Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ciencias Matemáticas
Escuela Profesional de Computación Científica

Modelos predictivos para la detección de riesgo académico en los estudiantes de pregrado de la carrera de computación científica utilizando técnicas de Machine Learning

TESIS

Para optar el Título Profesional de Licenciada en Computación Científica

AUTOR

Eunice Luzdemivida GARCIA CHAMBI

ASESOR

Mg. Luis Javier VÁSQUEZ SERPA

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

https://creativecommons.org/licenses/by-nc-sa/4.0/

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Garcia, E. (2023). *Modelos predictivos para la detección de riesgo académico en los estudiantes de pregrado de la carrera de computación científica utilizando técnicas de Machine Learning*. [Tesis de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Escuela Profesional de Computación Científica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Eunice Luzdemivida Garcia Chambi
Tipo de documento de identidad	DNI
Número de documento de identidad	73856496
URL de ORCID	No aplica
Datos de asesor	
Nombres y apellidos	Luis Javier Vásquez Serpa
Tipo de documento de identidad	DNI
Número de documento de identidad	43389380
URL de ORCID	https://orcid.org/0000-0002-5414-6764
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Martha Hilda Timoteo Sánchez
Tipo de documento	DNI
Número de documento de identidad	25702279
Miembro del jurado 1	
Nombres y apellidos	Frank Duberlee Alvarez Huertas
Tipo de documento	DNI
Número de documento de identidad	45173557
Miembro del jurado 2	
Nombres y apellidos	Luis Javier Vásquez Serpa
Tipo de documento	DNI
Número de documento de identidad	43389380
Datos de investigación	
Línea de investigación	A.3.4.1. Optimización numérica
Grupo de investigación	No aplica

Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	Edificio: Facultad de Ciencias Matemáticas, Universidad Nacional Mayor de San Marcos. País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Av. Venezuela s/n cuadra 34. Lima 1-Perú Ciudad Universitaria. UNMSM. Latitud: -12.060186431587528 Longitud: -77.08213204783175
Año o rango de años en que se realizó la investigación	Mayo 2023 - octubre 2023
URL de disciplinas OCDE	Ciencias de la computación https://purl.org/pe-repo/ocde/ford#1.02.01 Ciencias de la información https://purl.org/pe-repo/ocde/ford#1.02.02

ACTA DE SUSTENTACIÓN DE TESIS PARA LA OBTENCIÓN DEL TÍTULO PROFESIONAL DE LICENCIADO(A) EN COMPUTACIÓN CIENTÍFICA (PROGRAMA DE TITULACIÓN PROFESIONAL 2023)

En la UNMSM – Ciudad Universitaria – Facultad de Ciencias Matemáticas, siendo las 08:45 horas del sábado 28 de octubre del 2023, se reunieron los docentes designados como Miembros del Jurado Evaluador (PROGRAMA DE TITULACIÓN PROFESIONAL 2023): Dra. Martha Hilda Timoteo Sánchez (PRESIDENTA), Dr. Frank Duberlee Alvarez Huertas (MIEMBRO) y el Mg. Luis Javier Vásquez Serpa (MIEMBRO ASESOR), para la sustentación de la Tesis titulada: "**MODELOS PREDICTIVOS PARA LA DETECCIÓN DE RIESGO ACADÉMICO EN LOS ESTUDIANTES DE PREGRADO DE LA CARRERA DE COMPUTACIÓN CIENTÍFICA UTILIZANDO TÉCNICAS DE PREDICCIÓN DE MACHINE LEARNING**", presentado por la señorita **Bachiller EUNICE LUZDEMIVIDA GARCIA CHAMBI**, para optar el Título Profesional de Licenciada en Computación Científica.

Luego de la exposición de la Tesis, la Presidenta invitó a la expositora a dar respuesta a las preguntas formuladas.

Realizada la evaluación correspondiente por los Miembros del Jurado Evaluador, la expositora mereció la aprobación sobresaliente con mención con un calificativo promedio de diecinueve (19)...

A continuación, los Miembros del Jurado Evaluador dan manifiesto que la participante **Bachiller EUNICE LUZDEMIVIDA GARCIA CHAMBI**, en vista de haber aprobado la sustentación de su Tesis, será propuesta para que se le otorgue el Título Profesional de Licenciada en Computación Científica.

Siendo las 09:30 horas se levantó la sesión firmando para constancia la presente Acta.

Dra. Martha Hilda Timoteo Sánchez
PRESIDENTA

Dr. Frank Duberlee Alvarez Huertas
MIEMBRO

Mg. Luis Javier Vásquez Serpa
MIEMBRO ASESOR

CERTIFICADO DE SIMILITUD

Yo …LUIS JAVIER VÁSQUEZ SERPA……. en mi condición de asesor acreditado con la Resolución Decanal N°…001667-2023-D-FCM/UNMSM…….. de la tesis, cuyo título es: "MODELOS PREDICTIVOS PARA LA DETECCIÓN DE RIESGO ACADÉMICO EN LOS ESTUDIANTES DE PREGRADO DE LA CARRERA DE COMPUTACIÓN CIENTÍFICA UTILIZANDO TÉCNICAS DE PREDICCIÓN DE MACHINE LEARNING"

presentado por el bachiller ……EUNICE LUZDEMIVIDA GARCIA CHAMBI

para optar el .Título Profesional de Licenciado en Computación Científica de la Facultad de Ciencias Matemáticas,

CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de ..9.% de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad correspondiente.

Firma del Asesor _____



DNI: 43389380

Nombres y apellidos del asesor:

Luis Javier Vásquez Serpa

Dedicatoria

A mi madre, Carmen Chambi Huillca,
quien ha desempeñado el papel de madre y padre
en mi vida. Tu persistencia y amor incondicional me
han impulsado a superar los obstáculos y a perseguir
mis metas académicas. Esta tesis es el resultado de
la confianza que siempre has tenido en mí.
Mamá, este logro también es tuyo.

Agradecimientos

A mi querida madre, por su apoyo incondicional
y por motivarme a enfrentar este nuevo desafío.

A mi amiga Dayanne, por los momentos compartidos
y por siempre darme ánimos y nunca dudar de mi.

A mi abuela, tíos y hermanos, por su comprensión
y todo el cariño que me brindan día a día.

Gracias a todos ustedes por ser parte
de mi vida y por creer en mí.

Resumen

MODELOS PREDICTIVOS PARA LA DETECCIÓN DE RIESGO ACADÉMICO EN LOS ESTUDIANTES DE PREGRADO DE LA CARRERA DE COMPUTACIÓN CIENTÍFICA UTILIZANDO TÉCNICAS DEL MACHINE LEARNING

En este trabajo se desarrolla modelos predictivos mediante diversas técnicas de aprendizaje automático, utilizando como conjunto de datos los historiales académicos de 241 estudiantes de pregrado de la carrera de Computación Científica de la Universidad Nacional Mayor de San Marcos. El objetivo es determinar si un estudiante de pregrado enfrentará dificultades académicas (riesgo académico) en su próximo semestre en función de su desempeño académico desde que ingreso a la universidad.

Así mismo, se utiliza diferentes métricas de evaluación para comparar los modelos que fueron construidos por medio de 5 técnicas de aprendizaje automático, para luego analizar los resultados obtenidos en cada modelo. Llegando a la conclusión de que existe mas de un modelo adecuado con un exhaustividad (Recall) superior al 80 % para identificar estudiantes en riesgo académico. También se encontró que el promedio del ultimo semestre cursado es una de las características mas significativas para la predicción de estudiantes en riesgo académico.

De esta manera, se logra brindar a la escuela de Computación Científica una herramienta que ayuda a identificar a los estudiantes en riesgo académico para el próximo semestre. Finalmente, se sugiere replicar la metodología desarrollada a las otras carreras de la FCM, y evaluar la posibilidad de implementar estrategias de apoyo educativo para aquellos estudiantes en situación de riesgo académico identificados por los distintos modelos.

Palabras claves: Riesgo académico, modelos predictivos, aprendizaje automático, métricas de evaluación.

Abstract

PREDICTIVE MODELS FOR THE DETECTION OF ACADEMIC RISK IN UNDERGRADUATE STUDENTS IN SCIENTIFIC COMPUTING CAREER USING MACHINE LEARNING TECHNIQUES

In this work, predictive models are developed using various machine learning techniques, using as a data set the academic records of 241 students of the Scientific Computing career at the Universidad Nacional Mayor de San Marcos. The goal is to determine if an undergraduate student will face academic difficulties (academic risk) in their next semester based on their academic performance since entering to the university.

Likewise, different evaluation metrics are used to compare the models that were built using 5 machine learning techniques, and then analyze the results obtained in each model. Reaching the conclusion that there is more than one suitable model with a completeness (Recall) greater than 80 % to identify students at academic risk. It was also found that the average of the last semester completed is one of the most significant characteristics for predicting students at academic risk.

In this way, it is possible to provide the School of Scientific Computing with a tool that helps identify students at academic risk for the next semester. Finally, it is suggested to replicate the methodology developed to the other FCM careers, and evaluate the possibility of implementing educational support strategies for those students at academic risk identified by the different models.

Keywords: Academic risk, predictive models, machine learning, evaluation metrics.

Índice general

Resumen **4**

Abstract **5**

Índice general **7**

Índice de figuras **9**

1. Introducción **10**

2. Marco Teórico **12**
2.1. Antecedentes 12
2.2. Nociones Previas 14
2.2.1. ¿Que es el Machine Learning? 14
2.2.2. Funcionamiento del Machine Learning 14
2.2.3. Tipos de Machine Learning 15
2.2.4. Algoritmos del Machine Learning 17
2.2.5. Métricas de clasificación usadas en Machine Learning 22

3. Metodología **26**
3.1. Planteamiento del problema 26
3.1.1. Objetivos del proyecto 27
3.2. Recopilación de datos 27
3.3. Preparación de los datos 28
3.3.1. Variable Temporal 30
3.3.2. Variables Adicionales 30

4. Implementación Computacional **32**
4.1. Importación de librerías 32
4.2. Extracción de la data 33
4.3. Preparación del data_train y data_test 34
4.4. Construcción de los modelos 35
4.4.1. Modelos de Árbol de decisión 35
4.4.2. Modelos de Regresión Logística 37
4.4.3. Modelos de Random Forest 38
4.4.4. Modelos de K_Vecinos más cercanos 39
4.4.5. Modelos de Naive Bayes 40

5. Resultados **42**

6. Conclusiones y Recomendaciones **44**
6.1. Conclusiones 44
6.2. Recomendaciones 45

Bibliografía **46**

Índice de figuras

1.1. Indicadores de seguimiento del OEI-01 correspondiente al periodo 2019-2023. Fuente: Arbulu Velez, 2023 10

2.1. Entrenamiento de un algoritmo. Fuente: Corporativa, 2023 14
2.2. Funcionamiento del Machine Learning. Fuente: Elaboración propia. . . . 14
2.3. Ejemplo de algoritmo supervisado. Fuente: Alan, 2021 16
2.4. Ejemplo de algoritmo no supervisado. Fuente: Alan, 2021 16
2.5. La función logística o función sigmoide. Fuente: Gonzalez, 2022c 17
2.6. Estructura de un árbol de decisión. Fuente: Gonzalez, 2020 18
2.7. Ejemplo de Random Forest. Fuente: Heras, 2020b 19
2.8. Ejemplo de K Vecinos más Cercanos (k=3). Fuente: Gonzalez, 2022a . . 20
2.9. Matriz de Confusión con un ejemplo de marketing. Fuente: Heras, 2020a 23
2.10. Precisión. Fuente: Heras, 2020a 23
2.11. Exhaustividad (Recall). Fuente: Heras, 2020a 24

3.1. Metodología empleada. Fuente: Elaboración propia. 26
3.2. Semestre Vs Año de Ingreso. Fuente: Elaboración propia. 28
3.3. División de la muestra. Fuente: Elaboración propia. 29
3.4. Código Alumno Vs Estado Asignatura. Fuente: Elaboración propia. . . . 30
3.5. Promedio de notas del semestre 20121. Fuente: Elaboración propia. . . . 31
3.6. Promedio ponderado de notas. Fuente: Elaboración propia. 31

4.1. Lista de librerías a importar 32
4.2. Uso de la librería pandas 33
4.3. Información del dataFrame 33
4.4. Separación de variables predictoras y variables a predecir 34
4.5. Creación de las datas_train y datas_test 34
4.6. Uso de la librerías matplotlib.pyplot y tree 35
4.7. Árbol de decisión con max_depth=4 35
4.8. Entrenamiento de los modelos de árbol de decisión 36
4.9. Predicción de los modelos de árbol de decisión 36
4.10. Reporte de métricas del modelo árbol_riesgo_R 36
4.11. Reporte de métricas del modelo árbol_riesgo_P 37
4.12. Entrenamiento de los modelos de regresión logística 37
4.13. Predicción de los modelos de regresión logística 37
4.14. Reporte de métricas del modelo regresión_riesgo_R 38
4.15. Reporte de métricas del modelo regresión_riesgo_P 38
4.16. Entrenamiento y predicción de los modelos de random forest 38
4.17. Reporte de métricas del modelo bosque_riesgo_R 39

4.18. Reporte de métricas del modelo bosque_riesgo_P 39
4.19. Entrenamiento y predicción de los modelos de K_Vecinos más cercanos . 39
4.20. Reporte de métricas del modelo kvecinos_riesgo_R 40
4.21. Reporte de métricas del modelo kvecinos_riesgo_P 40
4.22. Entrenamiento y predicción de los modelos de Naive Bayes 40
4.23. Reporte de métricas del modelo gnb_riesgo_R 41
4.24. Reporte de métricas del modelo gnb_riesgo_P 41

5.1. Cuadro comparativo de los modelos predictivos 42

Capítulo 1

Introducción

El rezago y la deserción estudiantil son problemáticas recurrentes en instituciones de educación superior, esto se manifiesta en respuesta a que los estudiantes no logran cumplir con los requisitos académicos esperados (estudiantes en riesgo académico).

