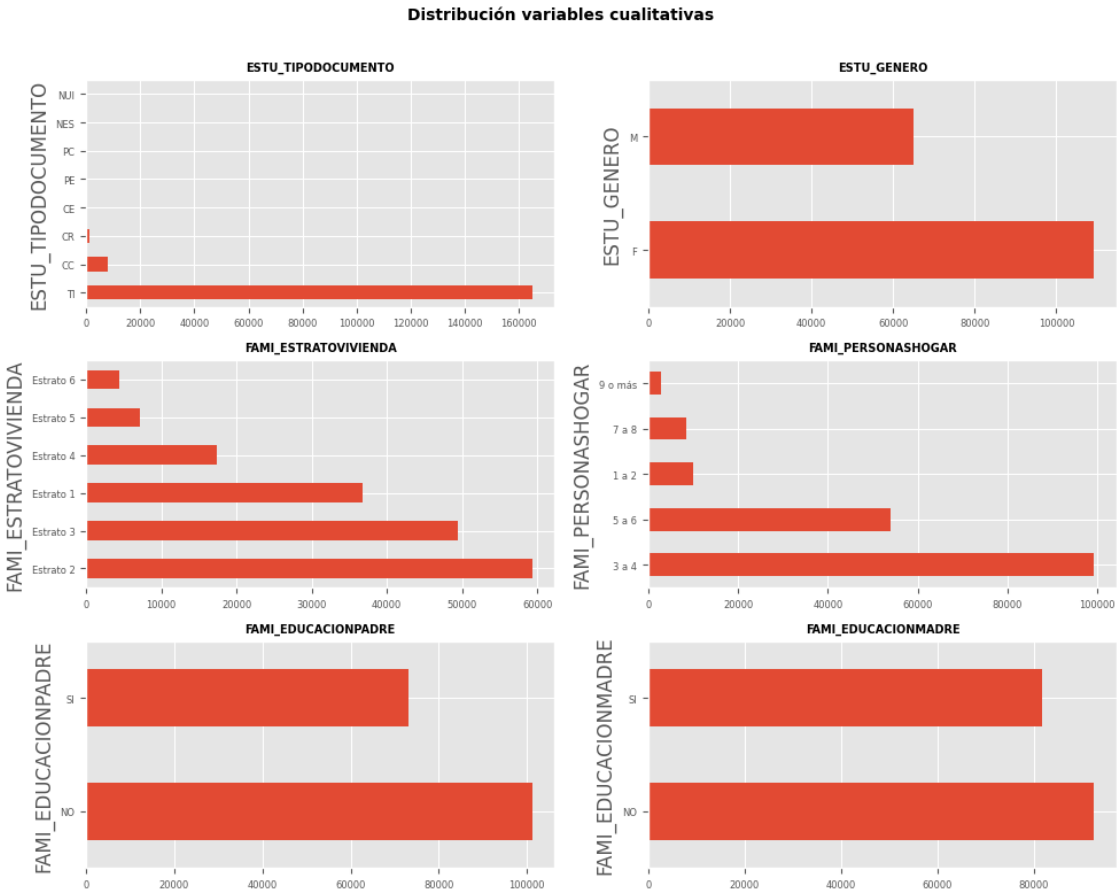
Por último, se evaluó el nivel socioeconómico de los estudiantes, y el nivel socioeconómico 3 (NSE3) es el más común entre los aspirantes a la educación superior. Además, se identificaron diez

áreas de conocimiento, siendo "CIENCIAS SOCIALES Y HUMANAS (CSH)" la más frecuente entre los estudiantes.

4.2.9 Distribución de variables cualitativas

A continuación, se visualiza la distribución de las variables cualitativas mediante gráficos de barras horizontales, lo que es crucial para comprender la composición y la prevalencia de diferentes categorías dentro de las variables.

Cada gráfico representa una variable cualitativa específica y muestra cuántas veces aparece cada categoría en los datos. Esta información es valiosa para descubrir patrones y desequilibrios que pueden tener un impacto significativo en la investigación. Los gráficos proporcionan una visión rápida y clara de las distribuciones de categorías, lo que facilita la identificación de tendencias y la toma de decisiones informadas en el análisis de datos.



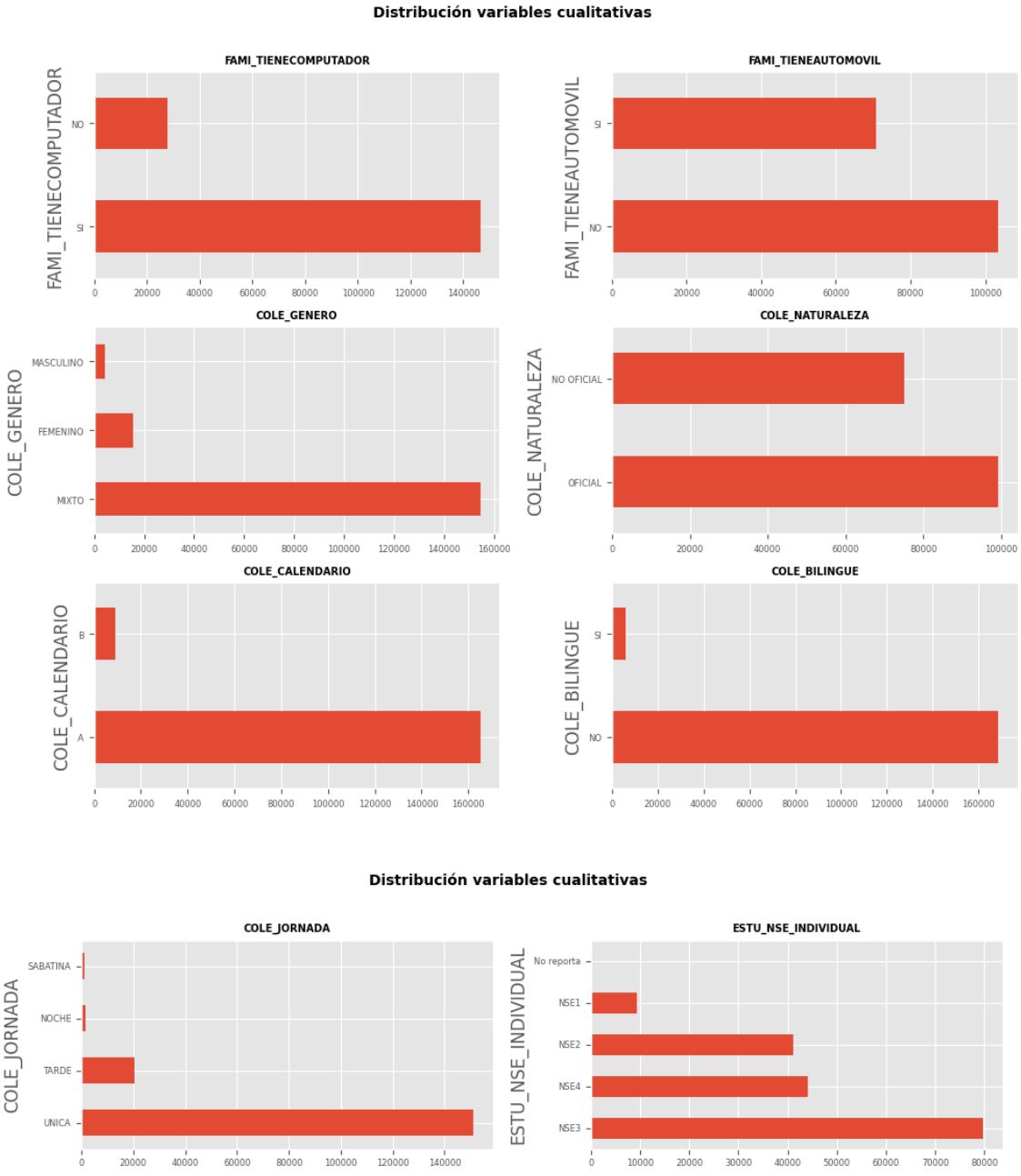


Figura 8. *Distribución de variables cualitativas*

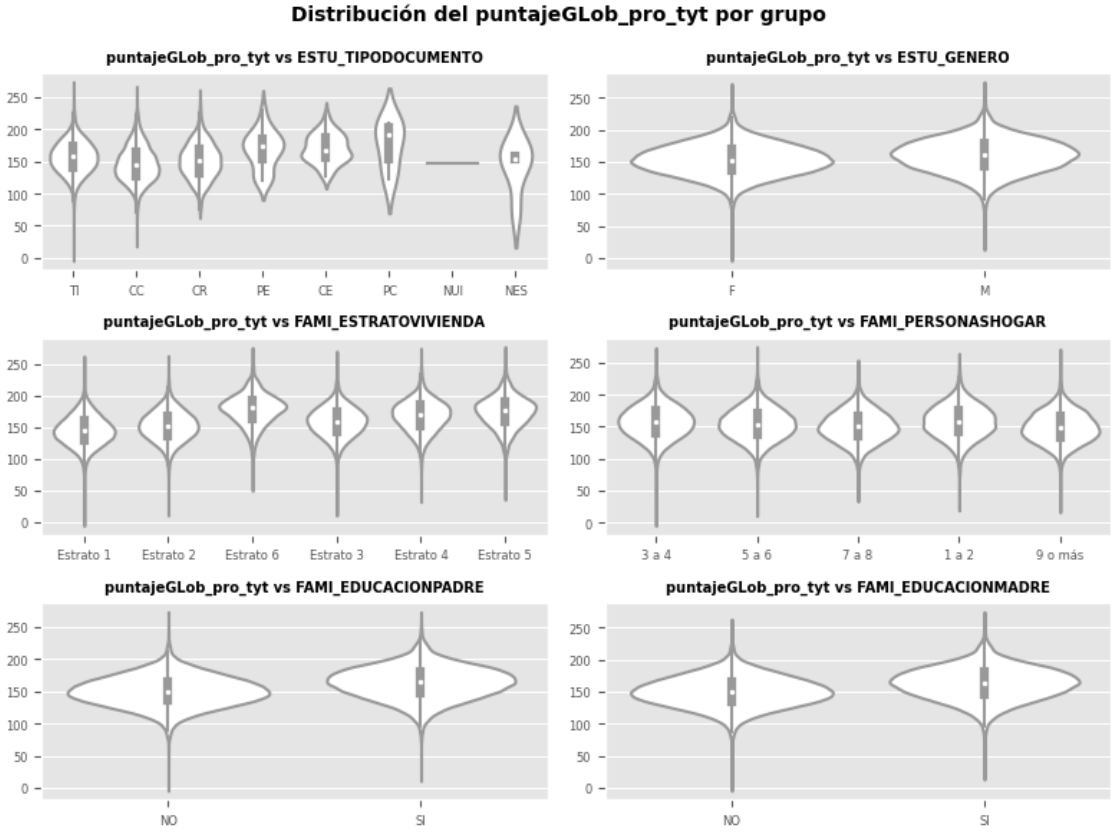
Fuente: Elaboración propia.

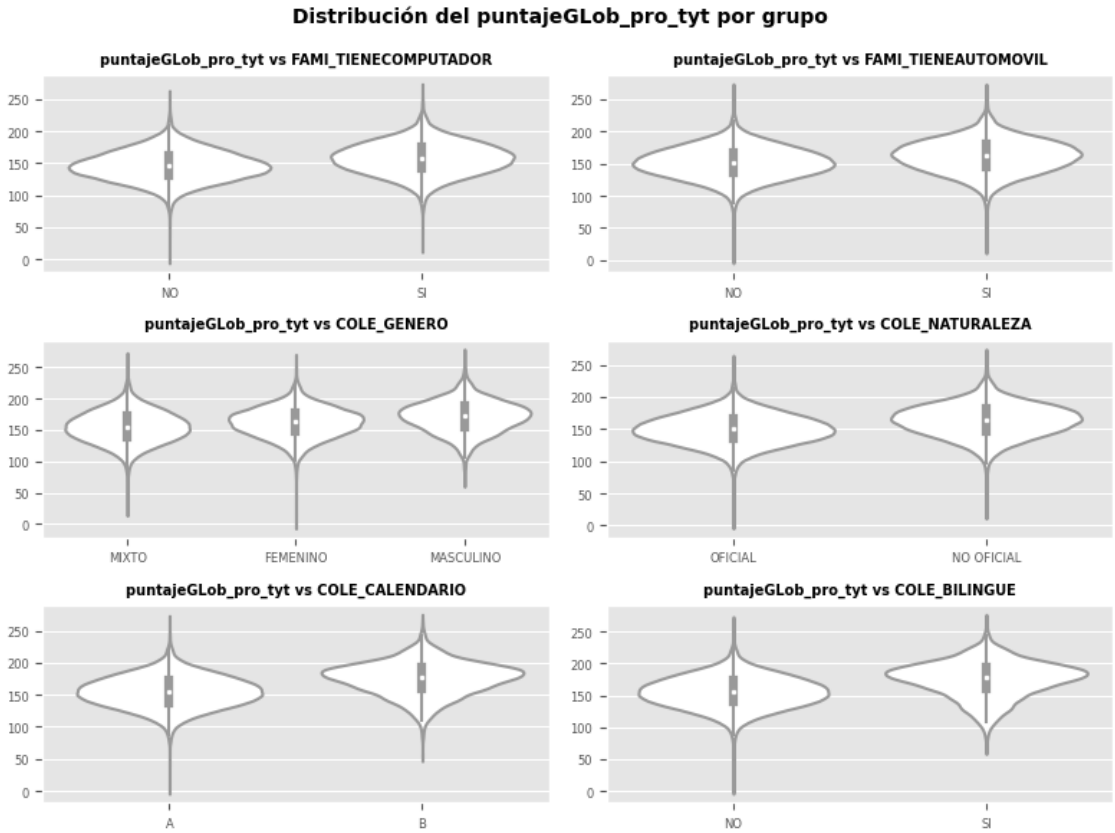
4.2.10 Correlación entre variables cualitativas y la variable respuesta

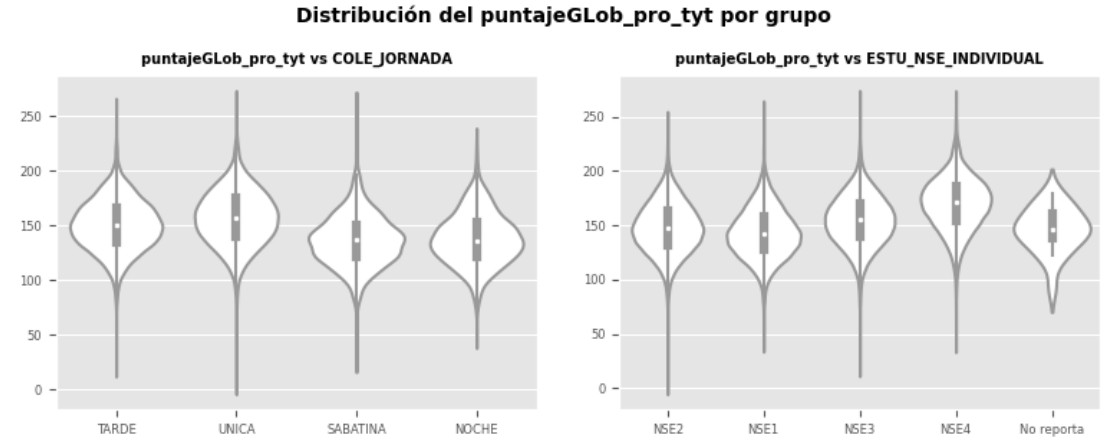
En este apartado se explorar la relación entre la variable "puntajeGLob_pro_tyt," que es una variable numérica representativa del puntaje global de las pruebas saber pro, y las variables

cualitativas del conjunto de datos. Esto se hace a través de la generación de gráficos de violín que muestran cómo se distribuyen los puntajes (en este caso, "puntajeGLob_pro_tyt") en función de diferentes categorías o grupos definidos por las variables cualitativas.

