ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Aplicación de modelos de series temporales y redes neuronales para analizar y proyectar la evolución académica de estudiantes de secundaria

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Ciencias de Datos

Presentado por:

Luis Vinicio Torres Vivanco

GUAYAQUIL - ECUADOR

Año: 2022

AGRADECIMIENTO

Mi más sincero agradecimiento a todas las instituciones que aportaron con la información necesaria para realizar este trabajo, a ESPOL y sus docentes por los conocimientos impartidos y a mi tutora por la orientación y ayuda.

DEDICATORIA

A mi familia, a mis hijos y en especial a mi madre.

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
SANDRA LORENA GARCIA BUSTOS

Sandra Lorena García Bustos, Ph.D.

PROFESORA TUTORA

Firmado electrónicamente por:
JOSE EDUARDO CORDOVA GARCIA

José Eduardo Córdova García, Ph.D.

MIEMBRO DEL TRIBUNAL

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Luis Vinicio Torres Vivanco y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Firmado electrónicamente por:
LUIS VINICIO TORRES VIVANCO

Luis Vinicio Torres Vivanco

RESUMEN

Las instituciones educativas implementan metodologías, mejores prácticas, entre otras actividades para mejorar la calidad de la enseñanza, su efecto usualmente se analiza utilizando indicadores estadísticos simples basados en información referente al rendimiento académico de los estudiantes que está compuesta por una gran cantidad de datos. El presente trabajo aprovecha esta información para analizar el rendimiento mediante modelos de series temporales y redes neuronales, proyectando su comportamiento y proveyendo a los directivos una herramienta para la toma de decisiones.

La información que se utilizó en este trabajo pasó por un proceso de extracción, depuración y transformación previo al entrenamiento del modelo desarrollado. Este modelo se utilizó en la implementación de una API que sirvió para la elaboración del *front end* en la forma de una aplicación web.

Luego de entrenar el modelo por 32 épocas se obtuvo una exactitud del 84.01% y una pérdida del 41.94% durante el entrenamiento, y una exactitud del 91.17% y una pérdida del 28% en la validación, valores aceptables frente al 90% mencionado en el estado del arte.

En conclusión, es factible utilizar la información del rendimiento académico de los estudiantes para elaborar modelos basados en series de tiempo y redes neuronales que permitan lograr los objetivos planteados con una exactitud aceptable, sin embargo, el hardware y software necesarios dificultan su implementación en las unidades educativas, generando una oportunidad para su provisión como una herramienta de tipo SaaS.

Palabras Clave: Rendimiento estudiantil, series de tiempo, redes neuronales, educación secundaria.

ABSTRACT

Educational institutions implement methodologies, best practices, among other activities to improve the quality of teaching, its effect is usually analyzed using simple statistical indicators based on student's academic performance information composed of a large amount of data. The present work takes advantage of this information to analyze the performance through time series and neural networks models, forecasting its behavior and providing managers with a decision-making tool.

The information used in this work went through a process of extraction, cleaning, and transformation prior to training the developed model. This model was used in the implementation of an API that was used by develop a web application as front end.

After training the model during 32 epochs, an accuracy of 84.01% and a loss of 41.94% were obtained during training, and an accuracy of 91.17% and a loss of 28% in validation, acceptable values if they are compared to the 90% mentioned in the state of the art.

In conclusion, we can use the student's academic performance information to develop models based on time series and neural networks that allow to achieve the objectives set in this work with an acceptable accuracy, however, the necessary hardware and software make it difficult to implement them into academic institutions, generating an opportunity for its provision as a SaaS-type product.

***Keywords**: Student performance, time series, neural networks, secondary education.*

ÍNDICE GENERAL

RESUMEN I

ABSTRACT II

ÍNDICE GENERAL III

ABREVIATURAS VII

ÍNDICE DE FIGURAS VIII

ÍNDICE DE TABLAS XI

CAPÍTULO 1 1

1. Introducción 1

1.1 Descripción del problema 1

1.2 Justificación del problema 1

1.3 Solución Propuesta 2

1.4 Objetivos 2

1.4.1 Objetivo general 2

1.4.2 Objetivos específicos 2

1.5 Metodología 3

1.6 Resultados Esperados 6

1.7 Dataset 7

CAPÍTULO 2 11

2. estado del arte 11

2.1 Fundamentos del problema 11

2.2 Soluciones de analítica y aprendizaje relacionadas al problema 12
2.2.1 Series temporales 14
2.2.2 Arquitectura de redes neuronales tipo "*Transformer*" 17
2.2.3 Implementación de series temporales con redes neuronales tipo "*transformer*" 25
2.3 Fuentes de datos relacionadas al problema 25
2.4 Librerías y software a utilizar 26
2.4.1 Extracción de datos para el entrenamiento 26
2.4.2 Diseño y entrenamiento del modelo 26
2.4.3 Desarrollo de la herramienta de software interactiva 26
CAPÍTULO 3 28
3. diseño e implementación 28
3.1 Exploración y validación de datos y fuentes. 28
3.1.1 Valores inexistentes 28
3.1.2 Atributos con valores atípicos 28
3.1.3 Acciones correctivas sobre los datos atípicos encontrados 29
3.1.4 Atributos adicionales 29
3.1.5 Prueba de normalidad 30
3.1.6 Estadística descriptiva 30
3.1.7 Correlación entre atributos numéricos 33
3.1.8 Correlaciones con la variable a predecir: calificación 34
3.1.9 Verificación de independencia entre variables categóricas 36