En la literatura se encuentran estudios que señalan que la integración social y académica es uno de los factores que mas impacta en el proceso de transición y adaptación del estudiante, esto significa que los estudiantes que logran adaptarse (social y académicamente) de forma exitosa en cada semestre, son los que continúan sus estudios y y perseveran hasta alcanzar la graduación.

Por otra parte, los estudiantes que enfrentan dificultades para adaptarse en cada semestre son aquellos que tienden a mostrar un bajo rendimiento académico, y como resultado, se encuentran en una situación de riesgo académico.

En consecuencia, una de las responsabilidades de la institución educativa es abordar las necesidades que puedan surgir en los estudiantes y proporcionar un seguimiento adecuado para identificar a aquellos que podrían estar en riesgo académico. Además, debe considerar la posibilidad de implementar estrategias de apoyo que ayuden a estos estudiantes a mejorar su rendimiento académico.

En la Universidad Nacional Mayor de San Marcos (UNMSM), se encuentra vigente el Plan Estratégico Institucional (PEI) correspondiente al periodo 2019-2023. En este plan se detalla el seguimiento y la evaluación de los diversos objetivos establecidos por la universidad. Uno de estos objetivos es el siguiente: Mejorar la formación académica con calidad para los estudiantes (Arbulu Velez, 2023).

INDICADOR	PARAMETRO DE MEDICION	LINEA BASE		VALOR ACTUAL		LOGRO ESPERADO					VALOR OBTENIDO					AVANCE (%)				
		Año	Valor	Año	Valor	2019	2020	2021	2022	2023	2019	2020	2021	2022	2023	2019	2020	2021	2022	2023
Porcentaje de egresados de pregrado que se encuentren satisfechos con la formación universitaria de pregrado.	Porcentaje	2017	60.60	2018	61.80	63.00	65.00	66.00	67.00	68.00	ND	80.00	27.11	53.09	ND	ND	100	41.08	79.24	ND
Porcentaje de graduados y titulados de los programas de posgrado	Porcentaje	2017	3.20	2018	8.50	11.80	14.48	17.47	20.47	23.49	8.30	2.70	6.67	11.97	ND	70.34	18.65	38.18	58.48	ND

Figura 1.1: Indicadores de seguimiento del OEI-01 correspondiente al periodo 2019-2023. Fuente: Arbulu Velez, 2023

- El primer indicador que se visualiza en la figura 1.1 proporciona información relevante para evaluar la percepción de los egresados sobre si la universidad está preparando de manera adecuada a sus futuros profesionales.
- El segundo indicador que se visualiza en la figura 1.1 proporciona información sobre la cantidad de graduados y titulados que provienen de los programas de postgrado.

Analizando la figura 1.1, se observa la caída del primer indicador del año 2020 al 2021, esto en parte causado por la emergencia sanitaria global originada por el COVID-19; por otra parte se entiende que la universidad trabajo en nuevas propuestas de mejora que ayudaron a levantar el indicador para el año 2022, a pesar de la mejora aun se tiene un desfase con el logro esperado.

La herramienta que propone este proyecto, destinado a identificar estudiantes en riesgo académico por medio de modelos predictivos, podría ser una valiosa contribución a la universidad en el contexto del OEI-01, ya que le permitiría a la universidad elaborar estrategias de apoyo con los estudiantes identificados. Como resultado, estos estudiantes tendrían la oportunidad de mejorar su desempeño académico y continuar con éxito sus estudios hasta la graduación. De esta manera, se estaría colaborando en el cumplimiento de la meta establecida por la universidad.

La estructura de esta tesis es como sigue:

En el capitulo 2, se tiene el marco teórico donde se encuentra la sección Antecedentes, la cual cubre los trabajos que han precedido a este proyecto, luego en la sección Nociones Previas abarcaremos conceptos acerca del aprendizaje automático y las principales métricas de evaluación que existen.

En el capitulo 3, se muestra la metodologıa empleada, la cual esta basada en el planteamiento del problema, la recopilación de datos, la preparación de los datos, la construcción de los modelos, la evaluación de los modelos y el despliegue de los modelos.

En el capitulo 4, se trabaja en Google Colab y se utilizan las técnicas de machine learning para construir los modelos predictivos de cada riesgo académico, adicional se muestra el reporte de métricas de clasificación de cada modelo.

En el capitulo 5, se consolidan y analizan las métricas precisión y exhaustividad de los modelos desarrollados, seguida por una exposición detallada de los resultados del análisis.

Finalmente en el capitulo 6, se abordan las conclusiones derivadas del estudio de investigación, junto con las sugerencias destinadas a potenciar futuros avances en este proyecto.

Capítulo 2

Marco Teórico

2.1. Antecedentes

En esta sección se presenta una base de estudios realizados (libros, tesis de grado, artículos y trabajos presentados) que van de acuerdo con el tema de investigación tratado.

En (Lamas, 2015), se enfatiza la relevancia de investigar el rendimiento académico y ofrece una perspectiva conceptual de los diversos enfoques empleados para comprender el proceso de aprendizaje y los factores que lo afectan. Además, aborda la evaluación de las variables que se consideran influyentes en la explicación del rendimiento académico, buscando así una mayor objetividad en su análisis.

En (Moreno Murcia y col., 2015), se examina la conexión entre el cuidado del profesor, las necesidades psicológicas básicas y la motivación académica en relación con la percepción de competencia académica y el promedio de calificaciones de los alumnos. Los hallazgos de la investigación evidencian una correlación positiva y notable entre todas las variables analizadas. Además, el estudio respalda la relevancia de la interacción entre el docente y el estudiante, sugiriendo que esta interacción puede influir en la percepción de competencia académica del estudiante, lo que a su vez puede aumentar su motivación para aprender y mejorar su desempeño académico.

En (Dorio Alcaraz, 2017), se contextualiza la investigación en la transición académica y social de los estudiantes que ingresan a la educación superior. Sus hallazgos respaldan la influencia de las variables socio-cognitivas en la adaptación de los estudiantes durante los primeros semestres y a lo largo de su proceso de formación. Dentro de su estudio, los alumnos expresaron la importancia de disponer de redes de apoyo tanto en el ámbito social como en el académico para superar los retos académicos y seguir avanzando en sus estudios hasta finalmente lograr su graduación.

En (York y col., 2015), el término “éxito académico” figura como uno de los constructos más recurrentemente empleados en la investigación y evaluación educativa. En su estudio, se realizo un análisis exhaustivo del uso de este concepto en diversos ámbitos académicos. Los resultados de su investigación señalaron que las calificaciones y el promedio de calificaciones destacan como las métricas de éxito académico más frecuentemente utilizadas.

Con el avance del maching learning y otras disciplinas para el análisis de datos educativos, se han desarrollado varios estudios en la literatura que exploran el empleo de la minería de datos y el maching learning en la predicción de rendimiento académico o en las problemáticas que surgen a partir del riesgo académico. Varios de esos estudios se detallan a continuación:

En (Vergaray, 2021), se exploraron modelos predictivos con el propósito de anticipar la deserción académica de estudiantes en la Universidad Nacional Intercultural de la Amazonía, para lo cual se utilizo datos socio económicos y de rendimiento académico. Los resultados obtenidos condujeron a la conclusión de que la implementación de modelos predictivos en entornos universitarios para identificar tempranamente a estudiantes en riesgo de deserción académica es una práctica viable y altamente beneficiosa.

En (Franco y col., 2021), utilizaron para su investigación los datos socio demográficos y resultados de exámenes de admisión de 415 estudiantes de las carreras del área de computación de la Universidad Autónoma de Yucatán, ubicada en México. El propósito principal del estudio consistía en crear modelos predictivos de riesgo académico mediante la aplicación de técnicas de minería de datos educativos. Los resultados obtenidos en esta investigación indican que el algoritmo LMT produjo el modelo más eficiente. Además, sugiere la implementación de herramientas de software destinadas a la identificación temprana de situaciones de riesgo académico y la aplicación oportuna de estrategias de intervención educativa.

En (Bustamante Peña y col., 2021), se plantea como objetivo la investigación y aplicación de una herramienta con el propósito de identificar de manera anticipada a aquellos alumnos que podrían abandonar sus estudios. Este estudio evalúa el uso de modelos de árboles de decisión para evaluar el comportamiento de los alumnos y adopta la metodología CRISP-DM para llevar a cabo el proyecto. Como resultado de esta investigación, proporciona a la Universidad Jorge Tadeo una herramienta que facilita la identificación de estudiantes en riesgo, permitiendo la adopción de medidas preventivas para reducir la tasa de deserción académica.

En (Ramírez & Páez, 2022), se presenta una metodología en su investigación con el propósito de desarrollar modelos predictivos de rendimiento académico utilizando características específicas de estudiantes de ingeniería en México. El estudio involucró a 228 estudiantes inscritos en una universidad pública en dicho país, y se recolectaron datos al comienzo del curso. Se emplearon tres técnicas de aprendizaje automático para la construcción de los modelos predictivos. Además, se destaca que, en el análisis, el promedio actual se identificó como la variable más relevante en la predicción del rendimiento académico de los alumnos participantes en la investigación.

En (Rico Páez, 2022), el propósito consistió en desarrollar modelos progresivos de predicción del rendimiento académico y someterlos a diversas técnicas de aprendizaje automático. En esta investigación se recopilación las calificaciones de actividades académicas de 260 estudiantes universitarios de México. Los modelos fueron construidos en fases sucesivas a lo largo del curso y posteriormente evaluados utilizando la métrica de exactitud en la predicción en un grupo de 112 estudiantes de un curso posterior. Los resultados de este estudio sugieren que este enfoque metodológico puede ser replicado en distintos tipos de cursos. Además, se destacó su flexibilidad en cuanto a la selección de la etapa temporal.

2.2. Nociones Previas

A continuación un breve resumen de algunos conceptos que se emplearan mas adelante en este proyecto.

2.2.1. ¿Que es el Machine Learning?

El Machine Learning o aprendizaje automático constituye un ámbito científico y, de manera más específica, una subdivisión dentro del campo de la inteligencia artificial. Su enfoque radica en permitir que los algoritmos identifiquen patrones recurrentes, conocidos como "patterns", en colecciones de datos. Estos datos abarcan números, palabras, imágenes, estadísticas, entre otros (Team, 2022).

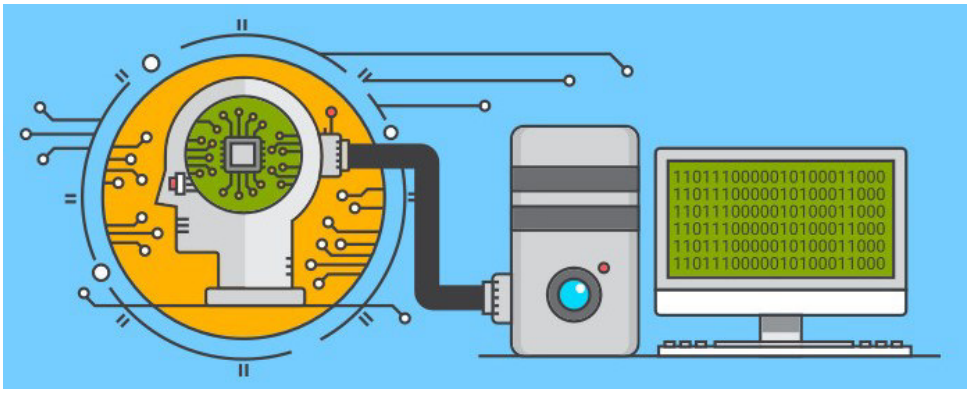

Figura 2.1: Entrenamiento de un algoritmo. Fuente: Corporativa, 2023

Cualquier información que sea susceptible de ser almacenada en formato digital puede desempeñar la función de dato en el ámbito del Machine Learning. Al identificar patrones en dichos datos, los algoritmos adquieren conocimiento y optimizan su capacidad para desempeñar de manera más efectiva una tarea particular.