Cada gráfico muestra una relación visual entre el puntaje y una variable cualitativa particular.




Distribución del puntajeGLob_pro_tyt por grupo
puntajeGLob_pro_tyt vs FAMI_TIENECOMPUTADOR
NO
SI
puntajeGLob_pro_tyt vs FAMI_TIENEAUTOMOVIL
NO
SI
puntajeGLob_pro_tyt vs COLE_GENERO
MIXTO
FEMENINO
MASCULINO
puntajeGLob_pro_tyt vs COLE_NATURALEZA
OFICIAL
NO OFICIAL
puntajeGLob_pro_tyt vs COLE_CALENDARIO
A
B
puntajeGLob_pro_tyt vs COLE_BILINGUE
NO
SI


Distribución del puntajeGLob_pro_tyt por grupo
puntajeGLob_pro_tyt vs COLE_JORNADA
TARDE
UNICA
SABATINA
NOCHE
puntajeGLob_pro_tyt vs ESTU_NSE_INDIVIDUAL
NSE2
NSE1
NSE3
NSE4
No reporta

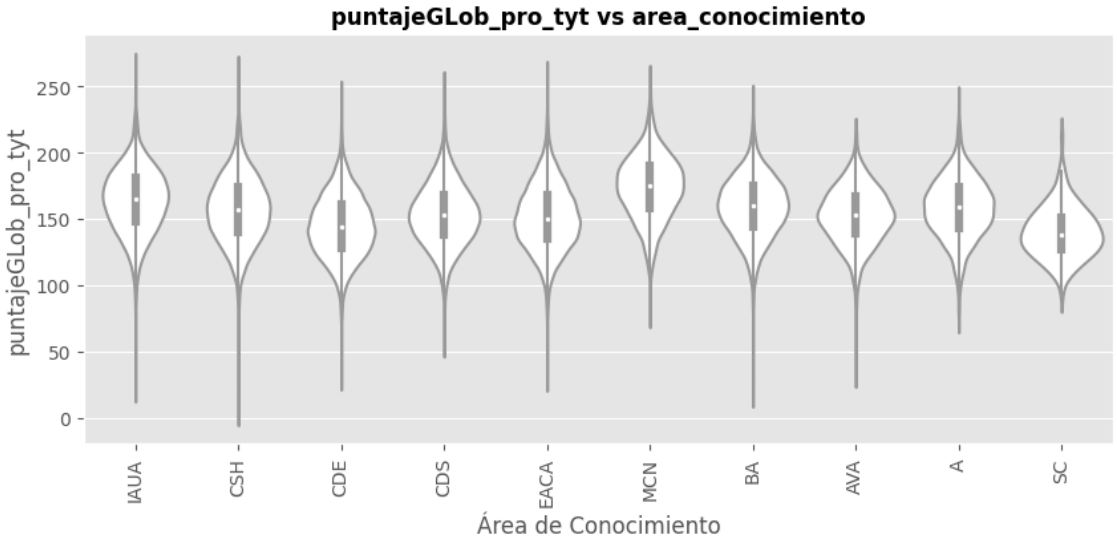


Figura 9. *Correlación entre variables cualitativas y la variable respuesta*
Fuente: Elaboración propia.

Las anteriores gráficas revelan algunas tendencias significativas. En primer lugar, los estudiantes de estratos socioeconómicos 6, 5 y 4 obtienen puntajes más altos en las pruebas Saber Pro, sugiriendo una correlación positiva entre el nivel socioeconómico y el rendimiento académico. Además, se observa que los puntajes tienden a ser mejores en hogares con 3 o 4 miembros.

En cuanto a la influencia de la educación de los padres, se nota que la educación formal tanto del padre como de la madre tiene un impacto positivo en los puntajes de los estudiantes.

Asimismo, factores económicos, como la posesión de un computador o automóvil, se correlacionan con puntajes más altos.

En relación con las características de las instituciones educativas, se observa que los estudiantes que asisten a colegios privados, de calendario B, de género masculino, bilingües y con jornada única tienden a obtener puntajes más altos en las pruebas.

Finalmente, se destaca que las áreas del conocimiento que obtienen los puntajes más altos corresponden a Matemáticas y Ciencias Naturales (MCN), lo que subraya la relevancia de estas materias en la evaluación del rendimiento académico.

4.2.11 Análisis de correlación entre variables cualitativas

En el análisis de correlaciones de variables cualitativas, se comenzó identificando aquellas con características booleanas, es decir, aquellas que podían tomar únicamente dos valores. Por ejemplo, se consideró si el colegio donde el estudiante cursó la educación media era bilingüe, asignándole cero para sí y uno para no. Respecto a las variables que reflejaban un orden en sus opciones de respuesta, como el nivel socioeconómico medido a través del estrato, se asignó cero para estrato uno, uno para estrato dos, dos para estrato tres, y así sucesivamente. Esta asignación numérica a variables cualitativas permitió desarrollar un análisis de correlaciones.

El gráfico de matriz de correlaciones de las variables cualitativas que a continuación se presenta, muestra interesantes relaciones. En primer lugar, se observa una alta correlación entre la educación formal del padre y la madre. Esto sugiere que el nivel educativo de los padres tiende a estar relacionado, lo que es coherente con la influencia de un entorno educativo compartido en el hogar.

Además, se nota una fuerte correlación entre la disponibilidad de un computador y acceso a Internet en el hogar. Esto sugiere que estas dos variables están estrechamente relacionadas y podrían interpretarse como indicadores del acceso a herramientas tecnológicas en el hogar.

Otra correlación destacada se encuentra entre el estrato de la vivienda y la naturaleza del colegio. Esto sugiere que las personas que pertenecen a estratos socioeconómicos más altos tienden a inscribir a sus hijos en colegios privados, que a menudo siguen un calendario académico tipo B. Esto se alinea con la idea de que el nivel socioeconómico influye en la elección de la institución educativa.

Por último, se observa una correlación entre el nivel socioeconómico NSE y el estrato de la vivienda. Esto es comprensible ya que una parte del cálculo de la variable NSE depende del estrato. Estas correlaciones arrojan luz sobre las relaciones subyacentes entre las variables cualitativas.

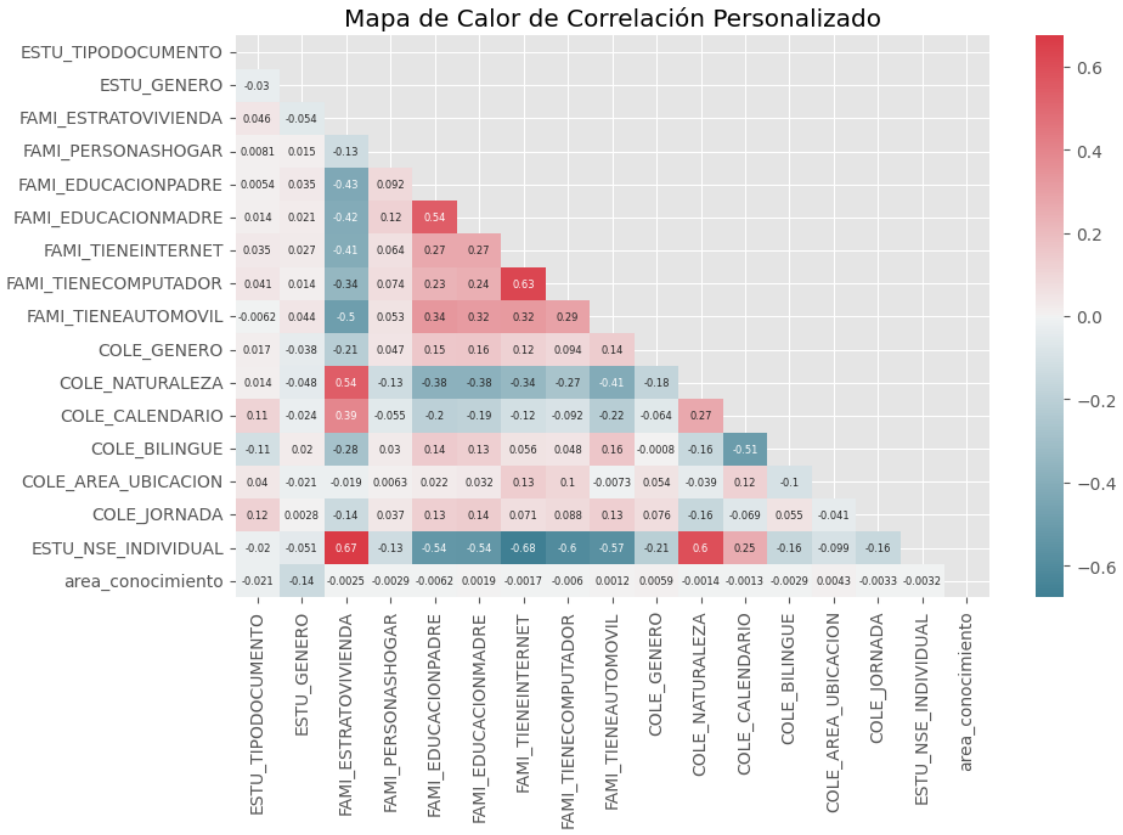


Figura 10. *Análisis de correlación entre variables cualitativas*

Fuente: Elaboración propia.

4.2.12 Reducción de variables cualitativas

Tras un análisis y con el objetivo de mitigar posibles desviaciones en los modelos, se ha decidido eliminar ciertas variables debido a su alta correlación. Las variables eliminadas son 'FAMI_EDUCACIONPADRE', 'ESTU_NSE_INDIVIDUAL', 'FAMI_TIENECOMPUTADOR', 'COLE_NATURALEZA' y 'COLE_CALENDARIO'. Respecto a la variable 'COLE_GENERO', se ha excluido del modelo dado que ya se considera la variable 'ESTU_GENERO', evitando así posibles errores al tener dos variables que tratan sobre el género de los estudiantes. Asimismo, se optó por retirar la variable 'COLE_JORNADA' debido a la falta de claridad en sus atributos; aunque inicialmente se pensó en agrupar los estudiantes de jornada matutina en una única categoría, esta decisión carecía de fundamentos adicionales, por lo que se reafirma su eliminación. Estas

modificaciones buscan reducir la multicolinealidad y mejorar la precisión de las predicciones. Para evaluar el impacto de estas eliminaciones, se ha generado un gráfico de correlación entre las variables cualitativas restantes sin las mencionadas, evidenciando una disminución en la correlación entre estas. Este proceso marca un avance significativo en la preparación de datos y el modelado.

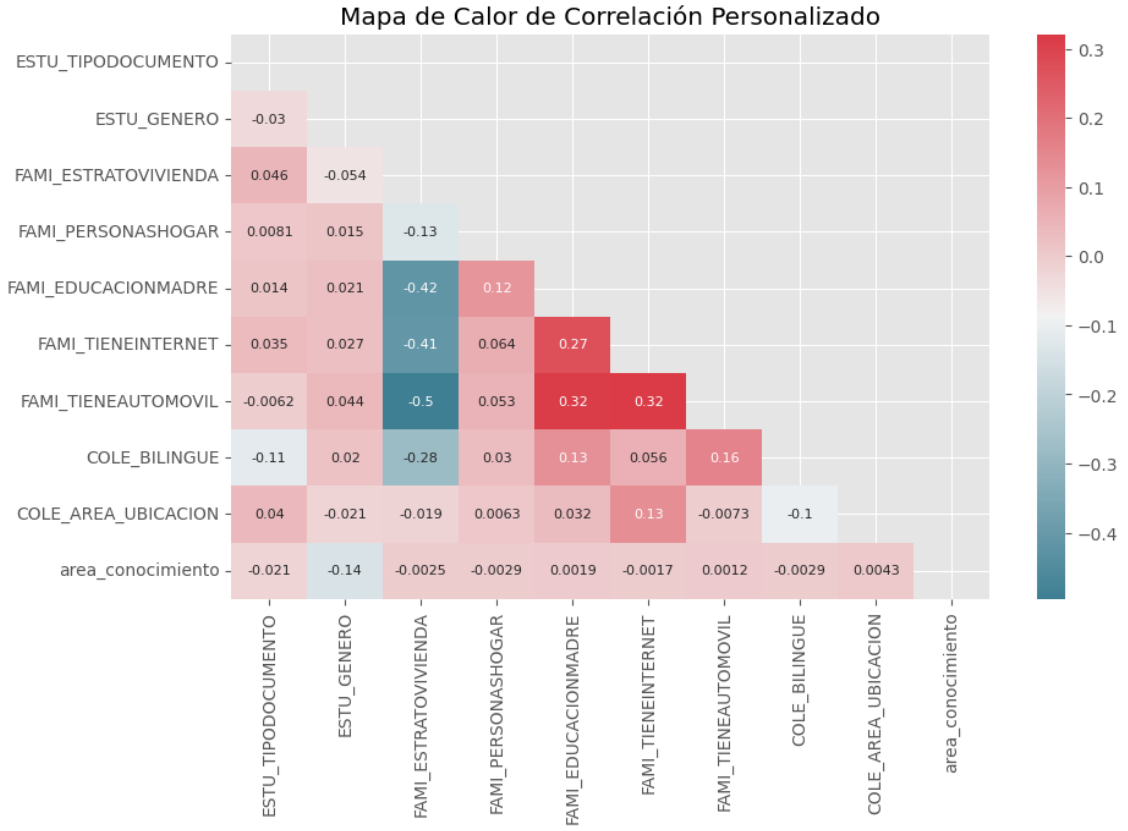


Figura 11. *Reducción de variables cualitativas*

Fuente: Elaboración propia.