3.1.10 Selección de los atributos a utilizar 39
3.2 Prototipos de algoritmos, modelos, y módulos del sistema 40
3.2.1 Análisis de series temporales 40
3.2.2 Selección del modelo de predicción 45
3.2.3 Modelo de predicción 45
3.2.4 Arquitectura de la solución 53
3.3 3.3 Infraestructura para procesamiento y almacenamiento 64
3.4 Plataformas y prototipos de visualización 64
3.4.1 Acceso a la aplicación 65
3.4.2 Operación general de las interfaces para predicción 65
3.4.3 Codificación de colores de la información en los gráficos 66
3.4.4 Predicción general 67
3.4.5 Predicción por área 68
3.4.6 Predicción por tipo de actividad 69
3.5 Métricas y comunicación de resultados 70
CAPÍTULO 4 72
4. análisis de resultados 72
4.1 Recolección de datos y estrategias para validación del proyecto 72
4.2 Puesta en marcha y funcionamiento 74
4.3 Pruebas de funcionalidad 75
4.4 Análisis costo/beneficio 77
4.4.1 Costos 78

4.4.2 Beneficios 78

CONCLUSIONES Y RECOMENDACIONES 81

Conclusiones 81

Recomendaciones 82

REFERENCIAS BIBLIOGRAFICAS 83

GLOSARIO 87

ABREVIATURAS

API	Application programming interface
CPU	Central process unit
CRUD	Create, recover, update, and delete
CSV	Comma separated values
ESPOL	Escuela Superior Politécnica del Litoral
GRNN	Generalized Regression Neural Network
HTTP	Hypertext transfer protocol
JSON	JavaScript object notation
LMS	Learning management system
LSTM	Long short-term memory
MLP	Multilayer Perceptron
NaN	Not a number
NLP	Natural language processing
PEA	Proceso de enseñanza / aprendizaje
RAM	Random access memory
REST	Representational state transfer
RNN	Recurrent neural networks
SaaS	Software as a service
SPA	Singe page application
SQL	Structured query language
TIR	Tasa interna de retorno
VAN	Valor actual neto

ÍNDICE DE FIGURAS

Figura 1.1 Grupos de información a ingresar al modelo. 4

Figura 1.2 Metodología para la extracción de variables significativas a utilizar en el modelo. 5

Figura 1.3 Metodología para la mejora continua del modelo. 6

Figura 2.1 Diagrama de bloques de una arquitectura de red neuronal tipo “Transformer”. 17

Figura 2.2 Diagrama de bloques del “Bloque atencional”. 20

Figura 2.3 Diagrama de bloques del “Bloque residual”. 21

Figura 2.4 Diagrama de bloques de la “Red neuronal” de un bloque “Codificador”. 22

Figura 2.5 Enmascaramiento de las probabilidades del bloque atencional. 23

Figura 2.6 Arquitectura de la herramienta de software interactiva. 26

Figura 3.1 Histogramas de los atributos numéricos. 32

Figura 3.2 Gráfico de correlación entre atributos numéricos. 33

Figura 3.3 Gráfico de correlación sin atributos fuertemente correlacionados. 35

Figura 3.4 Gráfico de correlación entre atributos con y sin pandemia. 36

Figura 3.5 Gráfico de correlación de tablas de contingencia. 38

Figura 3.6 Evolución de las calificaciones en la asignatura de Matemáticas en segundo año de BGU para tres estudiantes diferentes. 41

Figura 3.7 Evolución de las calificaciones en tres asignaturas diferentes para el mismo estudiante. 42

Figura 3.8 Tablas de correlación entre la variable a predecir y las variables categóricas. 44

Figura 3.9 Definición del modelo de predicción. 46

Figura 3.10 Transformación del dataset con los atributos seleccionados a un conjunto de tokens para el entrenamiento del modelo. 47

Figura 3.11 Estructura del token. 48

Figura 3.12 Segmentación del arreglo de tokens. 51

Figura 3.13 Evolución de la función de pérdida y validación en el proceso de entrenamiento. 52

Figura 3.14 Proceso de predicción de calificaciones. 53

Figura 3.15 Arquitectura de la solución, vista por etapas. 54

Figura 3.16 Arquitectura de la solución, componentes que interactúan con la aplicación web. 55

Figura 3.17 Ejemplo de la estructura JSON enviada como parámetro al servicio web POST /predictions/. 58

Figura 3.18 Procesos del servicio web para ejecución del modelo de predicción. 59

Figura 3.19 Ejemplo de la respuesta del servicio web de predicción. 59

Figura 3.20 Ejemplo de la herramienta de navegación en la documentación del API para la generación de predicciones. 63

Figura 3.21 Pantalla de control de acceso a la aplicación web. 65

Figura 3.22 Estructura general de las interfaces de consulta de predicciones. 66

Figura 3.23 Ejemplo de la interfaz de predicción general. 68

Figura 3.24 Ejemplo de la interfaz de predicción por área del conocimiento. 69

Figura 3.25 Ejemplo de la interfaz de predicción por tipo de actividad. 70

Figura 4.1 Gráfico de la evolución del entrenamiento del modelo. 72

Figura 4.2 Gráfico de la evolución de la exactitud del modelo. 73

Figura 4.3 Ejemplo de una predicción general de calificaciones. 75

Figura 4.4 Ejemplo de una predicción general de calificaciones. 76

Figura 4.5 Ejemplo de una predicción general de calificaciones. 77

ÍNDICE DE TABLAS

Tabla 1.1 Atributos del dataset. 7

Tabla 1.2 Niveles de estudio. 8

Tabla 1.3 Códigos de área de estudio. 8

Tabla 1.4 Códigos de grado de estudio. 9

Tabla 1.5 Códigos de tipo de actividad. 9

Tabla 1.6 Códigos de géneros. 10

Tabla 2.1 Componentes de la arquitectura de la herramienta de software interactiva. .27

Tabla 2.2 Librerías y paquetes de software a utilizar en la arquitectura de la herramienta de software interactiva. 27

Tabla 3.1 Valores inexistentes. 28

Tabla 3.2 Valores máximos y mínimos de los atributos numéricas. 29

Tabla 3.3 Atributos creados para el análisis de la información. 30

Tabla 3.4 Parámetros de estadística descriptiva. 30

Tabla 3.5 Parámetros de estadística descriptiva (continuación). 31

Tabla 3.6 Correlación más fuertes entre atributos. 34

Tabla 3.7 Segmentación de variables continuas. 36

Tabla 3.8 Resultados de la prueba de independencia entre variables categóricas. 37