En síntesis, los algoritmos de Machine Learning se instruyen de manera autónoma para llevar a cabo tareas y efectuar predicciones a partir de información, perfeccionando su eficacia con el paso del tiempo. Tras su entrenamiento, el algoritmo estará capacitado para identificar patrones en nuevos conjuntos de datos.

2.2.2. Funcionamiento del Machine Learning

El proceso de desarrollar un modelo de Machine Learning se divide en cuatro etapas clave, que se detallan a continuación.

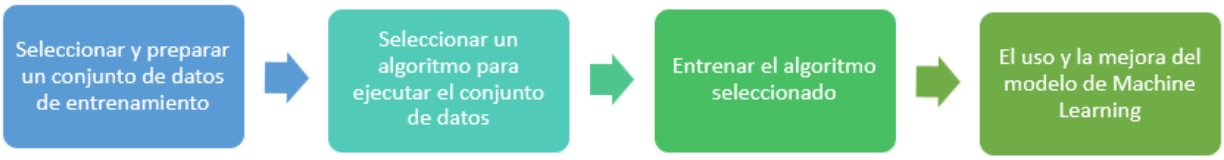

Figura 2.2: Funcionamiento del Machine Learning. Fuente: Elaboración propia.

1.- Seleccionar y preparar un conjunto de datos de entrenamiento.

Estos datos se emplearán para proporcionar información al modelo de aprendizaje automático, permitiendo que adquiera la capacidad de abordar la tarea específica para la cual fue creado (Team, 2022).

Para indicarle al modelo los patrones que debe reconocer se pueden etiquetar los datos, pero también pueden estar sin etiquetar, lo que requiere que el modelo identifique y extraiga por sí mismo las características recurrentes de manera independiente.

En ambas situaciones, resulta escencial llevar a cabo una cuidadosa preparación, estructuración y depuración de los datos, ya que de lo contrario, podría afectarse adversamente el proceso de entrenamiento de un modelo de Machine Learning, con repercusiones en la precisión de sus pronósticos futuros.

2.- Seleccionar un algoritmo para ejecutar el conjunto de datos.

La selección del algoritmo adecuado depende de factores como la naturaleza y el volumen de los datos de entrenamiento disponibles, así como de la complejidad del problema que se pretende resolver.

3.- Entrenar el algoritmo seleccionado.

Este proceso implica iteraciones en las que las variables son procesadas por el algoritmo, y luego se comparan los resultados con los valores que se esperaba obtener. Los "pesos" y el sesgo pueden ser modificados para mejorar la precisión del resultado.

Luego, se repiten las variables de forma sucesiva hasta que el algoritmo logre producir el resultado preciso en la mayoría de los casos. El resultado de este proceso de entrenamiento es lo que se conoce como el modelo de Machine Learning.

4.- El uso y la mejora del modelo de Machine Learning.

Aplicamos el modelo sobre datos adicionales, cuyo origen esta determinado por la naturaleza del problema a abordar. Por ejemplo, en la escuela de Computación Científica se utilizara el modelo de Machine Learning entrenado para detectar a los alumnos en riesgo académico en el próximo semestre.

2.2.3. Tipos de Machine Learning

Identificamos 3 tipos de Machine Learning: el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje reforzado.

Aprendizaje supervisado

El aprendizaje supervisado requiere información de entrada y salida que se encuentre etiquetada durante la etapa de entrenamiento. Frecuentemente, estos datos son clasificados por un profesional de análisis de datos en la etapa de preparación previa a su implementación para el adiestramiento y la evaluación del modelo (Europea, 2022).

La denominación "aprendizaje supervisado" deriva de su necesidad de supervisión humana en al menos alguna fase del proceso.

La gran mayoría de los datos disponibles se presenta en estado crudo, sin etiquetas, y habitualmente, resulta necesario el aporte humano para llevar a cabo una clasificación precisa de los datos.

Veamos un ejemplo, en la figura 2.3 los datos si tienen sus etiquetas. Los patos están claramente identificados como "Patos", y los animales que no son patos también están claramente identificados como "No patos".

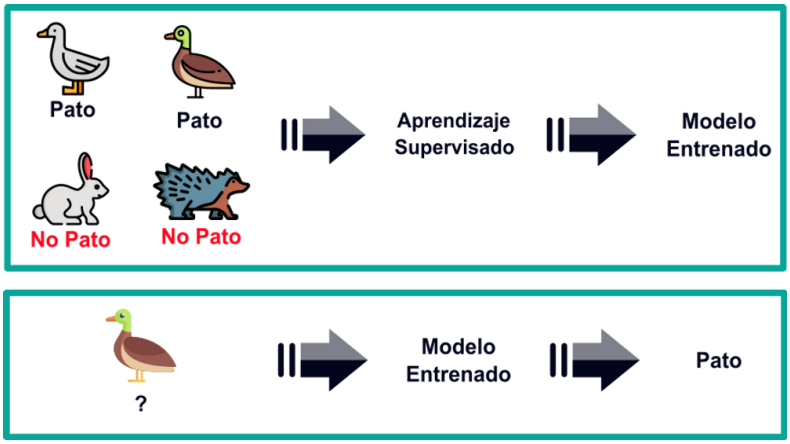

Figura 2.3: Ejemplo de algoritmo supervisado. Fuente: Alan, 2021

Aprendizaje no supervisado

El aprendizaje no supervisado implica el entrenamiento de modelos utilizando datos sin procesar y sin etiquetar, además no requiere una intervención humana tan extensa en comparación con el aprendizaje supervisado.

En este tipo de aprendizaje, la máquina examina los datos en busca de patrones potenciales. Procesa vastas cantidades de información y emplea algoritmos para identificar las características pertinentes requeridas para etiquetar, organizar y categorizar datos en tiempo real sin necesidad de intervención humana.

Veamos un ejemplo, en la figura 2.4 los datos no tienen etiquetas, así que se emplea el aprendizaje no supervisado para juntar los datos con atributos similares.

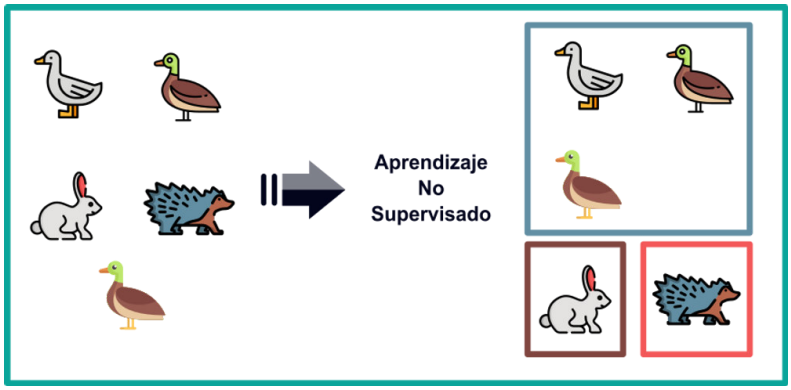

Figura 2.4: Ejemplo de algoritmo no supervisado. Fuente: Alan, 2021

Aprendizaje por refuerzo

Permite que un algoritmo adquiera conocimiento a través de la corrección de sus errores con el fin de alcanzar una meta específica. El algoritmo experimentará una variedad de enfoques distintos en su intento por lograr dicha meta.

Su desempeño se evaluará, y en función de su rendimiento, recibirá incentivos positivos o sanciones con el propósito de motivarlo a seguir o modificar su enfoque. Esta técnica se emplea especialmente para permitirle a la inteligencia artificial superar a los humanos en juegos u otras aplicaciones.

2.2.4. Algoritmos del Machine Learning

A continuación se exponen los algoritmos que utilizaremos para la elaboración de los modelos predictivos de este proyecto.

Regresión Logística

La Regresión Logística, también conocida como Logistic Regression, es un algoritmo de clasificación dentro del ámbito del Machine Learning. Su aplicación se orienta a predecir la probabilidad de una variable dependiente categórica. Este algoritmo, de naturaleza sencilla, es adecuado para abordar tareas de clasificación que son tanto binarias (con dos categorías) como multivariadas (que abarcan más de 2 categorías) (Gonzalez, 2022c).

El modelo de regresión logística binaria es aplicable a diversos desafíos de clasificación, como, por ejemplo, la identificación de correos no deseados, la anticipación de la probabilidad de diabetes, la evaluación de si un cliente específico adquirirá un producto en particular o si optará por la competencia.

La técnica de Regresión Logística toma su nombre de la función que constituye su elemento central, la función logística, también conocida como función Sigmoide. La función Sigmoide se representa como una curva con forma de "S" que tiene la capacidad de mapear cualquier valor real a un rango entre 0 y 1.

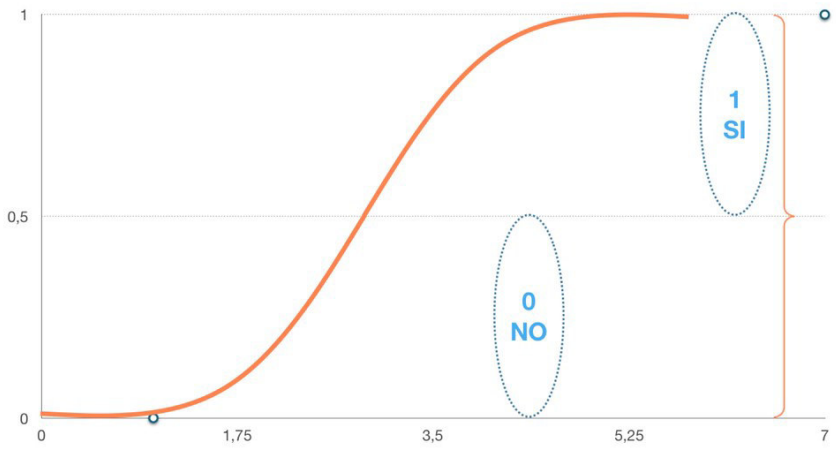

Figura 2.5: La función logística o función sigmoide. Fuente: Gonzalez, 2022c

Cuando la curva se dirija hacia el infinito positivo, la predicción se acercará a 1, mientras que si la curva se extiende hacia el infinito negativo, la predicción se aproximará a 0. Si el valor resultante de la función Sigmoide es mayor de 0.5, podemos etiquetarlo como 1 o "SÍ", y si es menor de 0.5, lo podemos etiquetar como 0 o "NO".

Veamos este algoritmo de manera matemática,

Ecuación de Regresión Lineal: $y = a_1x_1 + a_2x_2 + ... + a_nx_n + b$

Ecuación de función Sigmoide: $p = \frac{1}{1+e^{-y}}$

Si aplicamos la función Sigmoide en la Regresión Lineal nos quedaría de la siguiente forma:

$$p = \frac{1}{1 + e^{-(a_1x_1+a_2x_2+...+a_nx_n+b)}}$$

La regresión logística se utiliza ampliamente debido a su eficiencia y su capacidad para funcionar de manera asequible en producción, a pesar de la existencia de algoritmos más avanzados que demandan recursos computacionales significativos.

Árboles de decisión

Los árboles de decisión representan uno de los algoritmos más ampliamente adoptados en el ámbito del Machine Learning. Esta preferencia se atribuye a su capacidad de ser visualizados de manera comprensible para los seres humanos. Su estructura se asemeja a la de un diagrama de flujo, donde cada nodo interno refleja una característica o atributo, las ramificaciones encarnan reglas de decisión y cada hoja o nodo terminal implica un resultado (Gonzalez, 2020).