4.3 Preparación de los datos

En el proceso de preparación de los datos para la metodología CRISP, se encontró que las carreras presentes en los exámenes de Saber Pro son muy diversas. Por esta razón, se investigaron las áreas del conocimiento a las que están clasificadas cada una de las carreras en la plataforma del SNIES (Sistema Nacional de Información de Educación Superior), y se redujeron a nueve para facilitar el análisis. Además, se asignó una abreviatura a cada área para identificarlas de manera más

sencilla de la siguiente forma; Agronomía, Veterinaria y Afines – AVA. Bellas Artes – BA. Ciencias de la Educación – CDE. Ciencias de la Salud – CDS. Ciencias Sociales y Humanas – CSH. Economía, Administración, Contaduría y Afines – EACA. Ingeniería, Arquitectura, Urbanismo y Afines – IAUA. Matemáticas y Ciencias Naturales – MCN. Arquitectura – A, Sin Clasificar – SC.

Es importante mencionar que la recodificación o redefinición de variables es una técnica común en la construcción de modelos analíticos. La recodificación consiste en transformar los datos originales en una nueva representación que puede ser más adecuada para el análisis. Esta técnica se utiliza para convertir variables categóricas en numéricas, para agrupar valores, para reducir la dimensionalidad de los datos, entre otros propósitos. La recodificación puede ayudar a mejorar la calidad de los datos y hacer que sean más fáciles de interpretar y analizar (Han, 2006).

Dado lo anterior, se inició el proceso de preparación de los datos, realizando la selección de variables relevantes y se cargó la información en Python. Antes de cargar el archivo de Excel, se realizaron transformaciones de las variables por cada área del conocimiento. En primer lugar, se asignó el valor 0 a la variable "M" y el valor 1 a la variable "F" para la variable de género. Luego, se asignaron valores a las variables categóricas, como "tiene internet", que se definió con un valor de 0 para las variables "NO" y un valor de 1 para las variables "SI". En cuanto a la variable "personas hogar", se asignó un valor de 1 a 5 según correspondiera a la categoría. Para la educación del padre y de la madre, se codificó con 1 si tenían educación superior y 0 si no la tenían. Asimismo, se codificó la variable categórica "tiene automóvil", siendo 1 el valor de "SI" y 0 el valor de "NO". En cuanto al género del colegio, se estableció 1 para femenino, 2 para masculino y 3 para mixto. De igual forma, se codificó la naturaleza del colegio como 0 para "No oficial" y 1 para "Oficial", el área de ubicación del colegio como 0 para "rural" y 1 para "urbano", y la jornada del colegio como 1 para "completa", 2 para "mañana", 3 para "noche", 4 para "sabatina", 5 para "tarde" y 6 para "única". A continuación, se visualiza el resultado de la limpieza y reducción de datos.

```
1.  # Tipo de cada columna
```

```
# ================================================================================
# En pandas, el tipo "object" hace referencia a strings
# datos.dtypes
datos.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 174405 entries, 0 to 174404
Data columns (total 23 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   ESTU_TIPODOCUMENTO        174405 non-null  object
 1   ESTU_GENERO               174405 non-null  object
 2   FAMI_ESTRATOVIVIENDA      174405 non-null  object
 3   FAMI_PERSONASHOGAR        174405 non-null  object
 4   FAMI_EDUCACIONPADRE       174405 non-null  object
 5   FAMI_EDUCACIONMADRE       174405 non-null  object
 6   FAMI_TIENEINTERNET        174405 non-null  object
 7   FAMI_TIENECOMPUTADOR      174405 non-null  object
 8   FAMI_TIENEAUTOMOVIL       174405 non-null  object
 9   COLE_GENERO               174405 non-null  object
 10  COLE_NATURALEZA           174405 non-null  object
 11  COLE_CALENDARIO           174405 non-null  object
 12  COLE_BILINGUE             174405 non-null  object
 13  COLE_AREA_UBICACION       174405 non-null  object
 14  COLE_JORNADA              174405 non-null  object
 15  PUNT_LECTURA_CRITICA      174405 non-null  int64
 16  PUNT_MATEMATICAS          174405 non-null  int64
 17  PUNT_C_NATURALES          174405 non-null  int64
 18  PUNT_SOCIALES_CIUDADANAS  174405 non-null  int64
 19  PUNT_INGLES               174405 non-null  int64
 20  ESTU_NSE_INDIVIDUAL       174405 non-null  object
 21  area_conocimiento         174405 non-null  object
 22  puntajeGLob_pro_tyt       174405 non-null  int64
dtypes: int64(6), object(17)
memory usage: 30.6+ MB
```

4.3.1 División del conjunto de datos en subconjuntos de entrenamiento y prueba

Para evaluar la capacidad predictiva de un modelo, es esencial verificar qué tan cercanas están sus predicciones a los valores reales de la variable respuesta. Para lograr una medición precisa, es necesario contar con un conjunto de datos que incluya observaciones con valores conocidos de la variable respuesta, pero que no se hayan utilizado para entrenar el modelo. Para esto, se dividen los datos disponibles en un conjunto de entrenamiento y un conjunto de prueba. La elección del tamaño adecuado de estas divisiones depende de la cantidad de datos disponibles y del nivel de confianza requerido en la estimación del error. En general, una división del 80% para entrenamiento y 20% para prueba suele arrojar resultados satisfactorios. Además, es fundamental realizar esta división de forma aleatoria para garantizar la validez de las evaluaciones del modelo.

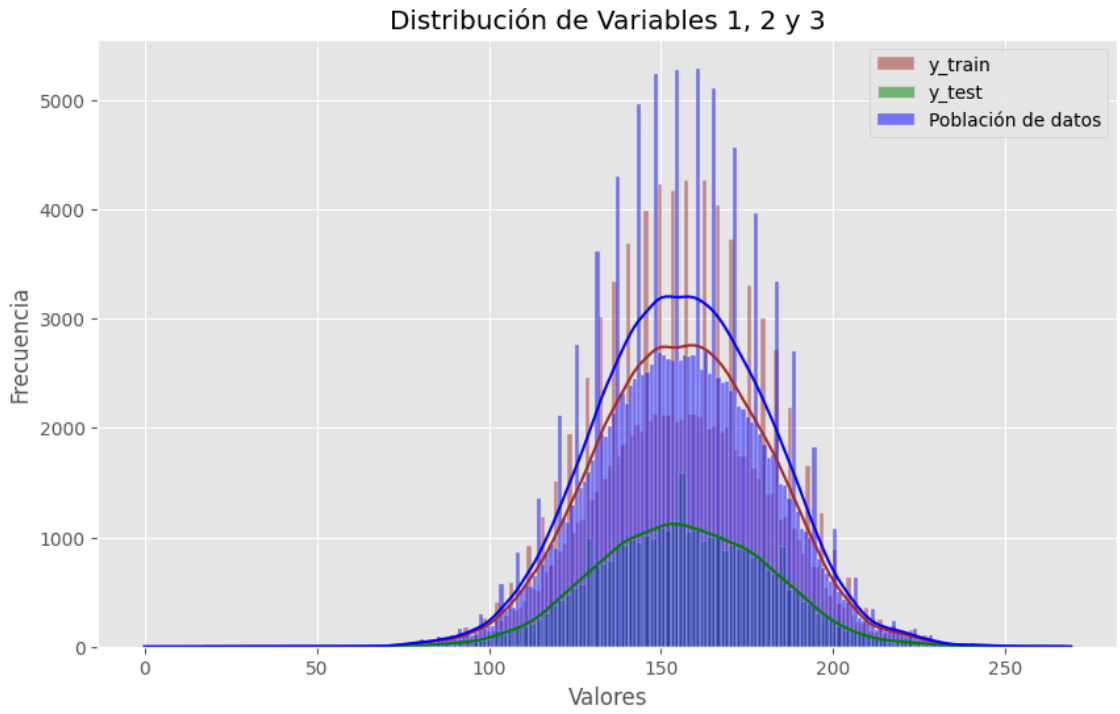


Figura 12. *Reducción de variables cualitativas*

Fuente: Elaboración propia.

La gráfica anterior muestra la población de datos y dos particiones de datos: una de entrenamiento y otra de prueba, ambas relacionadas con la variable "puntajeGLob_pro_tyt" o variable resultado u objetivo. Para la partición de entrenamiento, en las estadísticas descriptivas que a continuación se muestran, se observa que se compone de 139,524 observaciones, con un puntaje promedio de aproximadamente 156.14. La desviación estándar, que mide la dispersión de los valores alrededor de la media, es de alrededor de 25.06. Los puntajes varían desde un mínimo de 0 hasta un máximo de 269.

En cuanto a la partición de prueba, contiene 34,881 observaciones. Al analizar la variable "puntajeGLob_pro_tyt", se encuentra que la media es ligeramente inferior a la de la partición de entrenamiento, aproximadamente 156.03. La desviación estándar es similar, alrededor de 24.97, lo que sugiere una dispersión de puntajes comparable. Los valores de puntaje en esta partición varían

desde un mínimo de 18 hasta un máximo de 267, y los cuartiles son muy similares a los de la partición de entrenamiento.

Estos resultados indican que ambas particiones, tanto la de entrenamiento como la de prueba, presentan estadísticas descriptivas muy parecidas para la variable "puntajeGLob_pro_tyt". Esto sugiere que las particiones están equilibradas y representan adecuadamente la distribución de los puntajes, lo que es esencial para el desarrollo y evaluación de modelos analíticos.

```
print("Partición de entrenamento")
print("------------------------")
print(y_train.describe())

Partición de entrenamento
------------------------
count    139524.000000
mean        156.135969
std          25.055095
min           0.000000
25%         139.000000
50%         156.000000
75%         173.000000
max         269.000000
Name: puntajeGLob_pro_tyt, dtype: float64

print("Partición de test")
print("------------------------")
print(y_test.describe())

Partición de test
------------------------
count    34881.000000
mean       156.034775
std         24.972717
min         18.000000
25%        139.000000
50%        156.000000
75%        173.000000
max        267.000000
Name: puntajeGLob_pro_tyt, dtype: float64
```

Las dos imágenes siguientes ilustran la distribución de los datos de entrenamiento y pruebas en cada área del conocimiento. Esto garantiza la integridad y representatividad adecuada de los datos utilizados para construir y posteriormente validar el modelo. Este enfoque se ha implementado para minimizar posibles errores en la modelación y la validación, permitiendo la selección del modelo más adecuado para abordar la predicción del desempeño académico en la prueba Saber Pro con base en datos obtenidos al finalizar el bachillerato o la educación media.

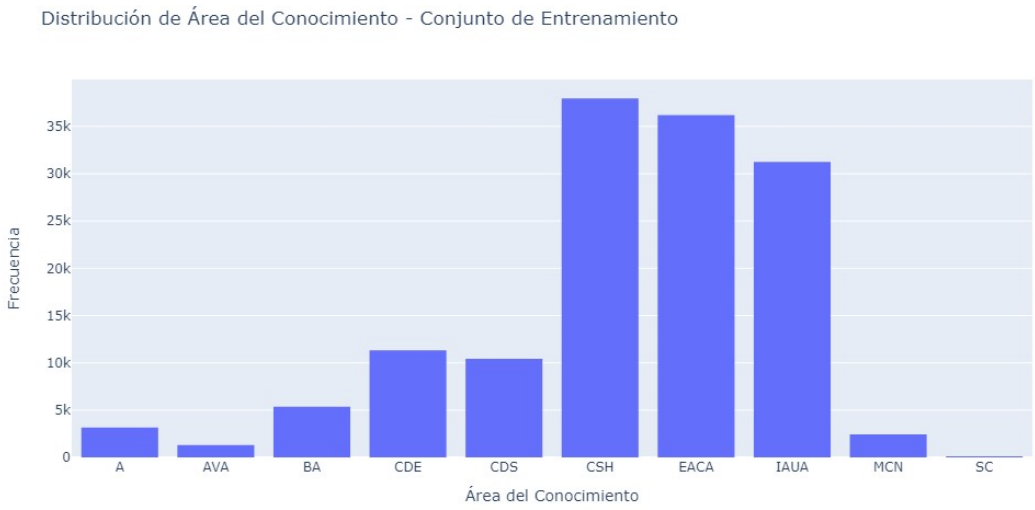


Figura 13. *Distribución de los datos de entrenamiento por área del conocimiento*
Fuente: Elaboración propia.

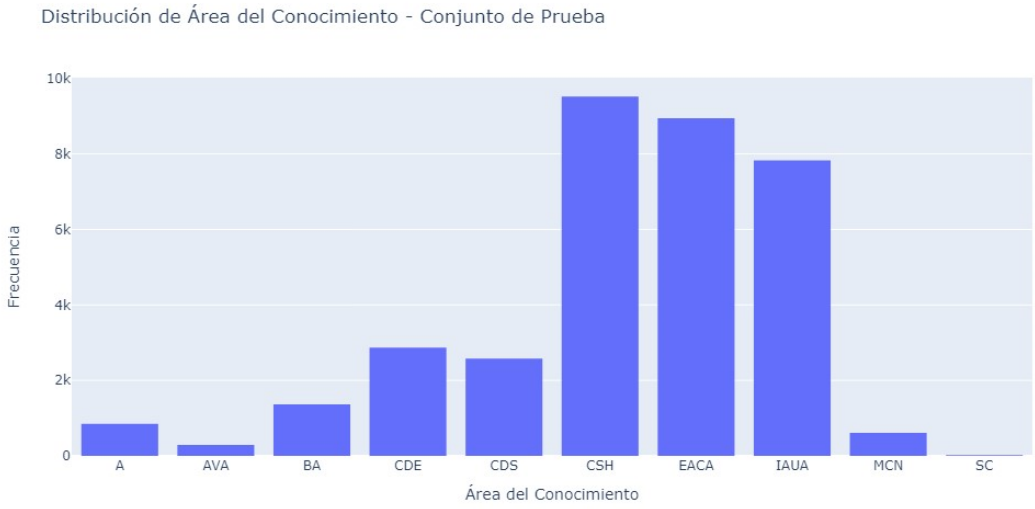


Figura 14. *Distribución de los datos de pruebas por área del conocimiento*
Fuente: Elaboración propia.

Al analizar las imágenes que representan la distribución de los datos de entrenamiento y pruebas en cada área del conocimiento, se concluye que este enfoque brinda una evaluación equilibrada de la integridad de los datos utilizados en la construcción y validación del modelo. La adecuada representatividad de los datos en las diversas áreas del conocimiento asegura que el modelo

considere la variabilidad inherente en cada disciplina, lo que contribuye a su capacidad para generalizar y predecir el desempeño académico en la prueba Saber Pro.