Tabla 3.9 Omisión de atributos del dataset para el entrenamiento del modelo. 39

Tabla 3.10 Atributos seleccionados para el entrenamiento del modelo. 40

Tabla 3.11 Factores de correlación entre la variable a predecir y el resto de las variables numéricas. 43

Tabla 3.12 Factores de correlación entre la variable a predecir y las variables categóricas. 43

Tabla 3.13 Parámetros para la definición del modelo de predicción.....46

Tabla 3.14 Estructura de la cabecera del token. 48

Tabla 3.15 Códigos numéricos para el grado de estudio. 48

Tabla 3.16 Códigos numéricos para el área de estudio. 49

Tabla 3.17 Códigos numéricos para los rangos de edad. 49

Tabla 3.18 Estructura de cada grupo del detalle del token. 49

Tabla 3.19 Códigos asignados a los tipos de actividad. 50

Tabla 3.20 Códigos asignados a elementos de delimitación de tokens.....51

Tabla 3.21 Arreglos generados para el entrenamiento. 51

Tabla 3.22 Procesos de la etapa de definición y entrenamiento del modelo. 54

Tabla 3.23 Estructura del parámetro del servicio web POST /predictions/. 56

Tabla 3.24 Estructura de los “scores” (calificaciones) enviados al servicio web POST /predictions/. 57

Tabla 3.25 Estructura del la respuesta del servicio web de predicción.....59

Tabla 3.26 Clases asignadas a la calificación, según la Ley Orgánica de Educación Intercultural.....60

Tabla 3.27 Servicios web definidos por la interfaz IRestCRUDController.....60

Tabla 3.28 Clases de los objetos gestionados por los servicios web implementados con la interfaz IRestCRUDController. 61

Tabla 3.29 Servicios web para la ejecución de predicciones. 61

Tabla 3.30 Rutas de acceso a la documentación del API para la generación de predicciones. 62

Tabla 3.31 Infraestructura de procesamiento....64

Tabla 3.32 Componentes de las interfaces gráficas para predicción....66

Tabla 3.33 Codificación de colores de los gráficos. 67

Tabla 3.34 Elementos de la interfaz de Predicción General. 67

Tabla 3.35 Elementos de la interfaz de Predicción General. 68

Tabla 3.36 Elementos de la interfaz de Predicción General. 69

Tabla 3.37 Indicadores para medir los resultados obtenidos. 70

Tabla 4.1 Resultados de las métricas en los primeros 32 entrenamientos. 73

Tabla 4.2 Distribución de los componentes de la solución en los servidores implementados. 74

Tabla 4.3 Valores de los indicadores para evaluación del trabajo. 77

Tabla 4.4 Costos relacionados al trabajo. 78

Tabla 4.5 Costos relacionados al trabajo. 78

Tabla 4.6 Escenarios para el análisis del costo/beneficio. 79

Tabla 4.7 Resumen de ingresos esperados por la comercialización del módulo. 79

Tabla 4.8 Resultados para los escenarios pesimista, esperado y optimista. 80

Tabla 4.9 Resultados para los escenarios pesimista, esperado y optimista. 80

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del problema

Uno de los objetivos primordiales de las instituciones educativas es la búsqueda de la excelencia, para ello las entidades implementan metodologías, mejores prácticas, estándares, entre otras medidas que van orientadas a mejorar día a día la calidad de la enseñanza. Lo ideal en este caso sería el análisis oportuno de los efectos que las acciones antes descritas tienen en el proceso de enseñanza/aprendizaje (PEA), sin embargo, a pesar de que se cuenta con información histórica con un nivel de detalle considerable; el volumen y complejidad de esta, hacen que el análisis de los resultados se realice de una manera somera y, en la mayoría de los casos, una vez que se ha terminado el ciclo académico, lo que dificulta tomar medidas oportunas para mejorar el desempeño académico de los estudiantes.

Por otra parte, como consecuencia de las limitaciones propias del análisis realizado, el cual se basa por lo general en indicadores estadísticos básicos, es muy común que la mayoría de las acciones que se toman sean de índole grupal, haciendo que el tratamiento individual de la problemática de cada estudiante sea mínimo y en algunos casos nulo.

1.2 Justificación del problema

Las primeras etapas de la vida de una persona son críticas para desarrollar y cimentar su pensamiento y las habilidades relacionadas al proceso de aprendizaje, por esta razón es común que las instituciones educativas, en coordinación con los entes rectores de la educación, investiguen e implementen metodologías que aprovechen estas primeras etapas. Uno de los resultados de esta implementación es la gran cantidad de datos que se generan día a día. Lamentablemente, estos datos en la mayoría de los casos son subutilizados, extrayendo de ellos indicadores estadísticos básicos con el ánimo de obtener información para la toma de decisiones. Esta práctica se ha mantenido casi inalterable desde que se llevan registros de las calificaciones obtenidas por los estudiantes. Con la incorporación

al proceso de herramientas informáticas, tales como sistemas y bases de datos, se ha agilitado y mejorado la calidad de la información estadística obtenida, sin embargo la cantidad y tipo de ésta se mantiene, a pesar de que el volumen y variedad de la información existente hace posible el uso de herramientas relacionadas a la inteligencia artificial para extraer más información de utilidad, e incluso modelar el comportamiento de los estudiantes para con ello realizar pronósticos que permitan la toma de decisiones oportunas e informadas.

1.3 Solución Propuesta

Para solucionar la problemática planteada podemos utilizar técnicas estadísticas y de inteligencia artificial para generar perfiles de los principales involucrados en el proceso de enseñanza/aprendizaje (docentes y estudiantes) con la finalidad de modelar su comportamiento y con ello predecir su estado al final del periodo académico. Estos perfiles procesados dentro de una herramienta que facilite a los directivos identificar los casos que presenten mayor riesgo académico, facilitará la toma oportuna de decisiones para velar por una mejora continua en el desempeño académico de los estudiantes y docentes.