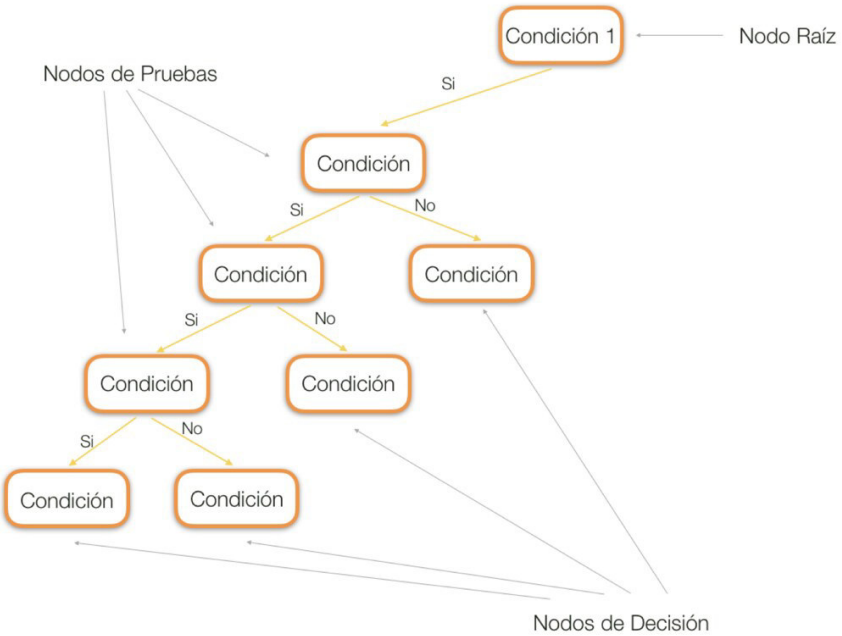

Figura 2.6: Estructura de un árbol de decisión. Fuente: Gonzalez, 2020

La premisa fundamental en cualquier problema relacionado con árboles de decisión se puede resumir de la siguiente manera:

1.- Elegir el atributo óptimo mediante el uso de una medida para la elección de atributos.

2.- Transformar dicho atributo en un nodo de decisión y fragmentar el conjunto de datos en subconjuntos más pequeños.

3.- Iniciar la construcción del árbol, siguiendo este proceso de manera repetitiva para cada atributo hasta que se satisfaga una de las siguientes situaciones:

- Todos los valores de los atributos sean idénticos.
- No queden más atributos por evaluar.
- No haya más casos que analizar.

Este algoritmo se destaca como uno de los escasos algoritmos de aprendizaje automático que ofrecen una visión transparente de cómo se realizan las decisiones a través del algoritmo.

Random Forest

El Random Forest (Bosque Aleatorio) constituye una técnica de aprendizaje automático ampliamente adoptada. En este enfoque, se implementan múltiples algoritmos de árbol de decisiones, y cada árbol examina segmentos diversos de los datos. Consecuentemente, cada árbol se entrena con muestras de datos diferentes que corresponden al mismo problema. A través de esta estrategia de combinar sus resultados, las imprecisiones de algunos árboles se equilibran con los de otros, generando una predicción que presenta una mejor generalización. (Heras, 2020b)

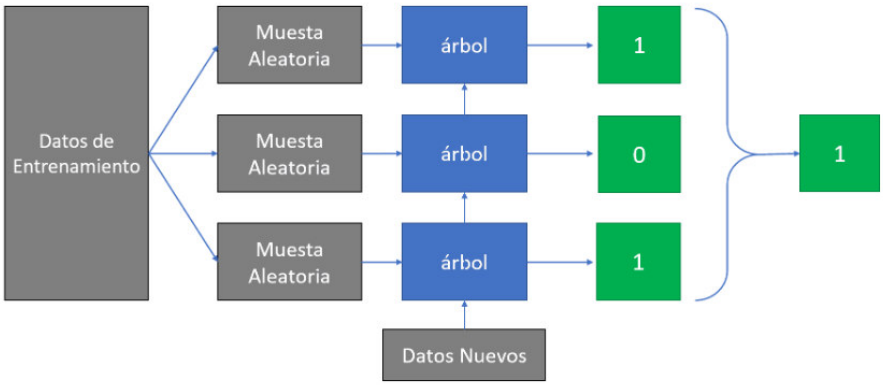

Figura 2.7: Ejemplo de Random Forest. Fuente: Heras, 2020b

Entre las principales ventajas del algoritmo Random Forest tenemos:

- Simplificación del proceso de entrenamiento en comparación con técnicas más complejas, manteniendo un rendimiento similar en términos de precisión.

- Versatilidad en su aplicación, ya que puede utilizarse tanto para clasificación como para predicción. En el primer escenario, cada árbol emite un "voto" a favor de una clase, y el resultado del modelo se determina tomando la clase con la mayoría de "votos" entre todos los árboles. En el segundo caso, el resultado del modelo se obtiene a través del promedio de las salidas de todos los árboles.
- Eficiencia en el desempeño, siendo una de las técnicas más precisas especialmente en conjuntos de datos extensos.
- Capacidad para manejar una gran cantidad de predictores sin excluir ninguno, además de identificar cuáles son los predictores más influyentes. Esto la convierte en una herramienta útil para la reducción de dimensionalidad.
- Mantenimiento de su precisión incluso en situaciones donde se registra una proporción significativa de datos faltantes.

K Vecinos más Cercanos o K-NN

El método de k vecinos más cercanos, ampliamente referido como KNN o K-NN, representa un enfoque de clasificación en el aprendizaje supervisado que carece de parámetros específicos, y se basa en la proximidad para efectuar predicciones o clasificar la pertenencia de un punto de datos individual en un grupo específico (Gonzalez, 2022a).

En este método, se elige un parámetro K (el número de puntos de datos más cercanos al valor que se busca predecir). Es esencial seleccionar un valor de K que se ajuste a la naturaleza de los datos, ya que esto contribuirá a mejorar la precisión de las predicciones.

Veamos un ejemplo, supongamos que el dato con forma de estrella en la figura 2.8 es el que se necesita predecir, puede ser un triangulo o un circulo. Primero, se encuentra los K puntos más cercanos al dato que se desea predecir y después los puntos son agrupados según la elección mayoritaria de sus k vecinos. Cada objeto emite un voto a favor de su categoría, y se considera como predicción la categoría que obtiene la mayoría de votos.

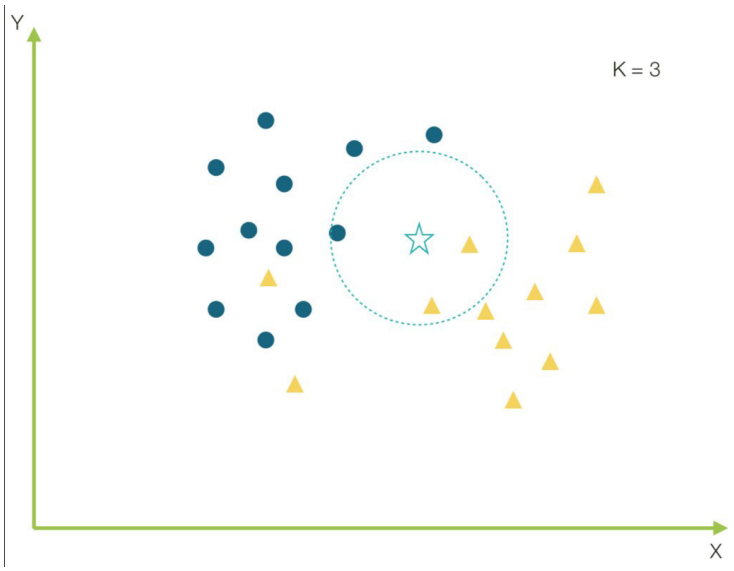

Figura 2.8: Ejemplo de K Vecinos más Cercanos (k=3). Fuente: Gonzalez, 2022a

Para identificar los puntos que son más similares entre sí en un conjunto de datos, el proceso general comienza por calcular la distancia entre estos puntos utilizando diversas métricas de distancia. Una de las opciones comunes es la distancia euclidiana, aunque también existen otras medidas que pueden ser más apropiadas según el contexto, como la distancia de Manhattan y la distancia de Minkowski.

En términos generales, el algoritmo de KNN se lleva a cabo siguiendo estos pasos esenciales:

- Calcular la distancia entre los puntos.
- Identificar los puntos vecinos más cercanos.
- Realizar un conteo de las etiquetas de los vecinos para determinar la etiqueta de interés.

Naive Bayes

El método de Bayes Ingenuo, también conocido como Naive Bayes, representa uno de los algoritmos de clasificación más sencillos y efectivos, basado en el Teorema de Bayes y asumiendo que las variables predictoras son independientes entre sí. La construcción de modelos con Naive Bayes es directa y destaca por su utilidad en conjuntos de datos de gran magnitud (Gonzalez, 2022b).

El método de clasificación de Naive Bayes se fundamenta en la suposición de que la influencia de una característica específica en un entorno no guarda relación con las demás características. Tomemos como ejemplo la evaluación de la idoneidad de un solicitante de préstamo, donde se consideran factores como los ingresos, historial crediticio, transacciones previas, edad y ubicación. A pesar de las posibles interrelaciones entre estas características, el algoritmo considera cada una de ellas de manera independiente. Esta suposición simplifica el proceso de cálculo, lo que le confiere su cualidad "ingenua". Esta asunción se conoce como independencia condicional de clase.

Veamos la representación matemática de Naive Bayes, donde la fórmula del teorema de Bayes es la siguiente:

$$P(h|D) = \frac{P(D|h).P(h)}{P(D)}$$

Donde:

$P(h)$: Representa la probabilidad de que la hipótesis h sea verdadera, sin tener en cuenta los datos. Se denomina como la probabilidad a priori de h.

$P(D)$: Es la probabilidad de los datos, sin considerar la hipótesis. Se denomina probabilidad a priori.

$P(h|D)$: Es la probabilidad de la hipótesis h dada los datos D. Se denomina como la probabilidad posterior.

$P(D|h)$: Representa la probabilidad de que los datos D sean verdaderos si la hipótesis h es cierta. Se denomina probabilidad a posteriori.

Cuando se trabaja con una única característica, el clasificador Naive Bayes determina la probabilidad de un suceso mediante los siguientes pasos:

- Paso 1: Calcula la probabilidad a priori para las etiquetas de clase.
- Paso 2: Establece la probabilidad para cada atributo en relación con cada clase.
- Paso 3: Combina estos valores utilizando el teorema de Bayes para determinar la probabilidad a posteriori.
- Paso 4: Determina qué clase tiene la probabilidad más alta, considerando que la variable de entrada pertenece a la clase con la probabilidad a posteriori más elevada.

2.2.5. Métricas de clasificación usadas en Machine Learning

Se emplean métricas de clasificación para analizar el desempeño de un modelo de clasificación, en función de cómo este asigna etiquetas de clase a los datos de entrada.

Algunas de las métricas de clasificación comunes son precisión, recall (exhaustividad), F1-score, accuracy (exactitud) y la matriz de confusión. A continuación, exploraremos cada una de estas métricas y analizaremos su aplicabilidad en situaciones prácticas.

En un escenario de ejemplo, consideremos la implementación de una estrategia de promoción para una entidad bancaria. La institución bancaria tiene un interés específico en comercializar un producto financiero, un fondo de inversión, entre su base de clientes, con el propósito de generar ingresos a través de las comisiones asociadas.

Una estrategia potencial sería establecer contacto con la totalidad de los clientes de la entidad y presentarles la oportunidad de invertir en este fondo. Sin embargo, este enfoque es considerado ineficaz, ya que la mayoría de los clientes probablemente carecerán de interés en el producto. Por ende, una estrategia más eficaz sería dirigirse a un grupo selecto de clientes, recopilar y analizar sus datos, emplear técnicas de aprendizaje automático y, posteriormente, anticipar cuáles otros clientes podrían tener una mayor probabilidad de aceptar la propuesta del banco (Heras, 2020a).

Dado que prácticamente ningún modelo de aprendizaje automático es impecable, al momento de evaluar el rendimiento de nuestro modelo nos encontraremos con diferentes escenarios que se describen en la matriz de confusión.

Matriz de confusión

La “matriz de confusión” es conocida por su capacidad para facilitar la identificación de las áreas donde el sistema está cometiendo errores al confundir dos categorías.

Utilicemos el ejemplo de marketing descrito lineas arriba para generar la matriz de confusión, en este ejemplo, habremos contactado a 100 clientes, de los cuales 80 expresan desinterés, mientras que los 20 restantes manifiestan interés.

- 0: no está interesado
- 1: sí está interesado

		predicción	
		0	1
realidad	0	70	10
	1	15	5

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Figura 2.9: Matriz de Confusión con un ejemplo de marketing. Fuente: Heras, 2020a

En la matriz de confusión que se encuentra a la izquierda, se encuentran registrados los valores correspondientes a este ejemplo, mientras que en la matriz ubicada a la derecha, se emplean los términos genéricos de acuerdo con la nomenclatura en inglés: False Positive [FP], False Negative [FN], True Negative [TN], True Positive [TP].