Es relevante destacar que, aunque se consideran todas las áreas del conocimiento, se observa que las dos con mayor representatividad son Ciencias Sociales y Humanas (CSH) y Economía, Administración, Contaduría y Afines (EACA), mientras que las áreas con menor representación son Agronomía, Veterinaria y Afines (AVA) y Matemáticas y Ciencias Naturales (MCN).

4.3.2 Identificación de valores ausentes

La mayoría de los algoritmos no aceptan datos incompletos, lo que plantea tres posibles enfoques para lidiar con valores faltantes:

- Eliminar observaciones con datos incompletos: Este método puede aplicarse si el conjunto de datos es lo suficientemente grande y el porcentaje de registros con valores faltantes es bajo, como es el caso de la base de datos actual.
- Eliminar variables con valores ausentes: La viabilidad de esta opción depende de cuán importantes sean las variables para el modelo, un tema que ya se discutió en el contexto de la reducción de variables cualitativas y numéricas.
- Imputar valores faltantes con información disponible: Aunque no se utilizó esta opción en el presente proyecto, es importante destacar que la imputación conlleva riesgos, especialmente al estimar valores en predictores críticos para el modelo.

En este contexto, se optó por eliminar observaciones con datos incompletos antes de ser cargados al ambiente de desarrollo del modelo, dado el tamaño de la base de datos y el bajo porcentaje de registros incompletos.

4.3.3 Estandarización de variables numéricas

```
# Selección de las variables por tipo
# ============================================================================
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
```

```
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.compose import make_column_selector

numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()

# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[
                            ('imputer', SimpleImputer(strategy='median')),
                            ('scaler', StandardScaler())
                        ]
                      )


# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[
                                ('imputer', SimpleImputer(strategy='most_frequent')),
                                ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False))
                            ]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

X_train_prep = preprocessor.fit_transform(X_train)
X_test_prep  = preprocessor.transform(X_test)
```

A continuación, se detalla las acciones que se realizó al conjunto de datos.

- Estandarización de variables numéricas: El primer paso implica estandarizar las columnas numéricas, es decir, asegurarse de que todas tengan una escala común y que sus valores tengan una distribución con media cero y desviación estándar igual a uno. Esto es importante porque muchos algoritmos de aprendizaje automático son sensibles a la escala de las variables. La estandarización garantiza que las variables numéricas tengan un impacto similar en el modelo, independientemente de sus unidades o magnitudes. La función StandardScaler de Scikit-Learn se utiliza para llevar a cabo este proceso.
- Codificación one-hot de variables cualitativas: El segundo paso consiste en realizar una codificación one-hot de las variables cualitativas (también conocidas como categóricas). Esta técnica se emplea para convertir variables que representan categorías en un formato binario, de

manera que cada categoría se convierte en una columna binaria (0 o 1). La codificación one-hot es fundamental para que los algoritmos de aprendizaje automático puedan utilizar estas variables en modelos matemáticos, ya que estos requieren datos numéricos. La función OneHotEncoder de Scikit-Learn se encarga de realizar esta codificación, y el argumento handle_unknown='ignore' permite que se manejen nuevas categorías que pueden aparecer en el conjunto de prueba.

- El objeto ColumnTransformer se encarga de aplicar ambas transformaciones a las columnas correspondientes. Es importante notar que, al especificar remainder='passthrough', se asegura de que las columnas no incluidas en ninguno de los dos pasos (por ejemplo, columnas numéricas que no requieren estandarización ni codificación one-hot) se conserven sin cambios.

4.4 Modelaje

Para este proyecto, se desarrollarán cinco modelos analíticos de predicción: K-Nearest Neighbor (KNN), Regresión Lineal (Ridge), Random Forest, Gradient Boosting Trees y Stacking. Para cada uno de los modelos se realizará un proceso sistemático y de verificación de la precisión de modelaje a través del cálculo de métricas que indicarán la integridad del modelo y su precisión al predecir. A continuación, se proporcionará un detalle del proceso de modelado, utilizando el modelo de Regresión Lineal Ridge como ejemplo. Posteriormente, se procederá a la construcción de los modelos restantes con el fin de compararlos y seleccionar el más apropiado.

4.4.1 Entrenamiento

```
1.  from sklearn.linear_model import Ridge
2.
3.  # Preprocedado
4.  # ==============================================================================
5.
6.  # Identificación de columnas numéricas y categóricas
7.  numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
8.  cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()
9.
10.
11. # Transformaciones para las variables numéricas
12. numeric_transformer = Pipeline(
13.                         steps=[('scaler', StandardScaler())]
```

```
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Pipeline
# ==============================================================================

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

# Train
# ==============================================================================
# Se asigna el resultado a _ para que no se imprima por pantalla
_ = pipe.fit(X=X_train, y=y_train)

# Validación cruzada
# ==============================================================================
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(
                estimator = pipe,
                X         = X_train,
                y         = y_train,
                scoring   = 'neg_root_mean_squared_error',
                cv        = 5
            )

print(f"Métricas validación cruzada: {cv_scores}")
print(f"Média métricas de validación cruzada: {cv_scores.mean()}")

Métricas validación cruzada: [-15.58245341 -15.27970465 -15.30816935 -15.38869873 -15.19339662]
Média métricas de validación cruzada: -15.350484551954347
```

El anterior código implementa un modelo de regresión lineal Ridge como parte del proceso de análisis de datos y modelado. A continuación, se detalla la finalidad de cada sección del código:

- Preprocesamiento de datos: Antes de ajustar un modelo, es crucial preprocesar los datos. Este proceso se divide en dos partes:
    - Identificación de columnas numéricas y categóricas: Se separan las columnas numéricas de las categóricas en el conjunto de datos, lo que es fundamental para aplicar transformaciones específicas a cada tipo de variable.

  - Transformaciones para variables numéricas y categóricas: Se definen las transformaciones necesarias para las variables numéricas y categóricas. Para las variables numéricas, se aplica un escalado estándar (StandardScaler) que asegura que todas tengan una media de cero y desviación estándar de uno. Para las variables categóricas, se realiza una codificación one-hot (OneHotEncoder) para convertirlas en variables binarias.

- Pipeline: Un pipeline se utiliza para combinar las etapas de preprocesamiento y modelado en un solo flujo de trabajo. En este caso, se crea un pipeline que consta de dos etapas: preprocesamiento y modelado. El modelo seleccionado es la regresión lineal Ridge.
- Entrenamiento del modelo: Se ajusta el modelo Ridge a los datos de entrenamiento (X_train e y_train) utilizando el pipeline definido.
- Validación cruzada: Se utiliza la validación cruzada (cross_val_score) para evaluar el rendimiento del modelo en varios pliegues (folds) de los datos de entrenamiento. En este caso, se emplea la métrica de error de raíz cuadrada media negativa (neg_root_mean_squared_error) como medida de rendimiento.
- resultados de la validación cruzada: Se imprimen las métricas de validación cruzada y su media. Estas métricas son útiles para evaluar qué tan bien se desempeña el modelo Ridge en diferentes conjuntos de datos de entrenamiento, lo que proporciona una estimación de su capacidad para generalizar a datos no vistos.

En cuanto al resultado de la validación cruzada obtenidos, dan luz sobre el rendimiento del modelo de regresión lineal Ridge y su capacidad para realizar predicciones precisas. Utilizando la métrica de error de raíz cuadrada media negativa, se evaluó el modelo en un proceso de validación cruzada con cinco pliegues. Los resultados de esta evaluación se presentan en forma de métricas para cada pliegue, lo que nos permite observar cómo el modelo se comporta en diferentes subconjuntos de

datos de entrenamiento. Esto es esencial para comprender la estabilidad y la consistencia de las predicciones.

La métrica promedio de validación cruzada es un indicador clave del rendimiento general del modelo. En este caso, la media de las métricas se sitúa alrededor de -15.35. La presencia del signo negativo se debe a que estamos trabajando con una métrica de puntuación, donde valores más bajos son mejores. Esto nos sugiere que el modelo Ridge está demostrando un rendimiento sólido en la tarea de predicción.

Una ventaja adicional de estos resultados es que nos permiten comparar el rendimiento del modelo Ridge con otros algoritmos que se utilizarán y mostrarán más adelante. Esta comparación nos ayudará a tomar decisiones fundamentadas sobre cuál es el modelo más adecuado para resolver nuestro problema particular de estimar el puntaje de desempeño académico en las pruebas universitarias saber pro con variables sociales, económicas, de educación y rendimiento académico, que posee un estudiante al culminar sus estudios de bachillerato.

Es importante destacar que también observamos la variabilidad en las métricas de validación cruzada entre los pliegues. Esta variabilidad puede proporcionar información sobre la robustez del modelo. Si la variabilidad es baja, podemos tener confianza en que el modelo ofrece un rendimiento consistente en diferentes conjuntos de datos de entrenamiento, lo cual se visualiza claramente en la siguiente figura, ya que el rango de variabilidad de la validación cruzada se encuentra entre -15.7 y -15.0.

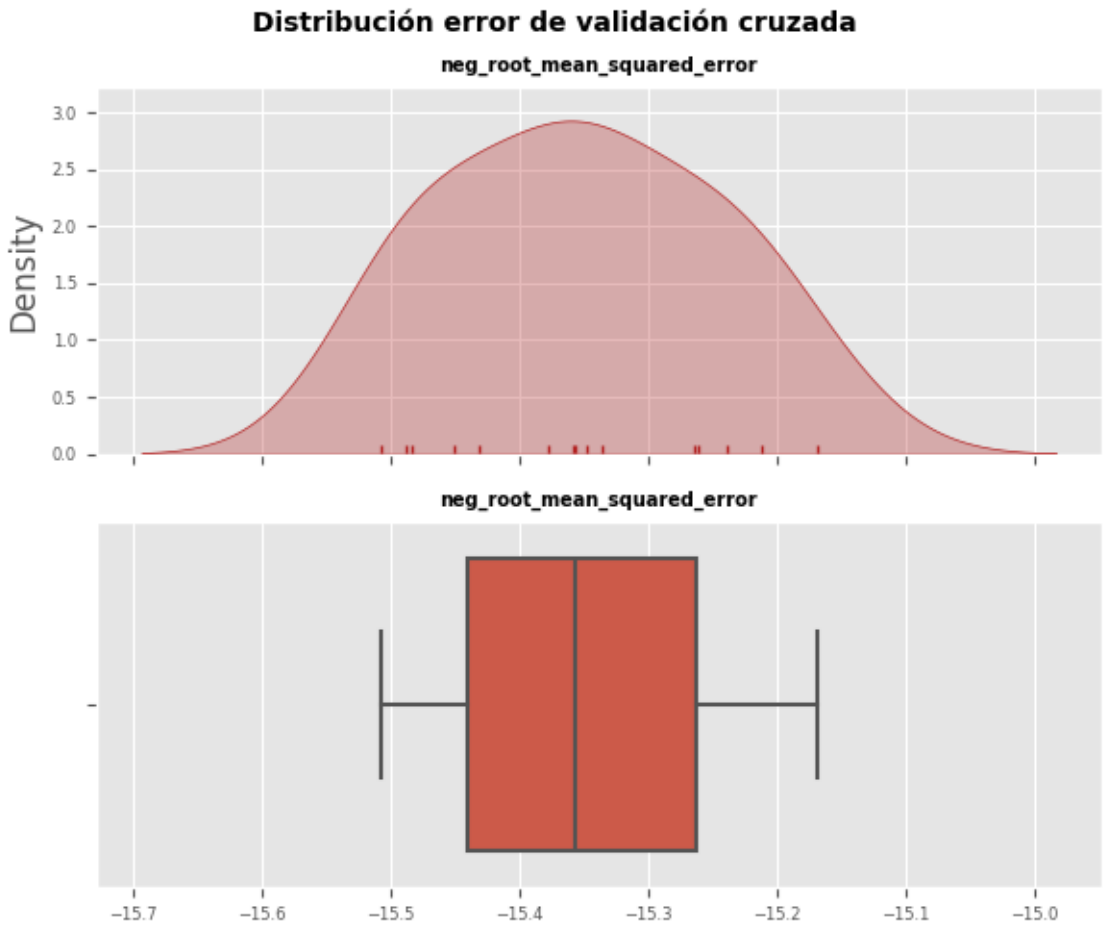


Figura 15. *Distribución del error de la validación cruzada*

Fuente: Elaboración propia.

4.4.2 Diagnóstico de residuos de la validación cruzada

La siguiente figura da un diagnóstico de los errores, también conocidos como residuos, de las predicciones realizadas durante la validación cruzada del modelo. Este diagnóstico es fundamental para evaluar la calidad del modelo y garantizar su fiabilidad en la predicción de datos no vistos.

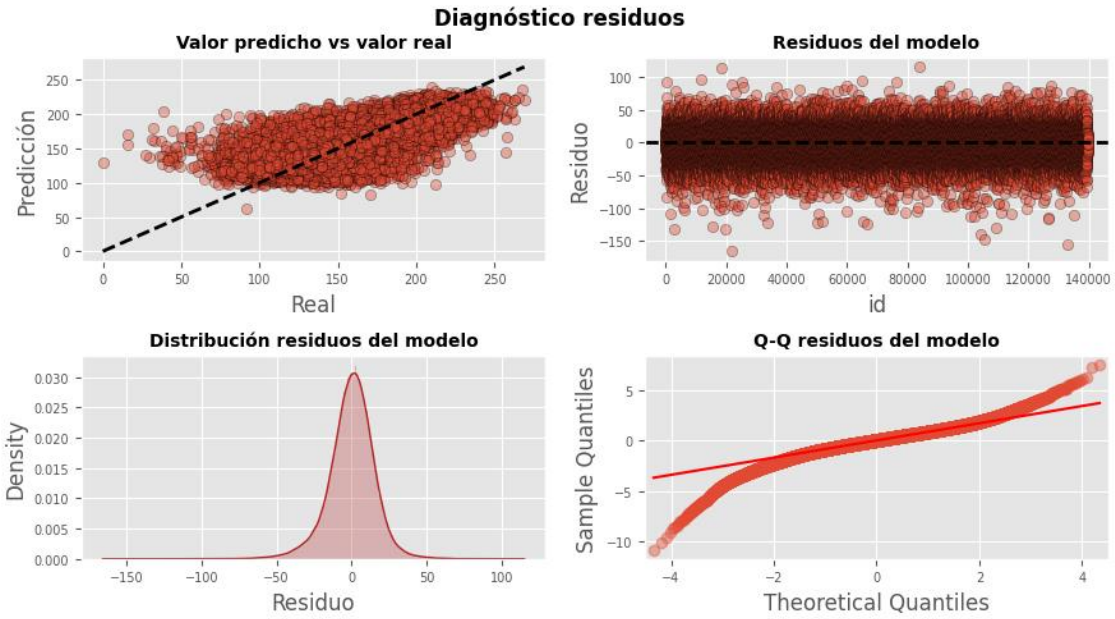


Figura 16. *Diagnóstico de residuos de la validación cruzada*

Fuente: Elaboración propia.