1.4 Objetivos

1.4.1 Objetivo general

Crear una aplicación usando modelos de series temporales y redes neuronales para analizar y proyectar la evolución educativa de estudiantes de educación secundaria.

1.4.2 Objetivos específicos

1. Aplicar técnicas estadísticas para extraer atributos y valores que reflejen significativamente la evolución académica y cognitiva de los estudiantes en el tiempo.
2. Desarrollar un modelo basado en técnicas de redes neuronales y series temporales que utilicen los atributos del estudiante y del docente para predecir el estado académico y cognitivo del estudiante.

3. Compilar los datos estadísticos y el modelo de inteligencia artificial en una herramienta de software que permita a los directivos la toma de decisiones oportunas en el proceso de enseñanza/aprendizaje de su institución.

1.5 Metodología

Por acuerdo con la entidad que nos provee la información todos los datos serán anonimizados, por lo que previo al proceso se realizará una ofuscación de la información sensible de estudiantes y docentes, también a la institución la referenciaremos con el término de “Unidad Educativa” en toda la documentación pública de este trabajo.

La información que utilizaremos se limita a un grupo de 20 estudiantes que en el 2021 cursan el segundo año de bachillerato general unificado. Todos los estudiantes pertenecen al mismo paralelo y en algunos casos cuentan con un historial académico en la misma institución desde el 2017. No se utiliza información de periodos anteriores al 2017 porque ésta no se encuentra en una base de datos informática.

Es importante considerar que existen múltiples factores que influyen en mayor o menor medida en el rendimiento académico de lo estudiantes, por lo que es importante determinar su importancia antes de modelar su comportamiento, en esto nos ayuda el trabajo de Arteaga e Ibarra (Arteaga & Ibarra, 2020), quienes identifican en los primeros lugares de importancia a los factores de: Ambiente de aprendizaje en el Aula, Currículo institucional, Preparación Profesional del Profesor, Papel de la Comunidad Local y Papel de las Políticas Educativas Nacionales. Para este trabajo estamos considerando el valor del Comportamiento como un indicador indirecto del “Ambiente de aprendizaje en el Aula”, mientras que la “Preparación Profesional del Profesor” la estamos parametrizando como el rendimiento promedio de los estudiantes a quienes el docente ha dictado cátedra. La medición del Currículo Institucional, el Papel de la Comunidad Local y el Papel de las Políticas Educativas Nacionales, están fuera del alcance del presente trabajo.

La información que se utilizará proviene principalmente de las calificaciones obtenidas por los estudiantes durante los periodos antes mencionados, sin embargo, se analizará también la significancia de otros parámetros como la conducta del estudiante, su género, edad y demás parámetros demográficos con la intención de determinar si influyen en las calificaciones y de ser así, se incluirán también como parámetros del modelo a entrenar. En el caso de los Docentes, se analizarán los promedios que obtuvieron los estudiantes en las materias dictadas por ellos en los diferentes cursos con la finalidad de tener un parámetro que mida su pedagogía, al igual que con los estudiantes, en el caso de los docentes se analizará también la incidencia de sus atributos demográficos para determinar si existe una relación significativa con los promedios obtenidos por los estudiantes en sus asignaturas. En lo que respecta a la asignatura, se considerarán los promedios obtenidos por los estudiantes en la misma asignatura a lo largo del tiempo, con la finalidad de tener un indicador de su dificultad. Por último, se analizará también la información del curso al que pertenece el estudiante con la finalidad de determinar si existe una relación entre los promedios obtenidos por todo el curso con los promedios obtenidos por un estudiante integrante del mismo; los promedios a analizar, en este caso, incluyen el promedio del rendimiento académico y el de su comportamiento.

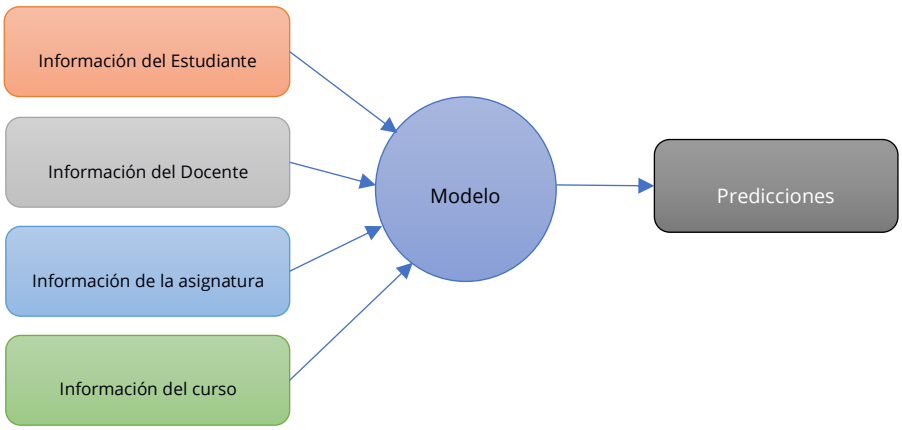


Figura 1.1 Grupos de información a ingresar al modelo.

En lo que respecta a la ejecución, el presente trabajo lo realizaremos en dos etapas: en la primera se utilizarán técnicas de análisis exploratorio de datos para la

preparación y limpieza de la información previo a la extracción de variables significativas.

Figura 1.2 Metodología para la extracción de variables significativas a utilizar en el modelo.