Es evidente que, en este caso particular, nuestro modelo no presenta un desempeño óptimo, aunque cabe señalar que la percepción de su desempeño puede variar dependiendo de la métrica que se utilice para la evaluación.

Precisión

Mediante esta medida, se puede evaluar la eficacia del modelo de aprendizaje automático en contextos de clasificación. En el ejemplo citado, la precisión responde a la interrogante ¿qué porcentaje de los clientes que contactemos gracias al modelo realmente mostrarán interés?

Para determinar la precisión, empleamos la siguiente fórmula:

$$Precision = \frac{TP}{TP + FP}$$

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Figura 2.10: Precisión. Fuente: Heras, 2020a

Colocando la información de la matriz de confusión, se obtiene que:

$$Precision = \frac{TP}{TP + FP} = \frac{5}{5 + 10} = 0{,}33$$

Es decir, que aproximadamente sólo un 33 % de los clientes a los que contactemos según el modelo mostrarán un interés real. Esto implica que el modelo en el ejemplo cometerá error en aproximadamente 66 % de las veces cuando pronostique si un cliente mostrará interés.

Recall (exhaustividad)

Esta medida nos proporciona información acerca de la medida en que el modelo de aprendizaje automático es competente en la identificación. En el ejemplo citado, la exhaustividad responde a la interrogante ¿qué porcentaje de los clientes que realmente mostraron interés podemos identificar a travez del modelo?

Para determinar la exhaustividad, empleamos la siguiente fórmula:

$$Recall = \frac{TP}{TP + FN}$$

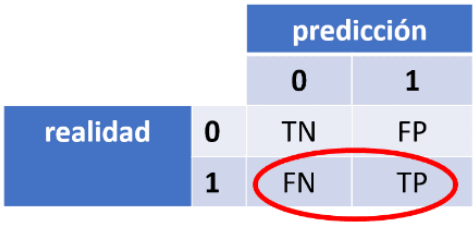

Figura 2.11: Exhaustividad (Recall). Fuente: Heras, 2020a

Colocando la información de la matriz de confusión, se obtiene que:

$$Precision = \frac{TP}{TP + FN} = \frac{5}{5 + 15} = 0{,}25$$

Es decir, el modelo únicamente puede detectar al 25 % de los clientes que podrían estar dispuestos a comprar el producto. En términos más concretos, esto implica que el modelo en cuestión solo logra identificar a uno de cada cuatro clientes que realmente aceptarían la propuesta.

F1-score

Esta medida se emplea con el propósito de fusionar las evaluaciones de precisión y recall en una única medida. Esta estrategia resulta conveniente dado que simplifica la comparación del desempeño conjunto de la precisión y el recall entre diversas soluciones.

Para calcular el F1-score, usaremos la media armónica entre la precisión y el recall:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Combinando la precisión y el recall del ejemplo citado, en F1-score nos quedaría:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot \frac{0{,}33 \cdot 0{,}25}{0{,}33 + 0{,}25} = 0{,}28$$

El valor F1-score se basa en la suposición de que la importancia asignada a la precisión y el recall es igual. Sin embargo, esta consideración no es universal para todos los problemas.

Por ejemplo, en el caso de que necesitemos prever si existe la posibilidad de que un fragmento de desechos espaciales colisione con un satélite, podría otorgarse mayor valor a la exhaustividad, a expensas de una precisión muy baja.

Accuracy (exactitud)

Esta medida evalúa el porcentaje de casos en los que el modelo ha acertado. A pesar de ser una de las métricas más comunes y populares, se sugiere evitar usarla. La complicación con la exactitud radica en su potencial para inducir engaño, es decir, puede generar la impresión de que un modelo deficiente (como el ilustrado en el ejemplo) posee un desempeño considerablemente superior al real.

Para calcular el accuracy (exactitud) usaremos la siguiente fórmula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

El accuracy del ejemplo citado es:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{5 + 70}{5 + 70 + 10 + 15} = 0{,}75$$

Es decir, el modelo tiene una taza de acierto del 75 %. Sin embargo, es evidente que la exactitud es una medida bastante engañosa.

Nota: La métrica de exactitud (accuracy) no presenta un rendimiento óptimo cuando las clases no están balanceadas, tal como ocurre en este ejemplo. Dado que la gran mayoría de los clientes no muestra interés en la propuesta, es relativamente sencillo obtener aciertos al predecir que no estarán interesados. En situaciones que involucran clases desequilibradas, es preferible emplear medidas como la precisión, el recall y el valor F1-score. Estas métricas ofrecen una percepción más precisa sobre la eficacia del modelo.

Capítulo 3

Metodología

En este capitulo se muestra la metodología empleada en el proyecto, la cual esta basada en el planteamiento del problema, la recopilación de datos, la preparación de los datos, la construcción de los modelos, la evaluación de los modelos y el despliegue de los modelos.

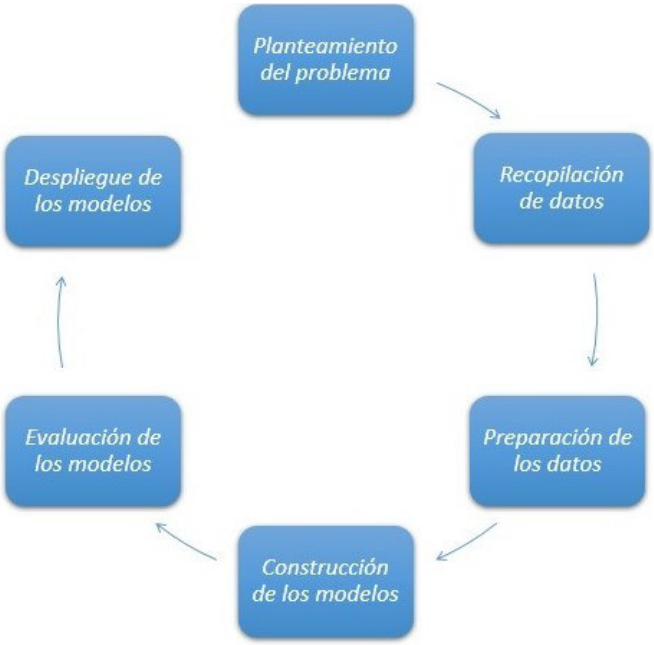

Figura 3.1: Metodología empleada. Fuente: Elaboración propia.

3.1. Planteamiento del problema

Uno de los problemas comunes que enfrentan las instituciones de educación superior, incluyendo la UNMSM, es la presencia de alumnos con bajo rendimiento académico. Esto puede llevar a que los alumnos experimenten atrasos en su progreso académico (rezago) e incluso a que abandonen sus estudios (deserción), lo que a su vez les impide completar satisfactoriamente su carrera universitaria de pregrado y graduarse.

A pesar de que en los últimos años se han observado incrementos en los avances de los indicadores relacionados con el objetivo de "mejorar la formación académica con calidad para los estudiantes"(figura 1.1, uno de los objetivos del PEI (Plan Estratégico Institucional), todavía no se ha logrado alcanzar la meta establecida. (Ver Arbulu Velez, 2023)

La elaboración de modelos predictivos destinados a identificar estudiantes en riesgo académico podría ofrecer a la UNMSM una ventaja significativa, ya que le permitiría la implementación de estrategias novedosas de intervención con aquellos estudiantes identificados por estos modelos. De esta manera, se contribuiría con el logro de la meta establecida por la universidad.

3.1.1. Objetivos del proyecto

Objetivo General:

Desarrollar modelos de predicción eficientes mediante la aplicación de diversas técnicas de machine learning con el proposito de detectar a estudiantes de pregrado en situación de riesgo académico.

Objetivos Específicos:

- OE1: Construir modelos predictivos de riesgo académico utilizando los historiales académicos de los estudiantes, datos extraídos del SUM.
- OE2: Analizar las métricas de precisión y exhaustividad (recall) de cada modelo construido.
- OE3: Elegir a los modelos mas adecuados para la detección de estudiantes en riesgo académico.

3.2. Recopilación de datos

Dado que este proyecto esta orientado a los estudiantes de pregrado de la carrera de Computación Científica en riesgo académico, se requirió el apoyo de la escuela y la oficina de matricula de la Facultad de Ciencias Matemáticas, quien facilito los registros académicos de los estudiantes desde el 2004 hasta el 2013.

Los datos fueron entregados por el área encargada mediante un archivo Excel, el cual contenía las columnas escuela, semestre, código alumno, plan de estudios, ciclo, código asignatura, asignatura, calificación y crédito.

En total se obtuvieron 16,383 instancias que correspondían a 839 alumnos, donde cada instancia contaba con 9 atributos (Ver tabla 3.1)

Variable	Descripción
Escuela	Código de la Escuela
Semestre	Periodo en que el estudiante tomo el curso
Código Alumno	Código del estudiante
Plan de estudios	Plan de estudios al que pertenece el curso
Ciclo	Ciclo al que pertenece el curso
Código asignatura	Código del curso
Asignatura	Nombre del curso
Calificación	Nota obtenida en el curso
Crédito	Numero de créditos del curso

Tabla 3.1: Descripción de las variables iniciales.

3.3. Preparación de los datos

Después de recopilar los datos debemos pasar a la etapa de preparación de datos, que aunque puede ser un proceso desafiante y que consume tiempo, es esencial, ya que sienta las bases para la creación de modelos predictivos de riesgo académico.

El análisis y la preparación de los datos se realizo en Excel, haciendo uso de las diversas funciones disponibles para la edición de los datos. De los 16,383 registros de la base, fueron eliminadas manualmente 59 registros, dado que presentaban información incompleta (o errónea).

	Año Ingreso										
Semestre	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Total
20041	284										284
20042	211										211
20050	35										35
20051	204	376									580
20052	137	268									405
20060	22	31									53
20061	98	243	334								675
20062	97	215	235								547
20070	17	43	38								98
20071	91	213	207	302							813
20072	92	201	169	259							721
20080	4	19	36	45							104
20081	80	155	153	223	317						928
20082	75	154	146	180	251						806
20090	7	12	31	46	32						128
20091	76	136	128	194	211	312					1057
20092	61	132	122	168	164	220					867
20100	8	18	21	26	41	33					147
20101	59	114	111	169	169	176	286				1084
20102	56	98	98	143	139	147	193				874
20110	10	15	18	33	32	34	41				183
20111	41	84	97	137	146	157	164	301			1127
20112	38	52	82	126	132	112	120	215			877
20120	4	15	18	26	16	26	40	29			174
20121	31	49	56	103	148	132	132	162	329		1142
20122	14	33	46	97	129	98	100	136	251		904
20130	2	3	6	9	21	15	25	25	49		155
20131	10	26	29	69	120	92	105	119	201	288	1059
20132	10	21	23	73	88	71					286
Total	**1874**	**2726**	**2204**	**2428**	**2156**	**1625**	**1206**	**987**	**830**	**288**	**16324**

Figura 3.2: Semestre Vs Año de Ingreso. Fuente: Elaboración propia.

Como se visualiza en la figura 3.2, se selecciona como muestra a los registros que contienen la información académica de los estudiantes que ingresaron a partir del año 2008 hasta el 2012 (5 años), la cual consta de 5873 registros.

Esta selección de muestra se considera relevante ya que deseamos que los modelos predictivos puedan ser entrenados con los datos mas recientes que se tengan (en este caso los últimos 5 años), también se observa que los ingresantes del año 2013 no están siendo considerados dentro de la muestra, esto debido a que solo tendríamos la información académica de su primer semestre y esto podría influir en la calidad de nuestros datos de entrenamiento.

Una vez elegida la muestra, se debe evaluar que atributos o variables se van a considerar para el dataset final (de donde se obtiene el data_train y el data_test). Para este problema consideraremos como variables independientes al Prom_P, Prom_US, Cursos_M, Cursos_R y Cursos_A, y como variables dependientes o categóricas consideraremos al Riesgo_R y al Riesgo_P.

Dado que estas ultimas variables no existen en nuestra data inicial (Ver figura 3.1), debemos construirlas y para ello dividiremos nuestra muestra en 2 partes (Ver figura 3.3).