El diagnóstico se divide en varias partes, que se muestran en gráficos y estadísticas. A continuación, se describen estas secciones:

- validación cruzada: Primero, se utiliza la validación cruzada con 5 divisiones (K-Fold) para obtener las predicciones del modelo en los datos de entrenamiento. Esto permite evaluar el rendimiento del modelo en diferentes subconjuntos de datos y reducir el riesgo de sobreajuste.
- Gráfico de Valor Predicho vs. Valor Real: La primera parte del diagnóstico presenta un gráfico de dispersión que compara los valores reales de la variable objetivo con las predicciones generadas durante la validación cruzada. Cada punto en el gráfico representa una observación. La línea diagonal punteada representa una predicción perfecta. La finalidad de este gráfico es evaluar la proximidad de las predicciones al valor real y detectar posibles patrones de sesgo en las predicciones. En este caso, se puede observar que las predicciones del puntaje en la prueba Saber Pro se encuentran cercanas a los valores reales.

- Gráfico de Residuos: La segunda parte del diagnóstico muestra un gráfico de los residuos del modelo. Los residuos son la diferencia entre los valores reales y las predicciones. Este gráfico permite evaluar si los residuos están distribuidos de manera aleatoria alrededor del valor cero, un supuesto fundamental en los modelos de regresión lineal. La línea horizontal punteada en cero ayuda a identificar cualquier tendencia o patrón en los residuos. En este caso, se puede observar que la amplitud de los residuos del modelo oscila en un rango de aproximadamente 50 a -50 puntos.
- Histograma de Residuos: En la tercera sección, se presenta un histograma que representa la distribución de los residuos generados por el modelo. Un histograma que exhiba simetría alrededor del valor cero y tenga una forma similar a una campana sugiere que los residuos siguen una distribución normal. Este es un supuesto fundamental en muchos modelos estadísticos y, en el caso de los datos y el modelo analizados, se cumplió esta condición.
- Gráfico Q-Q de Residuos: En la última parte, se muestra un gráfico Q-Q (quantil-cuantil) de los residuos, que compara su distribución con una distribución teórica normal. Si los puntos en este gráfico siguen aproximadamente una línea recta, esto indica que los residuos siguen una distribución normal. Aunque la mayoría de las distribuciones de las pruebas aplicadas muestran una distribución normal en la gráfica Q-Q plot de residuos, la prueba de inglés no refleja esta distribución de manera tan clara. Esto sugiere que los datos pueden contener algunos puntos atípicos, aunque no pueden considerarse como anomalías graves. Es crucial destacar que estos datos atípicos no deben eliminarse del conjunto de datos procesados, ya que representan una pequeña población y pueden proporcionar información valiosa sobre ciertos aspectos del fenómeno en estudio.

La figura que se presenta a continuación muestra la distribución de la variable objetivo junto con la distribución de las predicciones. En el contexto del modelo de regresión lineal Ridge, ambos conjuntos de datos exhiben una distribución que se asemeja a una distribución normal. Esto sugiere

que el modelo realiza predicciones con mayor precisión cuando se trata de puntajes cercanos a la media de los datos, mientras que su precisión disminuye a medida que los valores se alejan de la media.

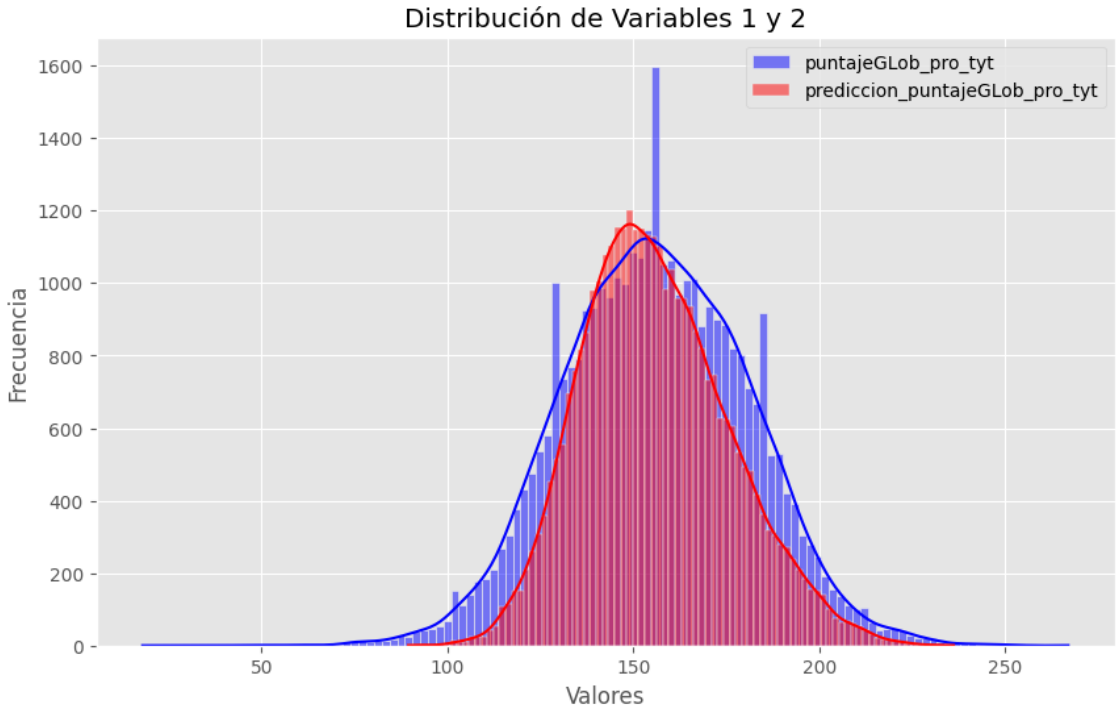


Figura 17. *Distribución de la objetivo y datos predichos*

Fuente: Elaboración propia.

4.4.3 Regresión Lineal (Ridge)

La Regresión Lineal Ridge representa una evolución de la regresión lineal convencional, un algoritmo muy extendido en el ámbito de la estadística y el aprendizaje automático. Lo que distingue a la Regresión Ridge es su enfoque de refinamiento de la regresión lineal para abordar un problema común, el sobreajuste.

En términos sencillos, la regresión lineal busca establecer una relación lineal entre las características de entrada y la variable de salida. Sin embargo, cuando tratamos con conjuntos de datos complejos o con muchas características, la regresión lineal puede volverse propensa al

sobreajuste. Esto significa que el modelo se vuelve demasiado específico para el conjunto de datos de entrenamiento y no generaliza bien a nuevos datos.

Para abordar este desafío, la Regresión Ridge introduce una pequeña modificación en la función de coste que guía el proceso de aprendizaje del modelo. Agrega una penalización a los coeficientes de las características, lo que impide que tomen valores extremadamente altos. Esta penalización controla el sobreajuste al restringir la magnitud de los coeficientes, lo que a menudo resulta en modelos más estables y robustos. (Ridge Regression Explained, Step by Step, 2021).

A continuación, se muestra el código del modelo de Regresión Lineal Ridge:

```
from sklearn.model_selection import RandomizedSearchCV, RepeatedKFold
from sklearn.linear_model import Ridge

# Pipeline: preprocesado + modelo
# ==============================================================================
# Identificación de columnas numéricas y catégoricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()


# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore',
 sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline.
pipe = Pipeline([('preprocessing', preprocessor),
                 ('modelo', Ridge())])

# Optimización de hiperparámetros
# ==============================================================================
# Espacio de búsqueda de cada hiperparámetro
param_distributions = {'modelo__alpha': np.logspace(-5, 5, 250)}

# Búsqueda random grid
grid = RandomizedSearchCV(
        estimator  = pipe,
        param_distributions = param_distributions,
```

```
        n_iter     = 5,
        scoring    = 'neg_root_mean_squared_error',
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = RepeatedKFold(n_splits = 5, n_repeats = 3),
        refit      = True,
        verbose    = 0,
        random_state = 123,
        return_train_score = True
       )

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

# Resultados del grid
# ==============================================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param.*|mean_t|std_t)')\
    .drop(columns = 'params')\
    .sort_values('mean_test_score', ascending = False)\
    .head(1)

param_modelo__alpha     mean_test_score
   std_test_score  mean_train_score        std_train_score
1       323.623582      -15.352319      0.093741        -15.347758      0.023321

# Error de test del modelo final
# ==============================================================================
modelo_final = grid.best_estimator_
predicciones = modelo_final.predict(X=X_test)

# Calcula las métricas
rmse_lm = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=False
)
mse_lm = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=True
)
mae_lm = mean_absolute_error(
    y_true=y_test,
    y_pred=predicciones
)

# Imprime las métricas
print(f"El error (RMSE) de test es: {rmse_lm}")
print(f"El error (MSE) de test es: {mse_lm}")
print(f"El error (MAE) de test es: {mae_lm}")

El error (RMSE) de test es: 15.223821913650927
El error (MSE) de test es: 231.76475365855813
El error (MAE) de test es: 11.291604306942276
```

Este código realiza un proceso de modelado predictivo utilizando el algoritmo Ridge Regression con una optimización de hiperparámetros mediante búsqueda aleatoria. Primero, se importa las librerías esenciales, como RandomizedSearchCV, RepeatedKFold y Ridge, para llevar a cabo el modelado predictivo. Luego, se establece un pipeline, que es básicamente una secuencia de pasos

que incluye tanto el preprocesamiento de datos como el modelo Ridge. Este enfoque estructurado facilita la replicación y el mantenimiento del proceso.

Se identificó las columnas numéricas y categóricas en los datos de entrenamiento, lo cual es fundamental para aplicar las transformaciones adecuadas durante el preprocesamiento. Luego, se crea transformadores específicos para las variables numéricas y categóricas utilizando Pipelines, aplicando escalado y codificación one-hot según corresponda.

Se utilizó un ColumnTransformer para combinar estas transformaciones de variables, lo que permitió procesar ambos tipos de variables simultáneamente. La combinación final de preprocesamiento y el modelo Ridge se realiza mediante un pipeline, simplificando la implementación y optimización de todo el proceso predictivo.

Ahora, entramos en la etapa de optimización de hiperparámetros, definiendo un espacio de búsqueda para el hiperparámetro 'alpha' del modelo Ridge. Se aplicó una búsqueda aleatoria con RandomizedSearchCV, lo que ayuda a encontrar la combinación óptima de hiperparámetros para el modelo.

Posteriormente, se evalúa los resultados del grid, presentando la mejor combinación de hiperparámetros y sus métricas asociadas. Con esta información, ajustamos el modelo con los mejores hiperparámetros encontrados y evaluamos su desempeño en un conjunto de datos de prueba. Finalmente, imprimimos las métricas de evaluación, incluyendo el error (RMSE, MSE, MAE) en los datos de prueba.

En cuanto los resultados arrojados en la etapa de optimización, se identificó que el valor más adecuado para el hiperparámetro 'alpha' en el modelo Ridge fue de aproximadamente 323.62. Este valor se eligió para maximizar la métrica de evaluación, en este caso, la raíz cuadrada del error medio cuadrático negativo (neg_root_mean_squared_error).

Posteriormente, al evaluar el modelo final en un conjunto de datos de prueba, se obtuvieron métricas clave. El RMSE (Root Mean Squared Error) reveló un error promedio en las predicciones de alrededor de 15.22 puntos, indicando la raíz cuadrada del promedio de los errores al cuadrado. El MSE (Mean Squared Error) mostró un error cuadrático promedio de aproximadamente 231.76, mientras que el MAE (Mean Absolute Error) presentó un error absoluto promedio de alrededor de 11.29 puntos.

Modelo	RMSE	MSE	MAE
Regresión Lineal (Ridge) métricas base o de comparación	15,2238	231,7648	11,2916

Tabla 5. *Evaluación del modelo de Regresión Lineal (Ridge)*

Fuente: Elaboración propia.

Los valores obtenidos a partir de este modelo base desempeñarán un papel crucial al ser comparados con los diversos modelos que se implementarán en este proyecto empresarial. Se interpreta que un RMSE bajo denota una sólida capacidad predictiva del modelo, y valores reducidos de MSE y MAE indican que el modelo realiza predicciones precisas sobre los datos de prueba. En resumen, estos resultados se utilizarán como referencia fundamental para gestionar los fenómenos de underfitting y overfitting en la estimación de los demás modelos desarrollados.

4.4.4 K-Nearest Neighbor (KNN)

El algoritmo K-Nearest Neighbor (KNN), a menudo abreviado como K-NN, se sitúa dentro de la categoría de algoritmos de aprendizaje supervisado y se caracteriza por su enfoque no paramétrico. A diferencia de algunos algoritmos que asumen una forma particular para la distribución de los datos, el KNN no hace suposiciones rígidas sobre la estructura de estos, lo que lo convierte en una herramienta versátil en el campo del aprendizaje automático.

El principio fundamental que guía al KNN es su dependencia de la proximidad. El algoritmo evalúa y toma decisiones basándose en la cercanía entre los puntos de datos. En otras palabras,

considera que los datos que están más cerca en el espacio de características son más similares entre sí que aquellos que están más alejados.

Cuando se aplica el KNN para clasificación, el algoritmo busca los K puntos de datos más cercanos al punto de datos que se desea clasificar. Luego, analiza la etiqueta de clase de esos vecinos más cercanos y asigna al punto de datos en cuestión la etiqueta de clase que es más común entre esos vecinos. De esta manera, el KNN utiliza la sabiduría de la multitud para tomar decisiones de clasificación.

En el contexto de regresión, el KNN se comporta de manera similar. En lugar de clasificar, busca los K puntos de datos más cercanos y promedia sus valores para predecir el valor numérico del punto de datos en cuestión. (¿Qué es KNN? | IBM, s. f.)