En la segunda etapa, utilizaremos estas variables significativas para diseñar y entrenar un modelo basado en redes neuronales que, mediante el aprendizaje automático, permita predecir las calificaciones de un estudiante. El uso de redes neuronales con este propósito no está muy difundido, debido a que la problemática de predicción académica, por su carácter temporal, por lo general se lo trata utilizando técnicas de regresión lineal, modelos probabilísticos o árboles de decisión (Hellas et al., 2018), sin embargo existen ya investigaciones que utilizan redes neuronales para la predicción del rendimiento académico, como es el caso del trabajo de Iyanda, A. R., Ninan, O. D., Ajayi, A. O., & Anyabolu, O. G. (Iyanda et al., 2018), en donde se hace una comparación del uso de dos tipos de redes neuronales con este fin: Multilayer Perceptron (MLP) y Generalized Regression Neural Network (GRNN), llegando a la conclusión de que el uso de redes neuronales para predecir el rendimiento académico de los estudiantes es completamente viable y además tiene un alto nivel de exactitud. Está también el trabajo de Lau, Sun y Yang (Lau et al., 2019) quienes diseñaron una red neuronal que, con 11 variables de entrada, 2 capas ocultas y una capa de salida, logran una exactitud de predicción del 84,8%. Debemos considerar también la investigación realizada por Tsiakmaki, Kostopoulos, Kotsiantis y Ragos (Tsiakmaki, M., Kostopoulos et al., 2020), quienes estudiaron la efectividad de transferir el aprendizaje obtenido en redes neuronales profundas a modelos de predicción académica, y demostraron que se logra un nivel aceptable de precisión siempre

que la red neuronal profunda haya sido entrenada con datos de estudiantes que han asistido a cursos similares.

Una vez construido el modelo se definirá un proceso que permita mantenerlo en constante entrenamiento, utilizando para ello el contraste entre las calificaciones predichas y las realmente obtenidas por los estudiantes en las diversas actividades planificadas.

Figura 1.3 Metodología para la mejora continua del modelo.

Para la elaboración de la herramienta de software se utilizará una arquitectura basada en microservicios con una aplicación web que los consuma.

1.6 Resultados Esperados

El proyecto generará datos estadísticos de las variables más significativas relacionadas a las calificaciones obtenidas por los estudiantes. También se entrenará un modelo que permita predecir las calificaciones que obtendrá un estudiante para luego utilizarlo en la proyección de su desempeño académico a futuro. El modelo estará en constante evolución debido a que se implementará un proceso de entrenamiento automático que se ejecutará recurrentemente.

Las proyecciones y la evolución histórica se presentarán en una herramienta de software que permitirá a los directivos tomar decisiones oportunas y así mejorar el rendimiento académico de sus estudiantes. Los resultados se segmentarán por cada uno de ellos debido a que se propone realizar un seguimiento individual de su desempeño académico, para posibilitar la toma de decisiones aplicables de manera personalizada.

1.7 Dataset

Para el proyecto se utilizarán los datos de 20 estudiantes de segundo año de bachillerato general unificado que estudian en una Unidad Educativa privada. Todos los estudiantes pertenecen al mismo paralelo y algunos de ellos cuentan con un historial de calificaciones en la institución desde el 2017. Los datos serán generados en formato CSV para ser cargados al modelo. El dataset se compone principalmente de las calificaciones obtenidas por los estudiantes a lo largo del tiempo en las diferentes actividades académicas realizadas por ellos durante su vida estudiantil. Los atributos con que cuenta el dataset son los siguientes:

Tabla 1.1 Atributos del dataset.

Atributo	Descripción
id_estudiante	Número identificador del estudiante. Por anonimización, este número es secuencial y no corresponde a ninguna identificación personal.
id_docente	Número identificador del docente. Por anonimización, este número es secuencial y no corresponde a ninguna identificación personal.
calificacion	Calificación obtenida por el estudiante. Los valores fluctúan entre 0 como la mínima calificación y 10 como la máxima.
nivel	Nivel de estudio en el que el estudiante obtuvo la calificación. Los niveles se muestran en la Tabla 1.2, niveles de estudio.
area	Área de estudio en la que se obtuvo la calificación. Los códigos se muestran en la Tabla 1.3, códigos de área de estudio.
grado	Grado en el que el estudiante obtuvo la calificación. Los grados se muestran en la Tabla 1.4, grados de estudio.
tipo_actividad	Código del tipo de actividad en la que el estudiante obtuvo la calificación. Los códigos se muestran en la Tabla 1.5, códigos de tipo de actividad.
fecha_recepcion	Fecha de la actividad en el formato año-mes-día.

edad_estudiante	Edad, en años, del estudiante a la fecha en que obtuvo la calificación. Los valores esperados para esta columna van desde 5 años que es la edad en que los estudiantes ingresan al nivel inicial y 78 que corresponde a la esperanza de vida estimada en la provincia de Santa Elena para el 2020 (Instituto Nacional De Estadística Y Censos, 2013).
genero_estudiante	Género del estudiante. Los códigos de género se muestran en la Tabla 1.6.
edad_docente	Edad, en años, del docente a la fecha en que asignó la calificación al estudiante. Los valores esperados van desde 18 años que es la edad límite para la mayoría de edad en Ecuador y 78 que corresponde a la esperanza de vida estimada en la provincia de Santa Elena para el 2020 (Instituto Nacional De Estadística Y Censos, 2013).
genero_docente	Género del docente. Los códigos de género se muestran en la Tabla 1.6.
comportamiento_estudiante	Calificación del comportamiento obtenida por el estudiante en el periodo en una escala del 0 a 1, donde 0 es la peor calificación y 1 la mejor.
comportamiento_curso	Calificación del comportamiento obtenida por el curso donde estuvo el estudiante en el periodo. El valor es un promedio de la calificación del comportamiento de los estudiantes integrantes del curso, su escala es del 0 al 1, donde 0 es la peor calificación y 1 la mejor.
dias_entre_actividades	Días transcurridos entre las actividades calificadas, independientemente del tipo de actividad. Los valores esperados van desde 0 hasta 365.
dias_entre_actividades_mismo_tipo	Días transcurridos entre actividades calificadas del mismo tipo. Los valores esperados van desde 0 hasta 365.

Tabla 1.2 Niveles de estudio.