	Año Ingreso					
Semestre	2008	2009	2010	2011	2012	Total
20081	317					317
20082	251					251
20090	32					32
20091	211	312				523
20092	164	220				384
20100	41	33				74
20101	169	176	286			631
20102	139	147	193			479
20110	32	34	41			107
20111	146	157	164	301		768
20112	132	112	120	215		579
20120	16	26	40	29		111
20121	148	132	132	162	329	903
20122	129	98	100	136	251	714
Total	**1927**	**1447**	**1076**	**843**	**580**	**5873**

Muestra I

Muestra II

Figura 3.3: División de la muestra. Fuente: Elaboración propia.

- La muestra I, contiene a los semestres a partir del 20081 hasta el 20121, con los cuales crearemos nuestras variables independientes.
- La muestra II, contiene al semestre 20122, con el cual crearemos a nuestras variables dependientes o categóricas.

Lo que se busca es que mediante los datos académicos que tenemos del alumno desde su ingreso hasta el semestre actual (semestre 20081 al 20121 - Muestra I), identifiquemos si tendrá riesgo académico (Riesgo_R o Riesgo_P) en su próximo semestre (semestre 20122 - Muestra II).

Por medio de las variables iniciales, la creación de una variable temporal, algunas tablas y formulas disponibles en Excel, se crearan las variables que necesitamos con la muestra I y la muestra II.

3.3.1. Variable Temporal

La variable temporal nos ayudara en la creación de las variables adicionales, las cuales serán incluidas en el dataset final. Para este caso nuestra variable temporal creada es "Estado Asignatura".

Variable	Descripción
Estado Asignatura	Nota del Curso ≥ 10.5 = "Aprobado", Nota del Curso < 10.5 = "Desaprobado"

Tabla 3.2: Descripción de la variable temporal.

3.3.2. Variables Adicionales

Variables adicionales de la Muestra I

Las 5 variables descritas en la tabla 3.3 son creadas a partir de las variables iniciales, la variable temporal y el apoyo de algunas formulas y tablas de Excel.

Variable	Descripción
Prom_P	Promedio ponderado
Prom_US	Promedio del ultimo semestre
Cursos_M	Cantidad de cursos matriculados
Cursos_R	Cantidad de cursos reprobados
Cursos_A	Cantidad de cursos aprobados

Tabla 3.3: Descripción de las variables adicionales de la Muestra I

- Mediante la tabla de Excel "Código Alumno Vs Estado Asignatura"se crean las variables adicionales Cursos_A, Cursos_R y Cursos_M (debido al tamaño de la tabla, en la figura 3.4 solo se observan las primeras 6 filas).

Codigo Alumno	Estado Asignatura: Aprobado	Reprobado	Total
08140086	1	9	10
08140087	34	2	36
08140088	6	22	28
08140090		4	4
08140091	16	18	34
08140092		8	8

Figura 3.4: Código Alumno Vs Estado Asignatura. Fuente: Elaboración propia.

- Mediante la tabla de Excel "Promedio de notas del semestre 20121"se crea la variable adicional Prom_US (debido al tamaño de la tabla, en la figura 3.5 solo se observan las primeras 6 filas).

Codigo Alumno	Cantidad de Asignaturas	Suma de las Calificaciones	Promedio Calificaciones
08140087	3	42	14
08140088	3	3	1
08140091	4	5	1.25
08140094	4	43	10.75
08140095	4	55	13.75
08140096	4	61	15.25

Figura 3.5: Promedio de notas del semestre 20121. Fuente: Elaboración propia.

- Mediante la tabla de Excel "Promedio Ponderado de notas"se crea la variable adicional Prom_P (debido al tamaño de la tabla, en la figura 3.6 solo se observan las primeras 6 filas).

Codigo Alumno	Cantidad de Asignaturas	Suma de las Calificaciones	Promedio Calificaciones
08140086	10	51	5.1
08140087	36	497	13.80555556
08140088	28	138	4.928571429
08140090	4	0	0
08140091	34	275	8.088235294
08140092	8	25	3.125

Figura 3.6: Promedio ponderado de notas. Fuente: Elaboración propia.

Variables adicionales de la Muestra II

Las variables que se agregaron a la muestra II son Riesgo_R y Riesgo_P, las cuales fueron creadas de forma similar a las variables adicionales de la muestra I.

Variable	Descripción
Riesgo_R	Cursos_R $>$ 1 = "SI", Cursos_R $\leq$ 1 = "NO"
Riesgo_P	Prom_P $<$ 11 = "SI", Prom_P $\geq$ 11 = "NO"

Tabla 3.4: Descripción de las variables adicionales de la muestra II

Los algoritmos de ML que utilizamos para la clasificación y la generación de modelos predictivos requieren de variables categóricas (clases), para este estudio las variables categóricas a utilizar son Riesgo_R y Riesgo_P.

Finalmente, consolidamos las variables adicionales de la muestra I y la muestra II con la variable código alumno (variable inicial) en un solo archivo (sin duplicados), la cual contiene 241 filas (cada fila pertenece a 1 estudiante) y 8 columnas (las variables elegidas), este dataset contiene los datos preparados que nos servirán para construir los modelos predictivos de riesgo académico en el siguiente capitulo.

Capítulo 4

Implementación Computacional

En este capitulo implementaremos las técnicas de machine learning para construir los modelos predictivos, considerando el conjunto de variables elegidos previamente. Por cada técnica utilizada y por cada variable categórica (Riesgo_R y Riesgo_P) se obtiene un modelo predictivo, la cual sera evaluada utilizando las métricas de clasificación.

Para la construcción y evaluación de los modelos predictivos se hace uso de Google Colab, la cual permite escribir y ejecutar códigos de Python. Antes de implementar las técnicas de machine learning en Google Colab, debemos importar las librerías a utilizar, extraer la Data de nuestro archivo consolidado con las variables elegidas y preparar el Data_train y el Data_test de cada variable categórica (Riesgo_R y Riesgo_P), para después proceder con la creación de los modelos.

4.1. Importación de librerías

Mediante el siguiente código se importaran todas las librerías de Python que necesitamos para el tratamiento de datos, gráficos, construcción de modelos y evaluación.

```
# Tratamiento de datos
# ==============================================================================
import pandas as pd
import numpy as np

# Gráficos
# ==============================================================================
import matplotlib.pyplot as plt
from sklearn import tree

# Construcción de los modelos y evaluación
# ==============================================================================
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

Figura 4.1: Lista de librerías a importar

4.2. Extracción de la data

Dado que nuestras variables elegidas previamente están consolidadas dentro de un archivo Excel, es necesario llevar este archivo al notebook en el cual estamos trabajando y leer estos datos con apoyo de la librería pandas, la cual fue importada lineas arriba.

```
datos=pd.read_excel("/content/DataSet.xlsx",header=0)
datos.head()
```

	codigo alumno	Prom_P	Prom_US	Cursos_M	Cursos_R	Cursos_A	Riesgo_R	Riesgo_P
0	8140087	13.81	14.00	36	2	34	NO	NO
1	8140088	4.93	1.00	28	22	6	SI	SI
2	8140091	8.09	1.25	34	18	16	NO	SI
3	8140094	11.11	10.75	37	8	29	NO	NO
4	8140095	12.82	13.75	37	2	35	NO	NO

Figura 4.2: Uso de la librería pandas

En la figura 4.2 se puede visualizar que no existen valores nulos en las 5 primeras filas de nuestro dataframe “datos”, esto gracias al comando datos.head(). Para asegurarnos de que no existan valores nulos en las filas que no podemos visualizar, ejecutaremos el siguiente comando.

```
datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241 entries, 0 to 240
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   codigo alumno  241 non-null    int64
 1   Prom_P         241 non-null    float64
 2   Prom_US        241 non-null    float64
 3   Cursos_M       241 non-null    int64
 4   Cursos_R       241 non-null    int64
 5   Cursos_A       241 non-null    int64
 6   Riesgo_R       241 non-null    object
 7   Riesgo_P       241 non-null    object
dtypes: float64(2), int64(4), object(2)
memory usage: 15.2+ KB
```

Figura 4.3: Información del dataFrame

Según el informe que nos brinda el comando datos.info() podemos validar de que no existen valores nulos dentro de nuestro dataframe “datos”.

4.3. Preparación del data_train y data_test

Primero se separan las columnas con las variables predictoras de las columnas con las variables a predecir, esto con la ayuda del comando datos.iloc como se visualiza en la figura.

```
#Variables predictoras
X=datos.iloc[:,1:6]

#Variables a predecir
Y1=datos.iloc[:,6]  #pertenece a la variable Riesgo R
Y2=datos.iloc[:,7]  #pertenece a la variable Riesgo P

X.head() #Mostramos las 5 primeras filas
```

	Prom_P	Prom_US	Cursos_M	Cursos_R	Cursos_A
0	13.81	14.00	36	2	34
1	4.93	1.00	28	22	6
2	8.09	1.25	34	18	16
3	11.11	10.75	37	8	29
4	12.82	13.75	37	2	35

Figura 4.4: Separación de variables predictoras y variables a predecir

Mediante el comando train_test_split dividiremos los dataframes X, Y1 y Y2 en subconjuntos de datos de entrenamiento y prueba. Colocando el parámetro train_size=0.75 le estamos indicando al comando que tanto para el subconjunto X1_train como Y1_train debe tomar el 75 % de los datos de X y Y1 respectivamente. Con el parámetro random_size=0 le indicamos que cada vez que sea llamado reproduzca los mismos subconjuntos.

```
# X1_train, X1_test, Y1_train, Y1_test para la variable Riesgo R
# X2_train, X2_test, Y2_train, Y2_test para la variable Riesgo P

X1_train, X1_test, Y1_train, Y1_test = train_test_split(X, Y1, train_size=0.75,random_state=0)
X2_train, X2_test, Y2_train, Y2_test = train_test_split(X, Y2, train_size=0.75,random_state=0)

X1_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 180 entries, 227 to 172
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Prom_P    180 non-null    float64
 1   Prom_US   180 non-null    float64
 2   Cursos_M  180 non-null    int64
 3   Cursos_R  180 non-null    int64
 4   Cursos_A  180 non-null    int64
dtypes: float64(2), int64(3)
memory usage: 8.4 KB
```

Figura 4.5: Creación de las datas_train y datas_test

4.4. Construcción de los modelos

Una vez creados los subconjuntos de datos de entrenamiento y prueba para la variable categórica Riesgo_R y la variable categórica Riesgo_P, pasamos a llamar a los algoritmos de machine learning, los cuales fueron importados previamente, y a entrenar cada uno de los modelos con sus respectivas datas de entrenamiento.

4.4.1. Modelos de Árbol de decisión

Con el fin de visualizar cual es la variable de mayor relevancia (nodo raíz) en el proceso de clasificación de riesgo académico, primero llamaremos al algoritmo de árbol de decisión utilizando el parámetro max_depth=4, el cual creara un árbol que tendrá como máximo 4 niveles de profundidad y luego procederemos a graficarlo con ayuda del siguiente comando.