A continuación, se muestra el código del modelo K-Nearest Neighbor (KNN):

```
from sklearn.model_selection import RandomizedSearchCV, RepeatedKFold
from sklearn.neighbors import KNeighborsRegressor

# Pipeline: preprocesado + modelo
# ==============================================================================
# Identificación de columnas numéricas y catégoricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()


# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore',
    sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline.
pipe = Pipeline([('preprocessing', preprocessor),
                 ('modelo', KNeighborsRegressor())])

# Optimización de hiperparámetros
# ==============================================================================
```

```
# Espacio de búsqueda de cada hiperparámetro
param_distributions = {'modelo__n_neighbors': np.linspace(1, 100, 500, dtype=int)}

# Búsqueda random grid
grid = RandomizedSearchCV(
        estimator  = pipe,
        param_distributions = param_distributions,
        n_iter     = 5,
        scoring    = 'neg_root_mean_squared_error',
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = RepeatedKFold(n_splits = 5, n_repeats = 3),
        refit      = True,
        verbose    = 0,
        random_state = 123,
        return_train_score = True
       )

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

# Resultados del grid
# ==============================================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param.*|mean_t|std_t)')\
    .drop(columns = 'params')\
    .sort_values('mean_test_score', ascending = False)\
    .head(1)
param_modelo__n_neighbors       mean_test_score
   std_test_score  mean_train_score        std_train_score
3       95      -15.402668      0.0756  -15.241819      0.021151
# Importar la función mean_absolute_error desde sklearn.metrics
from sklearn.metrics import mean_squared_error, mean_absolute_error
# Error de test del modelo final
# ==============================================================================
modelo_final = grid.best_estimator_
predicciones = modelo_final.predict(X=X_test)

# Calcula las métricas
rmse_knn = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=False
)
mse_knn = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=True
)
mae_knn = mean_absolute_error(
    y_true=y_test,
    y_pred=predicciones
)

# Imprime las métricas
print(f"El error (RMSE) de test es: {rmse_knn}")
print(f"El error (MSE) de test es: {mse_knn}")
print(f"El error (MAE) de test es: {mae_knn}")

El error (RMSE) de test es: 15.271652993145304
El error (MSE) de test es: 233.2233851430439
El error (MAE) de test es: 11.36422120925432
```

El código presentado lleva a cabo un análisis de un modelo predictivo implementado mediante el algoritmo de Regresión de Vecinos más Cercanos (KNeighborsRegressor), optimizando sus hiperparámetros mediante una búsqueda aleatoria. Se inicia identificando las columnas numéricas y

categóricas en los datos de entrenamiento, seguido de la definición de transformaciones para estas variables y la combinación del preprocesamiento con el modelo en un único pipeline.

Posteriormente, se establece un espacio de búsqueda para el hiperparámetro "n_neighbors" del modelo KNeighborsRegressor. Se ejecuta una búsqueda aleatoria de hiperparámetros utilizando RandomizedSearchCV, configurando parámetros como el número de iteraciones y la métrica de evaluación. Los resultados obtenidos del grid son presentados, destacando el mejor valor encontrado para "n_neighbors" y las métricas asociadas al rendimiento del modelo.

Seguidamente, se obtiene el mejor estimador del grid y se realizan predicciones sobre los datos de prueba. Se calculan métricas de evaluación, incluyendo RMSE, MSE y MAE, sobre las predicciones del modelo final. El análisis revela que el error (RMSE) en los datos de prueba es aproximadamente 15.27, el error (MSE) es alrededor de 233.22, y el error (MAE) es aproximadamente 11.36. Los valores de RMSE y MAE son mejores que los obtenidos por el modelo de Regresión Lineal (Ridge).

Modelo	RMSE	MSE	MAE
K-Nearest Neighbor (KNN)	15,2717	231,2239	11,3642

Tabla 6. *Evaluación del modelo K-Nearest Neighbor (KNN)*

Fuente: Elaboración propia.

4.4.5 Random Forest

Random Forest, conocido en español como "Bosque Aleatorio," representa una poderosa técnica en el campo del aprendizaje automático. Su enfoque central radica en la combinación de múltiples árboles de decisión para generar una única y precisa predicción. Esta metodología ha ganado una gran popularidad debido a su simplicidad y versatilidad, lo que la convierte en una herramienta invaluable para una variedad de aplicaciones en problemas de clasificación y regresión.

En esencia, el Random Forest opera mediante la construcción de múltiples árboles de decisión, cada uno de los cuales representa un "experto" en la toma de decisiones. Estos árboles se crean utilizando diferentes subconjuntos de datos de entrenamiento y un enfoque de selección de características al azar. Esta diversidad es fundamental, ya que evita que el modelo se base en un solo árbol y, en cambio, aprovecha la sabiduría colectiva de varios.

Una vez que se han creado todos los árboles, Random Forest realiza una especie de "votación" para llegar a una decisión final. Cada árbol emite su propia predicción y la predicción final se determina mediante la mayoría. Este enfoque combinatorio reduce el riesgo de sobreajuste, ya que los errores en un árbol individual tienden a ser compensados por la precisión de otros. (Team Dst, 2022).

A continuación, se muestra el código del modelo Random Forest:

```
from sklearn.model_selection import RandomizedSearchCV, RepeatedKFold
from sklearn.ensemble import RandomForestRegressor

# Pipeline: preprocesado + modelo
# ==============================================================================
# Identificación de columnas numéricas y catégoricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()


# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore',
    sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline.
pipe = Pipeline([('preprocessing', preprocessor),
                 ('modelo', RandomForestRegressor())])

# Optimización de hiperparámetros
# ==============================================================================
# Espacio de búsqueda de cada hiperparámetro
```

```
37.
38. param_distributions = {
39.     'modelo__n_estimators': [50, 100, 1000, 2000],
40.     'modelo__max_features': [3, 5, 7, 1.0],
41.     'modelo__max_depth'   : [None, 3, 5, 10, 20]
42. }
43.
44. # Búsqueda random grid
45. grid = RandomizedSearchCV(
46.         estimator  = pipe,
47.         param_distributions = param_distributions,
48.         n_iter     = 5,
49.         scoring    = 'neg_root_mean_squared_error',
50.         n_jobs     = multiprocessing.cpu_count() - 1,
51.         cv         = RepeatedKFold(n_splits = 5, n_repeats = 3),
52.         refit      = True,
53.         verbose    = 0,
54.         random_state = 123,
55.         return_train_score = True
56.        )
57.
58. grid.fit(X = X_train, y = y_train)
59.
60. # Resultados del grid
61. # ==============================================================================
62. resultados = pd.DataFrame(grid.cv_results_)
63. resultados.filter(regex = '(param.*|mean_t|std_t)')\
64.     .drop(columns = 'params')\
65.     .sort_values('mean_test_score', ascending = False)\
66.     .head(1)
67. param_modelo__n_estimators      param_modelo__max_features      param_modelo__max_de
    pth mean_test_score std_test_score  mean_train_score        std_train_score
68. 4       100     5       20      -15.345513      0.06964 -9.781098
69.
70. # Error de test del modelo final
71. # ==============================================================================
72. modelo_final = grid.best_estimator_
73. predicciones = modelo_final.predict(X=X_test)
74.
75. # Calcula las métricas
76. rmse_rf = mean_squared_error(
77.     y_true=y_test,
78.     y_pred=predicciones,
79.     squared=False
80. )
81. mse_rf = mean_squared_error(
82.     y_true=y_test,
83.     y_pred=predicciones,
84.     squared=True
85. )
86. mae_rf = mean_absolute_error(
87.     y_true=y_test,
88.     y_pred=predicciones
89. )
90.
91. # Imprime las métricas
92. print(f"El error (RMSE) de test es: {rmse_rf}")
93. print(f"El error (MSE) de test es: {mse_rf}")
94. print(f"El error (MAE) de test es: {mae_rf}")
95.
96. El error (RMSE) de test es: 15.232683588169815
97. El error (MSE) de test es: 232.03464929729805
98. El error (MAE) de test es: 11.327585265677895
```

El código realiza un análisis del modelo RandomForestRegressor, abordando tanto el preprocesamiento de datos como la optimización de hiperparámetros. En la etapa de

preprocesamiento, se identifican las columnas numéricas y categóricas en los datos de entrenamiento, seguido por la definición de transformaciones específicas para cada tipo de variable. Se crea un preprocesador que aplica estas transformaciones de manera adecuada, y finalmente, se combina con el modelo RandomForestRegressor en un único pipeline.

En cuanto a la optimización de hiperparámetros, se establece un espacio de búsqueda que abarca parámetros clave como el número de estimadores, las características máximas y la profundidad máxima del modelo. Una búsqueda aleatoria (RandomizedSearchCV) se ejecuta sobre el pipeline definido, realizando 5 iteraciones y utilizando la métrica de error negativo de la raíz cuadrada media (neg_root_mean_squared_error) para evaluar el rendimiento. Los resultados de esta búsqueda revelan los mejores hiperparámetros identificados, siendo estos: n_estimators = 100, max_features = 5 y max_depth = 20.

Los resultados derivados de la optimización de hiperparámetros ofrecen una evaluación detallada del desempeño del modelo RandomForestRegressor. El error (RMSE) en el conjunto de prueba es aproximadamente 15.23, el error (MSE) oscila alrededor de 232.03, y el error (MAE) en la prueba se sitúa cerca de 11.33. Estas métricas son indicadores clave de la capacidad del modelo para realizar predicciones en datos de prueba no utilizados durante el entrenamiento. Al comparar estas métricas con los resultados obtenidos por el modelo base de Regresión Lineal (Ridge), se observa una ligera desmejora en los valores de RMSE, MSE y MAE.

Modelo	RMSE	MSE	MAE
Random Forest	15,2327	231,0346	11,3276

Tabla 7. *Evaluación del modelo Random Forest*

Fuente: Elaboración propia.

4.4.6 Gradient Boosting Trees

Gradient Boosting Trees es una técnica de machine learning ampliamente empleada en una variedad de tareas, como regresión y clasificación. Su característica distintiva radica en su

capacidad para construir un modelo predictivo mediante la combinación de múltiples modelos de predicción débiles. Estos modelos débiles suelen ser árboles de decisión simples que tienen la ventaja de hacer suposiciones mínimas acerca de la estructura subyacente de los datos.

La esencia del Gradient Boosting Trees radica en su capacidad para mejorar gradualmente el rendimiento del modelo al aprender de los errores cometidos en predicciones anteriores. Esto se logra mediante la construcción de nuevos modelos de árboles de decisión que se centran en corregir los errores del modelo anterior. Cada nuevo árbol se ajusta cuidadosamente para enfocarse en las instancias que han sido mal clasificadas o cuyas predicciones han tenido un alto error.

En esencia, Gradient Boosting Trees opera como un "maestro" que entrena una secuencia de "aprendices débiles", donde cada aprendiz se esfuerza por mejorar las áreas en las que los modelos anteriores no han sido precisos. Este enfoque iterativo y secuencial permite que el modelo se adapte y mejore continuamente su capacidad predictiva. (Gaurav, 2021)

A continuación, se muestra el código del modelo Gradient Bossting Trees:

```
from sklearn.model_selection import RandomizedSearchCV, RepeatedKFold
from sklearn.ensemble import GradientBoostingRegressor

# Pipeline: preprocesado + modelo
# ==============================================================================
# Identificación de columnas numéricas y catégoricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()


# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore',
 sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Se combinan los pasos de preprocesado y el modelo en un mismo pipeline.
```

```
pipe = Pipeline([('preprocessing', preprocessor),
                 ('modelo', GradientBoostingRegressor())])

# Optimización de hiperparámetros
# ==============================================================================
# Espacio de búsqueda de cada hiperparámetro

param_distributions = {
    'modelo__n_estimators': [50, 100, 1000, 2000],
    'modelo__max_features': [3, 5, 7, 1.0],
    'modelo__max_depth'   : [None, 3, 5, 10, 20],
    'modelo__subsample'   : [0.5,0.7, 1]
}

# Búsqueda random grid
grid = RandomizedSearchCV(
        estimator  = pipe,
        param_distributions = param_distributions,
        n_iter     = 5,
        scoring    = 'neg_root_mean_squared_error',
        n_jobs     = multiprocessing.cpu_count() - 1,
        cv         = RepeatedKFold(n_splits = 5, n_repeats = 3),
        refit      = True,
        verbose    = 0,
        random_state = 123,
        return_train_score = True
       )

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

# Resultados del grid
# ==============================================================================
resultados = pd.DataFrame(grid.cv_results_)
resultados.filter(regex = '(param.*|mean_t|std_t)')\
    .drop(columns = 'params')\
    .sort_values('mean_test_score', ascending = False)\
    .head(1)
param_modelo__subsample
   param_modelo__n_estimators      param_modelo__max_features      param_modelo__max_de
   pth mean_test_score std_test_score  mean_train_score        std_train_score
4       0.7     1000    7       3       -15.228578      0.086052        -
   14.975677      0.019761

# Error de test del modelo final
# ==============================================================================
modelo_final = grid.best_estimator_
predicciones = modelo_final.predict(X=X_test)

# Calcula las métricas
rmse_gbm = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=False
)
mse_gbm = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=True
)
mae_gbm = mean_absolute_error(
    y_true=y_test,
    y_pred=predicciones
)

# Imprime las métricas
print(f"El error (RMSE) de test es: {rmse_gbm}")
print(f"El error (MSE) de test es: {mse_gbm}")
print(f"El error (MAE) de test es: {mae_gbm}")

El error (RMSE) de test es: 15.118278676410482
```

```
El error (MSE) de test es: 228.56235013760787
El error (MAE) de test es: 11.208313536936114
```

El código realiza un proceso de preprocesamiento de datos y optimización de hiperparámetros para un modelo de regresión Gradient Boosting Trees. En primer lugar, se identifican y transforman las columnas numéricas y categóricas de los datos de entrenamiento, incorporándolas en un preprocesador que aplicará las transformaciones adecuadas. Luego, este preprocesador se fusiona con el modelo GradientBoostingRegressor en un pipeline consolidado.

En la fase de optimización de hiperparámetros, se establece un espacio de búsqueda que incluye variables como el número de estimadores, la máxima cantidad de características, la profundidad máxima del árbol y la tasa de submuestreo. A través de una búsqueda aleatoria (RandomizedSearchCV) con 5 iteraciones, se evalúa el rendimiento utilizando la métrica de error negativo de la raíz cuadrada media (neg_root_mean_squared_error), y se determinan los hiperparámetros óptimos encontrados durante este proceso.

Los mejores hiperparámetros obtenidos son: n_estimators: 1000, max_features: 7, max_depth: 3, y subsample: 0.7. Estos parámetros optimizados han contribuido a modelar de manera más precisa la relación entre las variables, como se refleja en las métricas de evaluación en el conjunto de datos de prueba.

En relación con los resultados, se observa que el error de prueba, medido por el RMSE, se sitúa en aproximadamente 15.12, mientras que el MSE se sitúa en torno a 228.56 y el MAE es cercano a 11.21. Estas cifras ofrecen una evaluación precisa de la capacidad predictiva del modelo Gradient Boosting Regressor después de optimizar sus hiperparámetros, demostrando su eficaz rendimiento en el conjunto de datos de prueba. Al comparar estos resultados con los obtenidos a partir del modelo base de Regresión Lineal (Ridge), se destaca una mejora en los valores de RMSE, MSE y MAE, subrayando la superioridad de la precisión predictiva alcanzada por el modelo Gradient Boosting Regressor.