Código	Descripción
0	Inicial
1	Preparatoria
2	Educación básica elemental
3	Educación básica media
4	Educación básica superior
5	Bachillerato

Tabla 1.3 Códigos de área de estudio.

Código	Descripción
CN	Ciencias naturales

CS	Ciencias sociales
ECA	Educación cultural y artística
EF	Educación física
ID	Interdisciplinar
LE	Lengua extranjera
LL	Lengua y Literatura
MAT	Matemáticas

Tabla 1.4 Códigos de grado de estudio.

Código	Descripción
0	Inicial
1	Primer grado
2	Segundo grado
3	Tercer grado
4	Cuarto grado
5	Quinto grado
6	Sexto grado
7	Séptimo grado
8	Octavo grado
9	Noveno grado
10	Décimo grado
11	Primero año de bachillerato
12	Segundo año de bachillerato
13	Tercer año de bachillerato

Tabla 1.5 Códigos de tipo de actividad.

Código	Descripción
AGC	Actividades grupales en clase.
AIC	Actividades individuales en clases.
EVAL	Evaluaciones parciales.
EXA	Examen Quimestral.
EXGRA	Examen de gracia.
EXREM	Examen remedial.
EXSUP	Examen supletorio.
LEC	Lecciones.
TAI	Trabajos académicos independientes.

TUTORIA	Tutorías académicas.

Tabla 1.6 Códigos de géneros.

Código	Género
F	Femenino
M	Masculino

La cantidad de registros con que se realizó el presente trabajo es 871,025 y corresponden a los periodos desde abril de 2017 hasta septiembre de 2021. Esta cantidad se incrementa mensualmente a una razón de aproximadamente 20,000 registros durante los ciclos educativos que duran por lo general 10 meses de estudio con 2 de vacaciones.

CAPÍTULO 2

2. ESTADO DEL ARTE

2.1 Fundamentos del problema

El presente trabajo es una alternativa viable para la proyección de calificaciones de estudiantes de educación general básica y bachillerato unificado, utilizando modelos de series temporales y redes neuronales. Entonces, es importante revisar algunos conceptos fundamentales que nos permitirán definir el marco teórico correspondiente.

La predicción de calificaciones es un campo de estudio que ha sido objeto de múltiples investigaciones en las que se han generado modelos que permiten estimar con un nivel aceptable de exactitud las calificaciones que obtendrán los estudiantes, para esto, los modelos utilizan valores que cuantifican factores que según diferentes estudios inciden directamente en el rendimiento académico y en su evaluación. Por ejemplo, según Saa et al. (2019), los factores que influyen de manera más significativa en el proceso de predicción del rendimiento académico son las calificaciones, la actividad de aprendizaje en línea, la información demográfica e información social de los estudiantes.

Ya en el contexto ecuatoriano, el parámetro relacionado a las actividades de aprendizaje en línea no había tenido un desarrollo significativo hasta antes del 2020, situación que cambió como producto de la pandemia causada por el COVID-19, en donde la mayoría de la población se vio obligada a utilizar los servicios relacionados al internet para el desarrollo de sus actividades diarias, entre ellas la educación, sin embargo, los parámetros para medir su incidencia son aun incipientes o en algunos casos ambiguos, por lo que no es posible considerarlos para el presente trabajo.

Por otra parte, en el caso de la información social, a pesar de que según el estudio de Saa et al. (2019) es un factor representativo, llegando a un 12%, debemos considerar que esta información debería ser capturada mediante instrumentos como encuestas a los estudiantes, la observación u otro medio que permita medir los parámetros sociales del estudiante, como la cantidad de amigos, aficiones, hábitos, etc., que permitan obtener

y alimentar el modelo con este tipo de información. Lamentablemente, para el caso que nos ocupa, esta información no está disponible, por lo que incluimos la recolección y uso de esta en nuestras recomendaciones.

También tenemos el trabajo de Amirah et al. (2015)., quienes consideran que los factores que inciden directamente en el rendimiento académico de los estudiantes son las calificaciones, los factores demográficos, factores sociales y psicométricos, coincidiendo en parte con las conclusiones de Saa et al. (2019).

Un caso interesante es la investigación de Anupama y Vijayalakshmi (2012), quienes utilizando un modelo basado en redes neuronales obtuvieron un 98% de exactitud en sus pronósticos, utilizando como parámetros solamente las calificaciones de los estudiantes. Este resultado sumado a la gran significancia de los factores de calificaciones y demográficos reportada por Saa et al. (2019) en su investigación, nos permiten confiar en que se pueden obtener resultados significativos en la predicción de calificaciones con los ítems de información con que contamos en nuestro dataset.

Por último, debido a la pandemia causada por el COVID-19, el comportamiento de la mayoría de las actividades susceptibles de medición, como es el caso del rendimiento académico varió significativamente, lo que causó que los datasets producto de estas actividades tengan comportamientos atípicos. Por este motivo, múltiples investigadores han realizado estudios para medir la incidencia de estas variaciones en los procesos que los utilizan. Un ejemplo de estos estudios lo realizó Srivastava (2021), concluyendo que las predicciones realizadas mediante redes neuronales tuvieron un mejor desempeño aun con los efectos de la pandemia y, que esta tuvo un impacto negativo en la salud psicológica y en el rendimiento académico de los estudiantes.

2.2 Soluciones de analítica y aprendizaje relacionadas al problema

La problemática de la predicción del rendimiento estudiantil tiene dos componentes principales, a saber: los factores que afectan el rendimiento estudiantil y los algoritmos utilizados para identificar y utilizar estos factores en la predicción. Existen múltiples estudios que identifican los factores que afectan el rendimiento, Hellas et al. (2018) ubican el rendimiento previo, el compromiso, factores psicométricos y factores

demográficos como los más frecuentemente utilizados para la predicción, Saa et al. (2019) incluyen el rendimiento previo, el aprendizaje en línea, factores demográficos y sociales dentro de los más utilizados, por último, Amirah et al. (2015) muestran que factores relacionados al rendimiento previo y a la información demográfica del estudiante son los más frecuentemente utilizados por los algoritmos que obtuvieron mayor exactitud en la predicción. Lo anterior evidencia que el rendimiento previo y la información demográfica del estudiante son los factores más frecuentemente utilizados y que contribuyen más significativamente en la predicción del rendimiento académico.