```
fig = plt.figure(figsize=(25,20)) #dimensiones del grafico

tree.plot_tree(arbol_riesgo_R,feature_names=list(X.columns.values),class_names=list(Y1.values), filled=True)

plt.show()
```

Figura 4.6: Uso de la librerías matplotlib.pyplot y tree

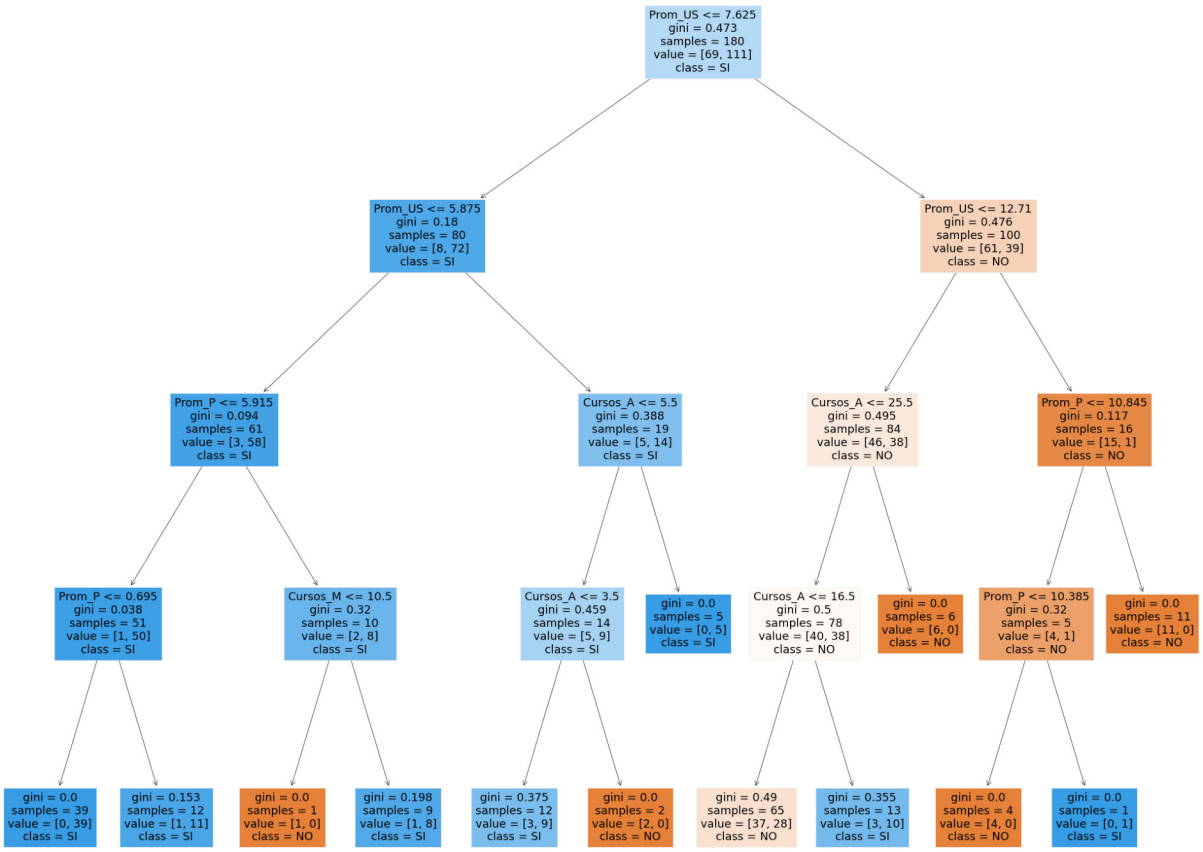

Figura 4.7: Árbol de decisión con max_depth=4

En la figura 4.7, podemos ver cuales son las decisiones internas que toma en cuenta el modelo de árbol de decisión al momento de clasificar si un estudiante tiene riesgo académico o no, adicional a ello se observa que el "promedio del ultimo semestre" es la variable o atributo que el modelo considera de mayor relevancia al momento de predecir.

Después de analizar la figura 4.7, se crean los 2 modelos de arboles de decisión sin colocar la cantidad máxima de niveles de profundidad a tener y se procede a entrenar con las datas de entrenamiento respectivas.

```
#Llamamos al constuctor del arbol de decision
arbol = DecisionTreeClassifier()

#Entrenamos los modelos
arbol_riesgo_R = arbol.fit(X1_train,Y1_train)
arbol_riesgo_P = arbol.fit(X2_train,Y2_train)
```

Figura 4.8: Entrenamiento de los modelos de árbol de decisión

Una vez creados los 2 modelos (uno para cada riesgo) se procede a predecir utilizando las datas de prueba respectivas.

```
#Utilizamos nuestro modelo para predecir los valores del dataset test
Y1_pred=arbol_riesgo_R.predict(X1_test)
Y2_pred=arbol_riesgo_P.predict(X2_test)

Y2_pred
```

```
array(['SI', 'SI', 'SI', 'SI', 'NO', 'SI', 'SI', 'SI', 'SI', 'SI', 'NO',
       'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'NO', 'SI', 'SI', 'NO', 'NO',
       'SI', 'NO', 'NO', 'NO', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'NO',
       'NO', 'NO', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI',
       'NO', 'NO', 'NO', 'SI', 'SI', 'SI', 'NO', 'SI', 'SI', 'NO', 'SI',
       'NO', 'NO', 'SI', 'SI', 'NO', 'SI'], dtype=object)
```

Figura 4.9: Predicción de los modelos de árbol de decisión

Luego de obtener las predicciones Y1_pred y Y2_pred, utilizaremos el comando classification_report para visualizar el reporte de métricas del modelo árbol_riesgo_R y el modelo árbol_riesgo_P.

```
report_1= classification_report(Y1_test, Y1_pred)

print(report_1)
```

```
              precision    recall  f1-score   support

          NO       0.90      0.55      0.68        33
          SI       0.63      0.93      0.75        28

    accuracy                           0.72        61
   macro avg       0.77      0.74      0.72        61
weighted avg       0.78      0.72      0.71        61
```

Figura 4.10: Reporte de métricas del modelo árbol_riesgo_R

```
report_2= classification_report(Y2_test, Y2_pred)

print(report_2)
```

```
              precision    recall  f1-score   support

          NO       0.70      0.78      0.74        18
          SI       0.90      0.86      0.88        43

    accuracy                           0.84        61
   macro avg       0.80      0.82      0.81        61
weighted avg       0.84      0.84      0.84        61
```

Figura 4.11: Reporte de métricas del modelo árbol_riesgo_P

4.4.2. Modelos de Regresión Logística

Para crear el modelo de cada riesgo a través de la técnica de Regresión logística, primero debemos llamar al algoritmo y luego proceder a entrenar cada modelo con su respectiva data de entrenamiento.

```
#Llamamos al algoritmo de regresion logistica
regresion = LogisticRegression()

#Entrenamos los modelos
regresion_riesgo_R = regresion.fit(X1_train,Y1_train)
regresion_riesgo_P = regresion.fit(X2_train,Y2_train)
```

Figura 4.12: Entrenamiento de los modelos de regresión logística

Una vez creados los 2 modelos (uno para cada riesgo) se procede a predecir utilizando las datas de prueba respectivas.

```
#Utilizamos nuestro modelo para predecir los valores del dataset test
Y1_pred=regresion_riesgo_R.predict(X1_test)
Y2_pred=regresion_riesgo_P.predict(X2_test)

Y2_pred
```

```
array(['SI', 'NO', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'SI',
       'SI', 'SI', 'SI', 'SI', 'SI', 'SI', 'NO', 'SI', 'SI', 'NO', 'SI',
       'NO', 'NO', 'NO', 'NO', 'SI', 'SI', 'NO', 'SI', 'NO', 'NO', 'NO',
       'NO', 'SI', 'NO', 'SI', 'SI', 'NO', 'SI', 'SI', 'NO', 'SI', 'SI',
       'SI', 'NO', 'SI', 'SI', 'SI', 'SI', 'NO', 'SI', 'SI', 'SI', 'NO',
       'NO', 'SI', 'SI', 'SI', 'SI', 'SI'], dtype=object)
```

Figura 4.13: Predicción de los modelos de regresión logística

Luego de obtener las predicciones Y1_pred y Y2_pred, utilizaremos el comando classification_report para visualizar el reporte de métricas del modelo regresión_riesgo_R y el modelo regresión_riesgo_P.

```
report_1= classification_report(Y1_test, Y1_pred)

print(report_1)
```

```
              precision    recall  f1-score   support

          NO       0.89      0.52      0.65        33
          SI       0.62      0.93      0.74        28

    accuracy                           0.70        61
   macro avg       0.76      0.72      0.70        61
weighted avg       0.77      0.70      0.69        61
```

Figura 4.14: Reporte de métricas del modelo regresión_riesgo_R

```
report_2= classification_report(Y2_test, Y2_pred)

print(report_2)
```

```
              precision    recall  f1-score   support

          NO       0.58      0.61      0.59        18
          SI       0.83      0.81      0.82        43

    accuracy                           0.75        61
   macro avg       0.71      0.71      0.71        61
weighted avg       0.76      0.75      0.76        61
```

Figura 4.15: Reporte de métricas del modelo regresión_riesgo_P

4.4.3. Modelos de Random Forest

Siguiendo los pasos de los modelos anteriores, primero debemos llamar al algoritmo de Random Forest, proceder a entrenar cada modelo con su respectiva data de entrenamiento y luego predecir utilizando cada modelo creado con su respectiva data de prueba.

```
#Llamamos al algoritmo de random forest
bosque = RandomForestClassifier()

#Entrenamos los modelos
bosque_riesgo_R = bosque.fit(X1_train,Y1_train)
bosque_riesgo_P = bosque.fit(X2_train,Y2_train)
```

```
#Utilizamos nuestro modelo para predecir los valores del dataset test
Y1_pred=bosque_riesgo_R.predict(X1_test)
Y2_pred=bosque_riesgo_P.predict(X2_test)
```

Figura 4.16: Entrenamiento y predicción de los modelos de random forest

Luego de obtener las predicciones Y1_pred y Y2_pred, utilizaremos el comando classification_report para visualizar el reporte de métricas del modelo bosque_riesgo_R y el modelo bosque_riesgo_P.

```
report_1= classification_report(Y1_test, Y1_pred)

print(report_1)
```

```
              precision    recall  f1-score   support

          NO       0.95      0.61      0.74        33
          SI       0.68      0.96      0.79        28

    accuracy                           0.77        61
   macro avg       0.81      0.79      0.77        61
weighted avg       0.83      0.77      0.77        61
```

Figura 4.17: Reporte de métricas del modelo bosque_riesgo_R

```
[21] report_2= classification_report(Y2_test, Y2_pred)

     print(report_2)
```

```
              precision    recall  f1-score   support

          NO       0.67      0.78      0.72        18
          SI       0.90      0.84      0.87        43

    accuracy                           0.82        61
   macro avg       0.78      0.81      0.79        61
weighted avg       0.83      0.82      0.82        61
```

Figura 4.18: Reporte de métricas del modelo bosque_riesgo_P

4.4.4. Modelos de K_Vecinos más cercanos

Primero llamamos al algoritmo de K_Vecinos más cercanos, luego se procede a entrenar cada modelo con su respectiva data de entrenamiento y después a predecir utilizando cada modelo creado con su respectiva data de prueba.

```
#Llamamos al algoritmo de k-vecinos mas cercanos
kvecinos = KNeighborsClassifier()

#Entrenamos los modelos
kvecinos_riesgo_R = kvecinos.fit(X1_train,Y1_train)
kvecinos_riesgo_P = kvecinos.fit(X2_train,Y2_train)
```

```
#Utilizamos nuestro modelo para predecir los valores del dataset test
Y1_pred=kvecinos_riesgo_R.predict(X1_test)
Y2_pred=kvecinos_riesgo_P.predict(X2_test)
```

Figura 4.19: Entrenamiento y predicción de los modelos de K_Vecinos más cercanos

Luego de obtener las predicciones Y1_pred y Y2_pred, utilizaremos el comando classification_report para visualizar el reporte de métricas del modelo kvecinos_riesgo_R y el modelo kvecinos_riesgo_P.

```
report_1= classification_report(Y1_test, Y1_pred)

print(report_1)
```

```
              precision    recall  f1-score   support

          NO       0.94      0.52      0.67        33
          SI       0.63      0.96      0.76        28

    accuracy                           0.72        61
   macro avg       0.79      0.74      0.71        61
weighted avg       0.80      0.72      0.71        61
```

Figura 4.20: Reporte de métricas del modelo kvecinos_riesgo_R

```
report_2= classification_report(Y2_test, Y2_pred)

print(report_2)
```

```
              precision    recall  f1-score   support

          NO       0.61      0.61      0.61        18
          SI       0.84      0.84      0.84        43

    accuracy                           0.77        61
   macro avg       0.72      0.72      0.72        61
weighted avg       0.77      0.77      0.77        61
```

Figura 4.21: Reporte de métricas del modelo kvecinos_riesgo_P

4.4.5. Modelos de Naive Bayes

Primero llamamos al algoritmo de Naive Bayes, luego se procede a entrenar cada modelo con su respectiva data de entrenamiento y después a predecir utilizando cada modelo creado con su respectiva data de prueba.

```
#Llamamos al algoritmo Naive bayes
gnb = GaussianNB()

#Entrenamos los modelos
gnb_riesgo_R = gnb.fit(X1_train,Y1_train)
gnb_riesgo_P = gnb.fit(X2_train,Y2_train)
```

```
#Utilizamos nuestro modelo para predecir los valores del dataset test
Y1_pred=gnb_riesgo_R.predict(X1_test)
Y2_pred=gnb_riesgo_P.predict(X2_test)
```

Figura 4.22: Entrenamiento y predicción de los modelos de Naive Bayes

Luego de obtener las predicciones Y1_pred y Y2_pred, utilizaremos el comando classification_report para visualizar el reporte de métricas del modelo gnb_riesgo_R y el modelo gnb_riesgo_P.