Modelo	RMSE	MSE	MAE
Gradient Boosting Trees	15,1182	228,5623	11,2083

Tabla 8. *Evaluación del modelo Gradient Boosting Trees*

Fuente: Elaboración propia.

4.4.7 Stacking

Stacking, en el contexto del aprendizaje automático, se presenta como una técnica avanzada y poderosa que permite combinar múltiples modelos de clasificación o regresión. Aunque existen otras estrategias de agrupamiento de modelos, como Bagging y Boosting, el enfoque del stacking se distingue por su enfoque único en la exploración de un amplio espectro de modelos para abordar un problema específico.

En esencia, el proceso de stacking implica la construcción de un nuevo modelo, denominado "metamodelo" o "modelo maestro", que utiliza las predicciones de varios modelos base como entradas, para este ejercicio, se utilizó como base el modelo de Random Forest. En lugar de depender de un solo algoritmo de aprendizaje, se aprovecha la diversidad de enfoques y técnicas presentes en los modelos base para mejorar el rendimiento predictivo.

La selección de los modelos base, sus configuraciones y la forma en que se combinan las predicciones son decisiones clave en el proceso de stacking. Esta técnica se ha convertido en una herramienta esencial en la caja de herramientas de los científicos de datos, ya que permite abordar problemas desafiantes mediante la creación de ensambles de modelos que pueden adaptarse a la complejidad y la variabilidad de los datos. (Patel, 2020)

A continuación, se muestra el código del modelo Stacking:

```
from sklearn.linear_model import Ridge
from sklearn.linear_model import RidgeCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import StackingRegressor

# Pipeline: preprocesado + modelos para el stacking
# ==============================================================================
# Identificación de columnas numéricas y catégoricas
numeric_cols = X_train.select_dtypes(include=['float64', 'int']).columns.to_list()
```

```
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.to_list()

# Transformaciones para las variables numéricas
numeric_transformer = Pipeline(
                        steps=[('scaler', StandardScaler())]
                      )

# Transformaciones para las variables categóricas
categorical_transformer = Pipeline(
                            steps=[('onehot', OneHotEncoder(handle_unknown='ignore',
    sparse_output=False))]
                          )

preprocessor = ColumnTransformer(
                    transformers=[
                        ('numeric', numeric_transformer, numeric_cols),
                        ('cat', categorical_transformer, cat_cols)
                    ],
                    remainder='passthrough',
                    verbose_feature_names_out = False
                ).set_output(transform="pandas")

# Se combinan los pasos de preprocesado y los modelos creando varios pipeline.
pipe_ridge = Pipeline([('preprocessing', preprocessor),
                       ('ridge', Ridge(alpha=3.4))])

pipe_rf = Pipeline([('preprocessing', preprocessor),
                       ('random_forest', RandomForestRegressor(
                                            n_estimators = 1000,
                                            max_features = 7,
                                            max_depth    = 20
                                           )
                       )])
# Definición y entrenamiento del StackingRegressor
# ==============================================================================
estimators = [('ridge', pipe_ridge),
              ('random_forest', pipe_rf)]

stacking_regressor = StackingRegressor(estimators=estimators,
                                       final_estimator=RidgeCV())
# Se asigna el resultado a _ para que no se imprima por pantalla
_ = stacking_regressor.fit(X = X_train, y = y_train)

# Error de test del stacking
# ==============================================================================
modelo_final = stacking_regressor
predicciones = modelo_final.predict(X=X_test)

# Calcula las métricas
rmse_stacking = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=False
)
mse_stacking = mean_squared_error(
    y_true=y_test,
    y_pred=predicciones,
    squared=True
)
mae_stacking = mean_absolute_error(
    y_true=y_test,
    y_pred=predicciones
)

# Imprime las métricas
print(f"El error (RMSE) de test es: {rmse_stacking}")
print(f"El error (MSE) de test es: {mse_stacking}")
print(f"El error (MAE) de test es: {mae_stacking}")

El error (RMSE) de test es: 15.125724877498866
```

```
El error (MSE) de test es: 228.78755306978812
El error (MAE) de test es: 11.218762007612163
```

El código presenta la creación de un modelo de ensamblaje conocido como Stacking. Se han definido dos modelos base, un modelo de Regresión Ridge y un Regresor Random Forest, que han sido procesados y configurados individualmente. Luego, se ha utilizado un modelo StackingRegressor para combinar estos dos modelos base en una única predicción final.

El modelo StackingRegressor es una técnica de ensamblaje que aprovecha las fortalezas de los modelos base y utiliza un modelo adicional (en este caso, otro modelo de Regresión Ridge) para aprender a combinar las predicciones de los modelos base. Esto permite obtener un modelo final que, en teoría, debería ofrecer un rendimiento mejor que cualquiera de los modelos base individualmente.

Los resultados derivados del análisis del modelo analítico Stacking revelan un Root Mean Squared Error (RMSE) de 15.1257, un Mean Squared Error (MSE) de 228.7876 y un Mean Absolute Error (MAE) de 11.2188. Al contrastar estas métricas con las del modelo base de Regresión Lineal (Ridge), se evidencia una mejora sustancial en los valores de RMSE, MSE y MAE. Este hallazgo subraya la eficacia del modelo Stacking en la tarea de predecir el desempeño académico de los estudiantes en la prueba saber pro, posicionándolo como una opción superior en términos de precisión predictiva.

Modelo	RMSE	MSE	MAE
Stacking	15,1257	228,7876	11,2188

Tabla 9. *Evaluación del modelo Stacking*

Fuente: Elaboración propia.

4.5 Evaluación

Al seleccionar un modelo para la etapa de despliegue, es esencial buscar el equilibrio perfecto entre precisión y generalización. En el presente análisis, se evaluó cinco modelos diferentes y, con

una mirada crítica, se consideró sus métricas clave. El objetivo es minimizar el error en el conjunto de prueba para garantizar un rendimiento óptimo en etapas de despliegue y operación. A continuación, se presenta el resumen de las métricas de los modelos:

Modelo	RMSE	MSE	MAE
Regresión Lineal (Ridge) métricas base o de comparación	**15,22380**	**231,76480**	**11,29160**
K-Nearest Neighbor (KNN)	15,27170	231,22390	11,36420
Random Forest	15,23270	231,03460	11,32760
Gradient Boosting Trees	15,11820	228,56230	11,20830
Stacking	**15,12570**	**228,78760**	**11,21880**

Tabla 10. *Comparación del desempeño de los modelos*

Fuente: Elaboración propia.

En primer lugar, la "Regresión Lineal (Ridge)" presenta notoria discrepancia en su capacidad para generalizar los datos de prueba comparada con las métricas de los demás modelos, lo que podría limitar su eficacia en el despliegue final. Además de tener un RMSE de 15.2238, la métrica Mean Squared Error (MSE) asciende a 231.7648, y el Mean Absolute Error (MAE) alcanza 11.2916.

Aunque el "K-Nearest Neighbor (KNN)" muestra una RMSE relativamente baja en el conjunto de prueba, lo que indica un rendimiento aceptable, es crucial tener en cuenta que este enfoque podría requerir más tiempo para realizar predicciones debido a su naturaleza basada en la proximidad. Además del RMSE de 15.2717, el MSE se sitúa en 231.2239 y el MAE en 11.3642.

El modelo "Random Forest" destaca por su buen desempeño en el conjunto de prueba, con un RMSE razonable de 15.2327. Además, parece generalizarse bien, como se evidencia en el MSE de 231.0346 y el MAE de 11.3276. Sin embargo, al comparar con los modelos "Gradient Boosting Trees" y "Stacking", puede haber margen para mejoras adicionales.

Los modelos más destacados en nuestro análisis son "Gradient Boosting Trees" y "Stacking". Ambos presentan la RMSE más baja en el conjunto de prueba, con valores de 15.1182 y 15.1257 respectivamente. Además, al considerar el MSE y el MAE, muestran valores más bajos (MSE de 228.5623 y 228.7876, MAE de 11.2083 y 11.2188 respectivamente), lo que sugiere un rendimiento sólido y una excelente capacidad de generalización. Sin embargo, al ponderar las fortalezas específicas del modelo "Stacking" y su capacidad única para combinar las predicciones de varios modelos base, se toma la decisión de seleccionarlo como la opción preferida para avanzar a la etapa de despliegue. Esto se respalda por su capacidad para mejorar la precisión predictiva a través de la combinación estratégica de modelos, reduciendo la variabilidad y mejorando la robustez del modelo final. A continuación, se presenta en la figura la comparación de los RMSE de todos los modelos para una mejor visualización de sus desempeños relativos, considerando también las métricas MSE y MAE.

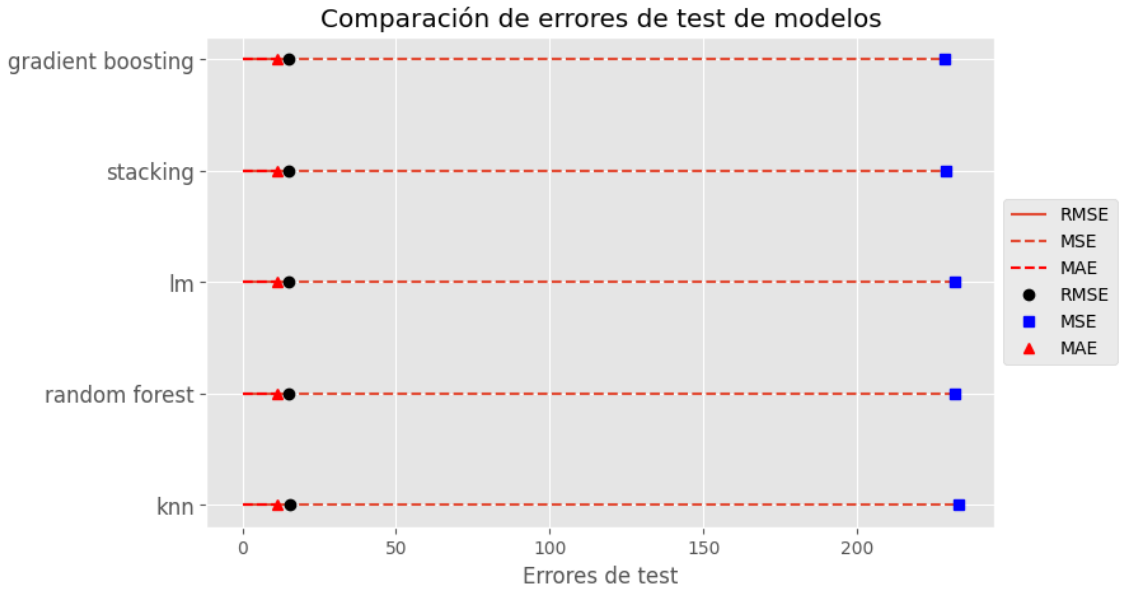


Figura 18. *Comparación del desempeño de los modelos*

Fuente: Elaboración propia.

4.5.1 Análisis de la relevancia en el modelo analítico seleccionado

En la figura siguiente se presenta, en un orden descendente de importancia, las variables utilizadas en el modelo analítico seleccionado de Stacking. En este análisis, se destaca que las variables relacionadas con el rendimiento académico, específicamente los resultados en ciencias naturales, inglés, lectura crítica, ciencias ciudadanas y matemáticas exhiben una influencia significativa en las predicciones del modelo en el conjunto de prueba. Este hallazgo sugiere que el desempeño en estas áreas académicas tiene un impacto sustancial en la capacidad predictiva del modelo. Seguidamente, las variables socioeconómicas y de educación, como el estrato de la vivienda, el género del estudiante, y la disponibilidad de acceso a internet o automóvil por parte de la familia, también desempeñan un papel relevante, aunque en menor medida en comparación con las variables de desempeño académico. Esta jerarquía de importancia brinda una visión detallada de cómo diversas categorías de variables contribuyen al poder predictivo global del modelo, destacando la significativa influencia de los indicadores de rendimiento académico.

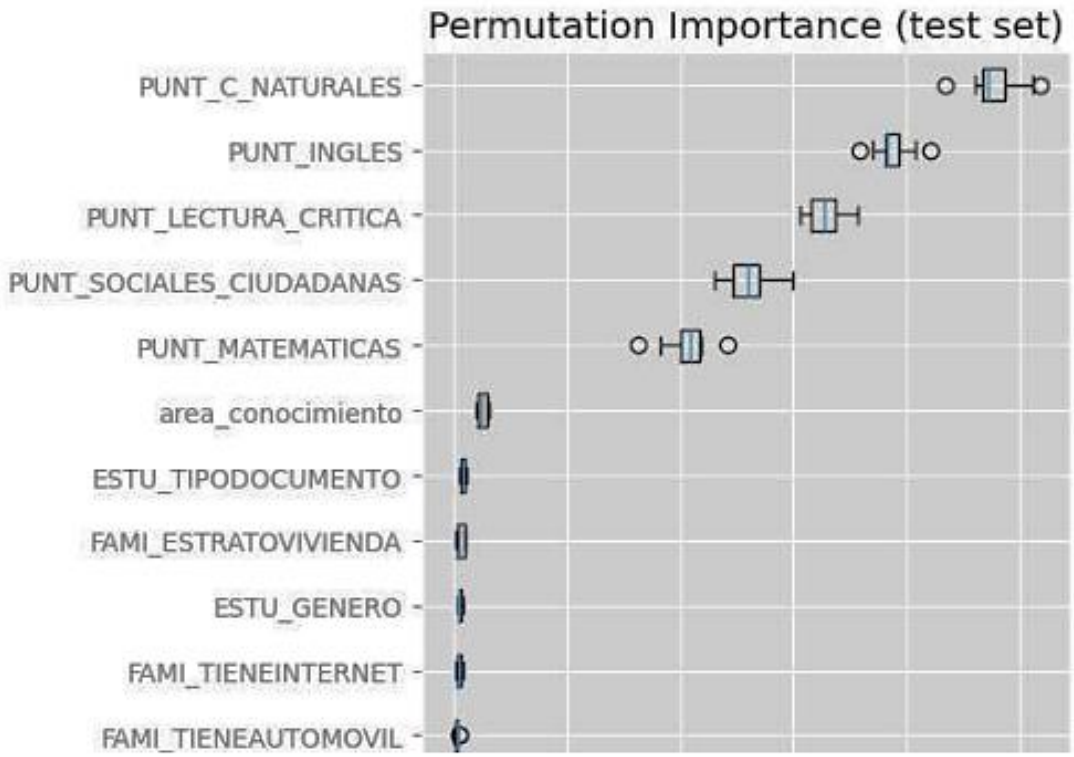


Figura 19. Relevancia de las variables del modelo

Fuente: Elaboración propia.