En lo que respecta a los algoritmos para la predicción, Hellas et al. (2018) demuestran en su trabajo que los algoritmos estadísticos y de clasificación son los más frecuentemente utilizados, dentro de estos se incluyen el modelamiento lineal, los árboles de decisión y las redes neuronales, por otra parte, Saa et al., (2019) en su estudio muestran que el 24,8% de los investigadores utilizan árboles de decisiones, seguido por clasificadores Naïve Bayes y las Redes Neuronales. Ya en lo referente a la exactitud de los modelos, Amirah et al. (2015) concluyen que los algoritmos con mejor rendimiento fueron las redes neuronales y los árboles de decisión, con un 98% y 90% de exactitud respectivamente. Por otra parte, Sivasakthi (2017) concluye en su investigación que una red neuronal MLP permitió obtener un 93% de exactitud en sus predicciones, superando a los árboles de decisión, los cuales obtuvieron un 91% de exactitud. Estos datos nos permiten concluir que sin bien es cierto que la predicción del rendimiento estudiantil ha sido tradicionalmente realizada utilizando métodos estadísticos (Hellas et al. (2018)), existen varios estudios en los que se han obtenido niveles de exactitud excepcionales utilizando algoritmos de clasificación como las redes neuronales, árboles de decisión, Naïve Bayes, etc.

El uso de redes neuronales para la solución de problemas de predicción no es algo nuevo, tampoco lo es su aplicación en el problema de la predicción del rendimiento estudiantil, si embargo, como ya se mencionó antes, su utilización en este ámbito es aun incipiente si lo comparamos con los modelos estadísticos o de árboles de decisión. Por otra parte, debemos considerar que la capacidad que tienen las redes neuronales de aprender a reconocer patrones, memorizarlos o incluso generarlos, le dan a este tipo de

redes un especial potencial para su aplicación en el campo que motiva el presente trabajo.

Dentro de las diferentes arquitecturas de redes neuronales que podrían utilizarse en la predicción del rendimiento estudiantil tenemos las Redes Neuronales Recurrentes (RNN). Esta arquitectura posee un mecanismo de memoria de corto y largo plazo llamado "*long short-term memory*" (LSTM por su siglas en inglés) lo que hace de este tipo de redes una excelente candidata para utilizarla en nuestro trabajo, sin embargo, las RNN tienen un problema relacionado a la perdida de peso que sufren los datos iniciales cuando la cantidad de éstos es muy larga; este inconveniente también es llamado "*desvanecimiento de gradiente*" (Muñoz et al. (2019)), y es una importante limitante que, en nuestro caso, no permite su utilización debido a que la cantidad de información procesada es extensa.

En el año 2017, se publicó el paper llamado "*Attention is all you need*" (Vaswani et al. (2017)). En este documento se propone una arquitectura de redes neuronales diferente llamada "*transformers*". Esta arquitectura tiene como principal característica que soluciona el problema del "*desvanecimiento de gradiente*" mediante la implementación de bloques especiales de procesamiento de información llamados "*mecanismos de atención*", los cuales permiten analizar la secuencia completa y detectar relaciones entre los datos, asignando a estas un peso que define su importancia en el contexto.

2.2.1 Series temporales

Podemos conceptualizar a las series temporales como conjuntos temporalmente ordenados de observaciones que pueden ser discretas o continuas y por lo general tienen una frecuencia fija, lo que significa que las observaciones ocurren en intervalos definidos de tiempo, Fierro A. A. (2021).

Entonces, podemos decir que una serie temporal Z se compone del conjunto de observaciones de una variable en los instantes de tiempo t desde 1 hasta T:

$$Z = \{Z_t\}_{t=1}^{T} \tag{2.1}$$

Componentes de las series temporales

Las series temporales tienen algunos componentes, los cuales nos permiten modelarlas matemáticamente, estos componentes son:

- **La tendencia (T)**: refleja el crecimiento o disminución de la serie en el tiempo.
- **El componente cíclico (C)**: refleja el comportamiento de la serie a lo largo de la tendencia, normalmente se muestra en la forma de una onda.
- **El componente estacional (S)**: contiene el comportamiento repetitivo de la serie en intervalos fijos de tiempo.
- **El componente aleatorio (R)**: contiene los residuos de la serie una vez que se han retirado los componentes de tendencia, cíclico y estacional. Se caracteriza por su aleatoriedad lo que lo hace difícil de modelar.

Dependiendo de cómo se combinen estos componentes podemos definir tres tipos de series temporales:

- **Series temporales aditivas:** En una serie temporal aditiva se suman los componentes de tendencia, cíclico, estacional y aleatorio:

$$Z_t = T_t + C_t + S_t + R_t \tag{2.2}$$

- **Series temporales multiplicativa:** En una serie temporal multiplicativa se multiplican los componentes de tendencia, cíclico, estacional y aleatorio:

$$Z_t = T_t * C_t * S_t * R_t \tag{2.3}$$

- **Series temporales mixtas:** En una serie temporal mixta se pueden combinar las operaciones de suma o multiplicación, por lo general se suma el componente aleatorio y se multiplican el resto:

$$Z_t = T_t * C_t * S_t + R_t$$

(2.4)

Series temporales "*univariadas*" y "*multivariadas*"

Cuando la serie temporal contiene los valores de solo una variable o característica se denomina "*univariada*", sin embargo, en muchos casos, un fenómeno se explica no solo por los valores de la variable de estudio sino también por la relación de esta con otras variables relevantes, este es el caso de las series temporales "*multivariadas*", las cuales consideran la interrelación de una serie temporal objeto de estudio con otras series temporales que explican el comportamiento de la primera. Podemos expresar las relaciones de la variable de estudio con las variables explicativas de la siguiente manera:

$$y = x_1 b_1 + x_2 b_2 + \cdots + x_k b_k + e$$

(2.5)

Donde y es la variable de estudio, $x_1, x_2, \ldots, x_k$ son las variables explicativas, b son los pesos de cada variable explicativa y e es un error aleatorio.