```
report_1= classification_report(Y1_test, Y1_pred)

print(report_1)
```

	precision	recall	f1-score	support
NO	0.73	0.48	0.58	33
SI	0.56	0.79	0.66	28
accuracy			0.62	61
macro avg	0.65	0.64	0.62	61
weighted avg	0.65	0.62	0.62	61

Figura 4.23: Reporte de métricas del modelo gnb_riesgo_R

```
report_2= classification_report(Y2_test, Y2_pred)

print(report_2)
```

	precision	recall	f1-score	support
NO	0.50	0.61	0.55	18
SI	0.82	0.74	0.78	43
accuracy			0.70	61
macro avg	0.66	0.68	0.67	61
weighted avg	0.73	0.70	0.71	61

Figura 4.24: Reporte de métricas del modelo gnb_riesgo_P

Capítulo 5

Resultados

Después de crear los modelos predictivos y obtener los reportes de clasificación de cada uno de ellos, es esencial llevar a cabo una comparación y análisis para determinar las fortalezas y debilidades que tiene cada modelo.

Algoritmo	Riesgo_R				Riesgo_P			
	Clase_No		Clase_Si		Clase_No		Clase_Si	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Árbol de decisión	0.90	0.55	0.63	0.93	**0.70**	**0.78**	**0.90**	**0.86**
Regresión Logística	0.89	0.52	0.62	0.93	0.58	0.61	0.83	0.81
Random Forest	**0.95**	**0.61**	**0.68**	**0.96**	0.67	0.78	0.90	0.84
K-Vecinos	0.94	0.52	0.63	0.96	0.61	0.61	0.84	0.84
Naive Bayes	0.73	0.48	0.56	0.79	0.50	0.61	0.82	0.74

Figura 5.1: Cuadro comparativo de los modelos predictivos

En la figura 5.1 se visualiza un resumen de las métricas obtenidas en cada modelo, tanto para el riesgo_R y el riesgo_P como para cada una de sus clases (Clase_No y Clase_Si).

Para este proyecto no se utiliza la métrica accuracy (exactitud) para la evaluación, esto debido a que estamos utilizando una muestra desbalanceada, es decir que tenemos mayor cantidad de datos en una clase que en otra, y esto hace que el rendimiento de esta métrica no sea óptima.

Por otra parte, se decide omitir la métrica F1-Score, esto debido a que esta métrica nos brinda un valor resultante de la combinación o fusión de las métricas precisión y recall, las cuales si estamos tomando en cuenta, debido a que estas 2 ultimas métricas nos ayudan a identificar las fortalezas y debilidades que tiene cada modelo.

Al momento de evaluar las métricas precisión y exhaustividad (recall) de los modelos, nos enfocaremos en la Clase_Si de cada riesgo, dado que el objetivo de este proyecto se enfoca en modelos predictivos que identifiquen a los estudiantes que SI están en riesgo académico.

La métrica precisión respecto a la Clase_Si nos ayuda a saber que porcentaje de estudiantes que el modelo indique en riesgo académico (Riesgo_R o Riesgo_P) estarán realmente en riesgo académico.

- En el caso de modelos predictivos de Riesgo_R, se observa que tenemos una precisión general de la Clase_Si alrededor del 60 %, es decir, que aprox. el 60 % de estudiantes que el modelo prediga estarán realmente en Riesgo_R (riesgo de desaprobar mas de un curso en el nuevo semestre).
- En el caso de modelos predictivos de Riesgo_P, se observa que tenemos una precisión general de la Clase_Si alrededor del 85 %, es decir, que aprox. el 85 % de estudiantes que el modelo prediga estarán realmente en Riesgo_P (riesgo de obtener un promedio de notas menor a 11 en el nuevo semestre).

La métrica exhaustividad (recall) respecto a la Clase_Si nos ayuda a saber que porcentaje de estudiantes que realmente están en riesgo académico (Riesgo_R o Riesgo_P) es capaz de identificar nuestro modelo.

- En el caso de modelos predictivos de Riesgo_R, se observa que tenemos una exhaustividad general de la Clase_Si alrededor del 90 %, es decir, que nuestros modelos son capaces de identificar aprox. al 90 % de estudiantes que realmente están en riesgo_R.
- En el caso de modelos predictivos de Riesgo_P, se observa que tenemos una exhaustividad general de la Clase_Si alrededor del 80 %, es decir, que nuestros modelos son capaces de identificar aprox. al 80 % de estudiantes que realmente están en riesgo_P.

Luego de evaluar y entender la información que nos brinda la métrica precisión y recall para la Clase_Si en los modelos de riesgo, podemos pasar a comparar los resultados generales.

- Entre los modelos de Riesgo_R y Riesgo_P, se nota que los modelos que predicen el Riesgo_P (riesgo de obtener un promedio de notas menor a 11 en el nuevo semestre) tienen muy buenos valores tanto para la precisión como para la exhaustividad.
- Al comparar los valores de precisión y exhaustividad de cada uno de los modelos de Riesgo_R, se nota que se obtuvo un mejor resultado con el modelo creado utilizando la técnica de Random Forest.
- Al comparar los valores de precisión y exhaustividad de cada uno de los modelos de Riesgo_P, se nota que se obtuvo un mejor resultado con el modelo creado utilizando la técnica de Árbol de decisión.

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

En este trabajo, se llevo a cabo el uso de diversas técnicas de machine learning con el propósito de crear modelos predictivos de riesgo académico que presenten un buen nivel de exhaustividad (recall) y precisión. Esto le facilita a la escuela la identificación de casos de estudiantes en riesgo académico, especialmente aquellos que se matricularon en el nuevo semestre de la carrera de computación científica, dentro de la Facultad de Ciencias Matemáticas de la UNMSM.

En las literaturas revisadas, se pudo notar la importancia de estudiar y analizar cuales son los factores que causan el bajo rendimiento académico (riesgo académico) en los estudiantes, la cual aun es una tarea compleja debido a que estas variables no se limitan únicamente a factores académicos, sino que también incluyen elementos del contexto socio demográfico, así como aspectos cognitivos e interpersonales. También se notó que las métricas más frecuentemente utilizadas para determinar el rendimiento académico de los alumnos son las calificaciones y el promedio de sus calificaciones.

La etapa de recolección y preparación de datos resulto ser de gran importancia para obtener información verídica y generar atributos relevantes que pudieran ser empleados en el entrenamiento de los modelos. Se llevó a cabo un análisis de datos, lo que culminó en la creación de un archivo que contenía 241 registros de estudiantes, cada uno con un conjunto de 8 atributos.

Las técnicas de clasificación de machine learning se aplicaron utilizando Google Colab y se empleó el dataset que contenía al conjunto de variables seleccionadas. Además, se tomaron en cuenta dos casos de variables categóricas (clases) para determinar los valores de predicción. En el primer caso, se evaluó la cantidad de cursos reprobados, considerando que la reprobación de más de 1 curso se interpretaba como un indicador afirmativo de riesgo académico, lo que se reflejó en la variable categórica “Riesgo_R”. En el segundo caso, se evaluó el promedio de notas, considerando que un promedio de notas menor a 11 se interpretaba como un indicador afirmativo de riesgo académico, lo que se reflejó en la variable categórica “Riesgo_P”.

Dentro de las variables elegidas, se observo que la variable "Prom_US" (promedio del ultimo semestre) posee un poder predictivo significativo para la identificación de ambos tipos de riesgo académico (Riesgo_R y Riesgo_P). Este atributo es seguido en importancia por las variables "Prom_P" y "Cursos_A" (promedio ponderado y cantidad de cursos aprobados).

Se desarrollaron modelos predictivos utilizando cinco algoritmos de clasificación de machine learning. Para cada algoritmo, se consideraron las dos variables categóricas y se calcularon los valores de exhaustividad y predicción con el fin de comparar su rendimiento. En términos generales, se observó que la variable "Riesgo_P" obtuvo los mejores resultados de precisión y exhaustividad en la Clase_Si en todos los algoritmos.

Adicional a ello, se determinó que el mejor algoritmo predictivo para la variable "Riesgo_R" fue "Random Forest", con unas métricas de precisión y exhaustividad para la Clase_Si del 68 % y 96 % respectivamente; en el caso de la variable "Riesgo_P", su mejor algoritmo fue "Árbol de decisión", con unas métricas de precisión y exhaustividad para la Clase_Si del 90 % y 86 % respectivamente.

6.2. Recomendaciones

- Durante la etapa de recolección de datos, se recomienda solicitar el apoyo de la escuela para acceder a los registros académicos de los años más recientes, excluyendo aquellos afectados por la pandemia. Esto contribuirá a obtener modelos más eficaces.
- Durante la etapa de preparación de datos, se aconseja aplicar técnicas de balanceo de datos con el fin de lograr una distribución más uniforme entre las categorías "Clase_Sí" y "Clase_No", lo que conducirá a la mejora de los modelos en términos de exhaustividad y precisión.
- Se recomienda la implementación de una interfaz que sea fácil de usar y amigable, con el objetivo de simplificar la ejecución de los modelos desarrollados.
- Se aconseja a la escuela llevar a cabo un monitoreo continuo de los estudiantes en riesgo académico que han sido identificados por los modelos, con el fin de evitar que el numero de alumnos en esta situación siga incrementado.
- Se recomienda replicar la metodología desarrollada en las demás escuelas de la Facultad de Ciencias Matemáticas, con el propósito de identificar no solo a los estudiantes en riesgo académico de la Escuela de Computación Científica, sino también de toda la facultad.

Bibliografía

Alan. (2021). Qué es el Machine Learning y cómo funciona. *Machine Learning con Python en Español*. https://machinelearningenespanol.com/2021/07/21/que-es-machine-learning-como-funciona/

Arbulu Velez, F. R. (2023). *Reporte de seguimiento del PEI 2109-2023* (inf. téc.). Universidad Nacional Mayor de San Marcos.

Bustamante Peña, D. L. y col. (2021). Modelo predictivo de rendimiento académico para el apoyo, prevención y disminución de la tasa de deserción universitaria.

Corporativa, I. (2023). Descubre los principales beneficios del machine learning. https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico

Dorio Alcaraz, I. (2017). La transición a la Universidad. El grado de maestro de Educación Infantil.

Europea, U. (2022). Aprendizaje supervisado y no supervisado. *Universidad Europea*. https://universidadeuropea.com/blog/aprendizaje-supervisado-no-supervisado/#:~:text=La%20principal%20diferencia%20entre%20el,sin%20etiquetar%20o%20sin%20procesar.

Franco, E. A., Martınez, R. E. L. & Domınguez, V. H. M. (2021). Modelos predictivos de riesgo académico en carreras de computación con mineria de datos educativos. *Revista de Educación a Distancia (RED)*, *21*(66).

Gonzalez, L. (2020). Árboles de decisión clasificación–teoría. *Aprende IA*. https://aprendeia.com/arboles-de-decision-clasificacion-teoria-machine-learning/

Gonzalez, L. (2022a). K Vecinos más cercanos - teoría. *Aprende IA*. https://aprendeia.com/algoritmo-k-vecinos-mas-cercanos-teoria-machine-learning/#

Gonzalez, L. (2022b). Naive Bayes – teoría. *Aprende IA*. https://aprendeia.com/algoritmo-naive-bayes-machine-learning/

Gonzalez, L. (2022c). Regresión logística; teoría. *Aprende IA*. https://aprendeia.com/algoritmo-regresion-logistica-machine-learning-teoria/

Heras, J. M. (2020a). Precision, recall, F1, accuracy en clasificación-IArtificial.net. https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

Heras, J. M. (2020b). Random Forest (Bosque aleatorio): Combinando árboles - IArtificial.net. https://www.iartificial.net/random-forest-bosque-aleatorio/

Lamas, H. A. (2015). Sobre el rendimiento escolar. *Propósitos y Representaciones*, *3*(1), 313-386. https://doi.org/10.20511/pyr2015.v3n1.74

Moreno Murcia, J. A., Silveira Torregrosa, Y. & Alias Garcıa, A. (2015). Modelo predictivo para la mejora de la percepción de competencia y rendimiento académico en estudiantes universitarios. *REDU. Revista de Docencia Universitaria*, *13*(2), 173-188.

Ramírez, N. D. G. & Páez, A. R. (2022). Modelos predictivos del rendimiento académico a partir de características de estudiantes de ingeniería. *IE Revista de Investigación Educativa de la REDIECH*, *13*, 1-18.

Rico Páez, A. (2022). Modelos predictivos progresivos del rendimiento académico de estudiantes universitarios. *RIDE. Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, *12*(24).

Team, D. (2022). Machine Learning: definición, funcionamiento, usos. https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos

Vergaray, K. R. (2021). Modelo predictivo para la detección temprana de estudiantes con alto riesgo de deserción académica. *Innovación y software*, *2*(2), 6-13.

York, T. T., Gibson, C. & Rankin, S. (2015). Defining and measuring academic success. *Practical assessment, research, and evaluation*, *20*(1), 5.