El análisis realizado mediante Permutation Importance ha posibilitado una evaluación de la importancia relativa de las características del modelo utilizado. Al identificar con claridad las características que contribuyen de manera más significativa a la capacidad predictiva del modelo, este enfoque brinda una guía para la toma de decisiones en el proceso de modelado y análisis de datos.

La fortaleza del análisis de Permutation Importance radica en su capacidad para evaluar el impacto de cada característica individualmente, sin verse afectado por la complejidad del modelo en su totalidad. Esto proporciona una comprensión más profunda sobre cómo cada característica influye en las predicciones del modelo y cuál es su contribución relativa en comparación con otras características.

Además, la naturaleza intuitiva del análisis de Permutation Importance facilita su interpretación y comunicación a diversas audiencias, incluso a aquellas que no son expertas en estadística o aprendizaje automático.

4.6 Despliegue

Para la etapa de despliegue, se llevará a cabo un estudio de caso que se basará en los datos de un estudiante de bachillerato. El objetivo principal es predecir el rendimiento académico en las pruebas universitarias SABER Pro para cada una de las áreas del conocimiento. Posteriormente, se realizará una jerarquización de los resultados, lo que permitirá recomendar los programas de pregrado con mayor probabilidad de éxito para el estudiante. A continuación, se detallan los datos del estudiante:

Variable	Registro
ESTU_TIPODOCUMENTO	TI
ESTU_GENERO	F
FAMI_ESTRATOVIVIENDA	Estrato 3
FAMI_PERSONASHOGAR	3 a 4
FAMI_EDUCACIONMADRE	SI
FAMI_TIENEINTERNET	NO
FAMI_TIENEAUTOMOVIL	NO
COLE_GENERO	MIXTO
COLE_CALENDARIO	A
COLE_BILINGUE	NO
COLE_AREA_UBICACION	URBANO
COLE_JORNADA	UNICA
PUNT_LECTURA_CRITICA	75
PUNT_MATEMATICAS	70
PUNT_C_NATURALES	60
PUNT_SOCIALES_CIUDADANAS	65
PUNT_INGLES	48
estu_area_conocimiento	INGENIERÍA, ARQUITECTURA, URBANISMO Y AFINES
area_conocimiento	IAUA
puntajeGLob_pro_tyt	185

Tabla 11. *Datos de estudio de caso*
Fuente: Elaboración propia.

Los registros proporcionados ofrecen una visión completa de la información tanto del estudiante como de su entorno académico. Se trata de una estudiante de género femenino, cuyo tipo de documento de identidad es TI. Vive en un hogar de estrato 3 con un tamaño del hogar de entre 3 y 4 personas. Destaca el hecho de que su madre posee educación, lo que sugiere un entorno familiar con una influencia educativa positiva. Sin embargo, es relevante mencionar que no cuentan con acceso a internet ni disponen de un automóvil familiar.

En cuanto a la institución educativa, el estudiante asiste a una escuela de género mixto, con un calendario tipo A y ubicación en zona urbana. La jornada escolar es única, lo que significa que experimenta una rutina continua a lo largo del día.

En cuanto al desempeño del estudiante en las pruebas, se destacan los puntajes sólidos obtenidos en lectura crítica (75) y matemáticas (70). Además, ha demostrado un rendimiento consistente en ciencias naturales (60), sociales ciudadanas (65) e inglés (48). Destaca también el interés de la estudiante en el área de conocimiento "INGENIERÍA, ARQUITECTURA, URBANISMO Y AFINES," respaldado por un puntaje global de 185 en la prueba SABER Pro.

4.6.1 Pronóstico del éxito académico

La tabla que se presenta a continuación contiene información relevante acerca de los resultados de la estudiante en la prueba de educación superior SABER Pro, específicamente el puntaje global "puntajeGLob_pro_tyt". Además, se incluyen las predicciones generadas por un modelo analítico y la clasificación de diferentes áreas del conocimiento según sus puntajes y predicciones.

En esta tabla, se observa que la estudiante ha alcanzado un puntaje global de 185, seleccionando así la tercera opción en la jerarquía. El modelo basado en Stacking ha generado una predicción de 169 para su desempeño académico.

Resulta significativo destacar que las áreas de "CIENCIAS SOCIALES Y HUMANAS" y "CIENCIAS DE LA SALUD" muestran las predicciones más altas, ambas con un puntaje de 170. Esto sugiere que la estudiante tiene una probabilidad considerable de destacarse en cualquiera de estas áreas.

No obstante, es esencial recordar que las predicciones de los modelos pueden no ser completamente precisas, y el éxito académico también depende de otros factores individuales, como la dedicación y el interés en el campo de estudio. Por lo tanto, aunque las predicciones ofrecen una perspectiva valiosa, se recomienda considerarlas junto con otros aspectos personales para obtener una evaluación completa del desempeño académico potencial.

Jerarquización	puntajeGLob_pro_tyt	predicciones	estu_area_conocimiento
1	185	174	CIENCIAS SOCIALES Y HUMANAS
2	185	171	MATEMÁTICAS Y CIENCIAS NATURALES
3	**185**	**170**	**INGENIERÍA, ARQUITECTURA, URBANISMO Y AFINES**
4	185	169	CIENCIAS DE LA SALUD
5	185	169	CIENCIAS DE LA EDUCACIÓN
6	185	168	ECONOMÍA, ADMINISTRACIÓN, CONTADURÍA Y AFINES
7	185	168	ARQUITECTURA
8	185	168	AGRONOMÍA, VETERINARIA Y AFINES
9	185	167	BELLAS ARTES

Tabla 12. *Predicción del modelo estudio de caso*

Fuente: Elaboración propia.

Con el objetivo de facilitar el acceso al modelo, se ha optado por compartir los códigos de desarrollo de los modelos y los datos empleados en su creación. Estos recursos se encuentran alojados en la nube y están disponibles para su descarga y utilización a través del siguiente enlace: https://drive.google.com/drive/folders/1-236jsa1gYqLkZQX6nkzhYB_WS4pl6vw?usp=sharing

5 Cronograma

A continuación, ser muestra de forma condensada el cronograma en el que se desarrolló el presente proyecto empresarial

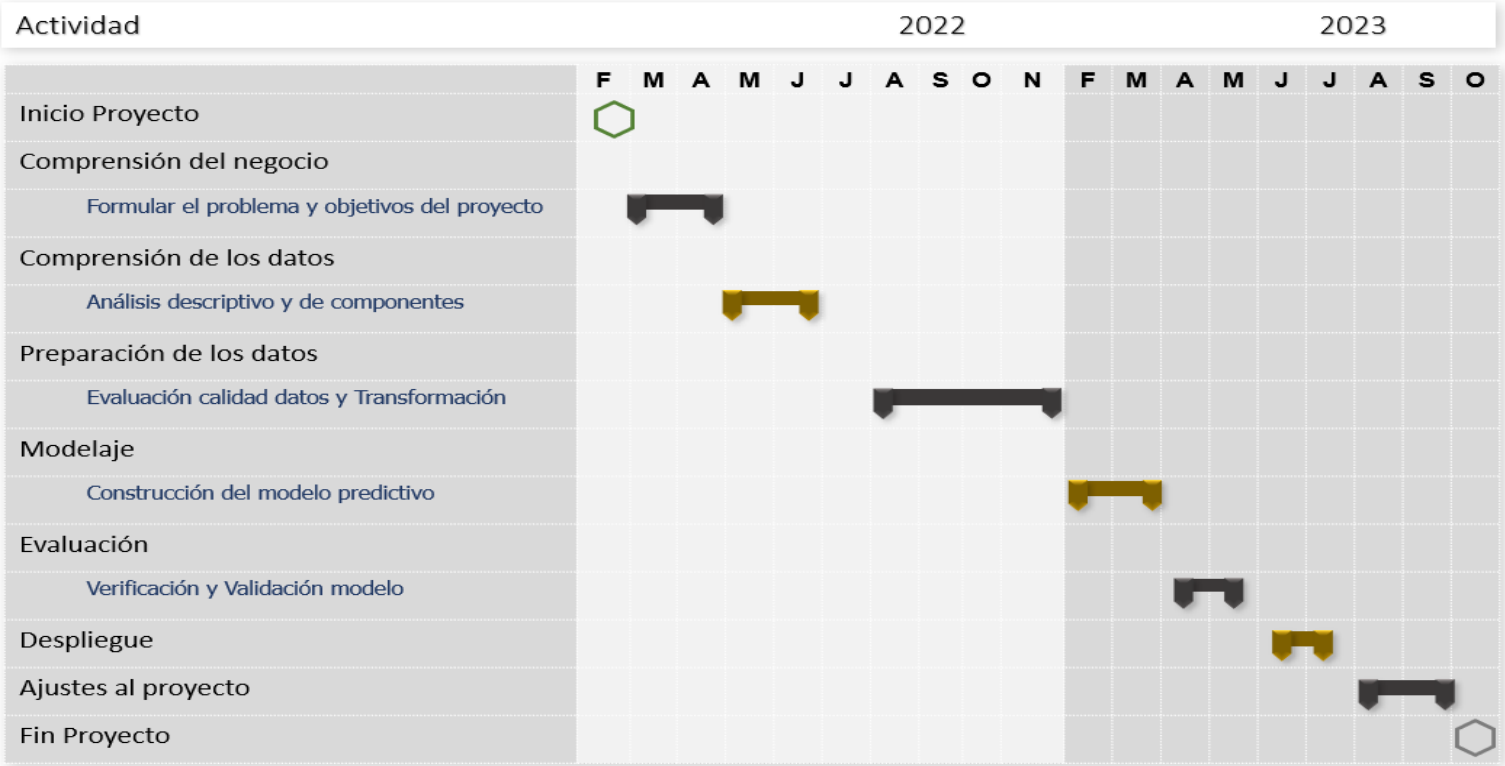


Figura 20. Cronograma del proyecto

Fuente: Elaboración propia.

6 Conclusiones

De acuerdo con los objetivos y alcance del proyecto "Diseño de un modelo predictivo del desempeño académico de estudiantes que ingresen a la educación superior", se pueden extraer las siguientes conclusiones:

- El objetivo general del proyecto, que consistía en desarrollar un modelo analítico para anticipar el desempeño académico de los estudiantes en su educación superior, se ha alcanzado satisfactoriamente. El modelo predictivo elegido por rendimiento y optimización fue Stacking, el cual proporciona una herramienta efectiva para predecir el rendimiento académico de los estudiantes en las áreas del conocimiento a las que aspiran ingresar.
- En cuanto a los objetivos específicos del proyecto, que incluían la identificación de variables relevantes para modelar, hemos podido clasificarlas en cuatro categorías principales: económicas, sociales, educativas y de desempeño académico. Estas categorías abarcan aspectos cruciales que influyen en el rendimiento académico de los estudiantes, desde factores económicos como el estrato de vivienda y la posesión de automóvil, hasta elementos sociales como el género y el nivel educativo de los padres. Las variables relacionadas con la institución educativa, como el tipo de calendario, la naturaleza de la institución y la ubicación, también desempeñan un papel esencial. Por último, los puntajes en lectura crítica, matemáticas, ciencias naturales, ciencias sociales, inglés y el puntaje global SABER Pro y TyT son fundamentales para evaluar el desempeño académico. Esta clasificación de variables proporciona una base sólida para el desarrollo de nuestro modelo analítico, permitiéndonos considerar una amplia gama de factores que influyen en el rendimiento académico de los estudiantes en las áreas de conocimiento de la educación superior.
- La aplicación de una metodología coherente y lógica en la creación de modelos analíticos de predicción ha sido satisfactoria. Esto garantiza la validez y fiabilidad de los modelos desarrollados, lo que es esencial para tomar decisiones informadas.

- El proyecto ha llevado a cabo una evaluación de la eficacia de los modelos predictivos, utilizando como referencia indicadores de precisión clave, entre los que se incluyen el Root Mean Squared Error (RMSE), el Mean Squared Error (MSE) y el Mean Absolute Error (MAE). Esta evaluación integral ha sido fundamental para la selección del modelo más idóneo. En este contexto, el modelo Stacking ha demostrado ser la elección óptima, cumpliendo así con uno de los objetivos específicos del proyecto. La consideración de estas tres métricas proporciona una visión completa y detallada de la capacidad predictiva de los modelos, garantizando una elección informada y precisa para su implementación.
- El alcance del proyecto se ha cumplido plenamente, brindando resultados concretos y valiosos. Los resultados esperados incluyen un modelo analítico completo, una lista de variables relevantes, modelos analíticos de predicción y un informe de evaluación.

7 Recomendaciones

- Construir modelos analíticos predictivos para recomendar áreas de conocimiento para ingreso a áreas técnicas o tecnológicas, considerando las posibles diferencias entre los grupos poblacionales que desean acceder a cada una.

8 Referencias bibliográficas

Banco Mundial. (2017, mayo 17). *Graduarse: Solo la mitad lo logra en América Latina*. World Bank. https://www.bancomundial.org/es/news/feature/2017/05/17/graduating-only-half-of-latin-american-students-manage-to-do-so

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. *SPSS inc*, *9*(13).

Gaurav. (2021, junio 12). An Introduction to Gradient Boosting Decision Trees. *Machine Learning Plus*. https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/

Han, J. (2006). *Data Mining: Concepts and Techniques*. Elsevier.

Hernández, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación McGraw-Hill*.

Kohavi, R., Rothleder, N. J., & Simoudis, E. (2002). Emerging trends in business analytics. *Communications of the ACM*, *45*(8).

Patel, A. (2020, marzo 18). Stacking -Ensemble meta Algorithms for improve predictions. *ML Research Lab*. https://medium.com/ml-research-lab/stacking-ensemble-meta-algorithms-for-improve-predictions-f4b4cf3b9237

¿Qué es KNN? | IBM. (s. f.). *¿Qué es KNN? | IBM*. Recuperado 25 de octubre de 2023, de https://www.ibm.com/mx-es/topics/knn

Ridge Regression Explained, Step by Step. (2021, mayo 23). *Ridge Regression Explained, Step by Step*. Machine Learning Compass. https://machinelearningcompass.com/machine_learning_models/ridge_regression/

Team Dst, D. (2022, enero 25). Random Forest: Bosque aleatorio. Definición y funcionamiento. *Formation Data Science | DataScientest.com*. https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento

Williamson, B. (2018). *Big data en educación: El futuro digital del aprendizaje, la política y la práctica*. Ediciones Morata.

Wu, X., Zhu, X., Wu, G.-Q., & Ding, W. (2013). Data mining with big data. *IEEE transactions on knowledge and data engineering*, *26*(1).

9 Anexos

Anexo A Datos brutos pruebas saber

Anexo B Script construcción modelos

Anexo C Ejecución scripts construcción modelos