Dado que al observar el fenómeno obtenemos valores tanto de y como de x, podemos decir que cada observación es:

$$\{y_i, x_{1i}, x_{2i}, \ldots, x_{ki}\}_{i=1}^{n}$$

(2.6)

Donde y es la variable de estudio, x son las variables explicativas, k es el número de variables explicativas y n es el número de observaciones.

Considerando esto, podemos decir que:

$$y_i = x_{1i} b_1 + x_{2i} b_2 + \cdots + x_{ki} b_k + e$$

(2.7)

Para nuestro caso, la variable de estudio es la calificación del estudiante, y las variables explicativas son las variables contenidas en el dataset y que permiten explicar el valor de la calificación.

2.2.2 Arquitectura de redes neuronales tipo "*Transformer*"

La arquitectura de redes neuronales tipo "*Transformer*" fue definida en el paper denominado "*Attention is all you need*" (Vaswani et al. (2017)), en un inicio fueron utilizadas ampliamente para el procesamiento de lenguaje natural (NLP por sus siglas en inglés) como una mejora a las redes neuronales recurrentes (RNN) solucionando algunas limitantes de éstas últimas, tales como el procesamiento en serie de la información o el problema del "*desvanecimiento de gradiente*". La solución de estas limitantes se realiza mediante la implementación del procesamiento en paralelo de la información, además de un mecanismo especial de análisis de secuencias llamado "*mecanismo de atención*".

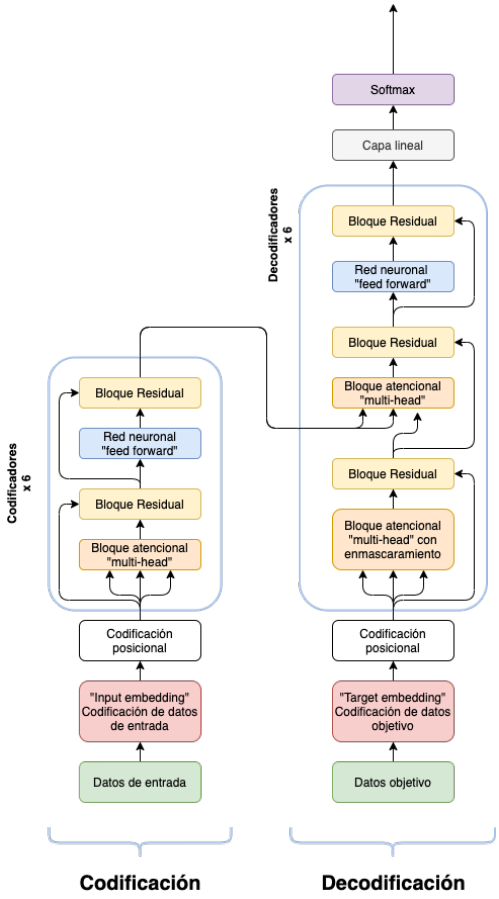


Figura 2.1 Diagrama de bloques de una arquitectura de red neuronal tipo "Transformer".

La figura 2.1 muestra un diagrama de bloques de la arquitectura de red neuronal tipo "*Transformer*", este diagrama se basa en el paper original de Vaswani et al, y nos servirá

para identificar y describir la interacción de su diferentes componentes. En la figura podemos apreciar dos etapas que se complementan entre si y agrupan dentro de ellas a los componentes de la red. A estas etapas las hemos denominado: “Codificación” y “Decodificación” y detallaremos su contenido y funcionamiento a continuación.

2.2.2.1 Codificación

La etapa de “Codificación” se encarga de extraer las relaciones más relevantes entre los datos de entrada, incluyendo entre estos datos al valor a predecir. Las relaciones encontradas se representan en la forma de un conjunto de vectores con la información de la relevancia de cada dato para con el resto. Los vectores que contienen esta información toman los nombres de: vector “*Queries*” (Q), “*Keys*” (K) y “*Values*” (V), al final de esta etapa se obtiene un conjunto de vectores con la información de la relevancia de cada dato dentro de la secuencia, estos vectores serán pasados a la etapa de “*Decodificación*” para finalizar el entrenamiento del modelo utilizando un conjunto adicional de datos de entrada a esta última etapa.

En las siguientes líneas veremos los bloques que componen la etapa de “Codificación”.

Codificación de datos de entrada

Al igual que en otras redes neuronales, la secuencia de datos debe ser representada numéricamente antes de ser procesada, esta acción se realiza en este bloque. Tanto los valores como su equivalencia se incluyen en un diccionario que nos servirá para decodificar la salida resultante del modelo.

Codificación posicional

Debido a la forma de proceso de la información, es importante codificar la posición de cada dato en la secuencia, esta codificación se realiza en este bloque para lo cual se toman los vectores resultantes del bloque de “*Codificación de datos de entrada*” y se les suma otros vectores que codifican la posición de cada dato en la seria, utilizando para ello funciones senoidales para las posiciones pares y cosenoidales para las impares. Como resultado de este bloque tenemos un conjunto de vectores que contienen la información de la serie de datos, así como la posición de cada dato en la secuencia.

Codificadores

Los vectores resultantes de las etapas iniciales se ingresan a bloques denominados "*Codificadores*", estos bloques se encargan de extraer la información más significativa de la secuencia de datos, poniendo "*Atención*" en aquellas relaciones con mayor relevancia. El paper original considera 6 bloques de este tipo en su diseño.

La estructura de todos los bloques "*Codificadores*" es idéntica entre ellos y tiene los siguientes componentes:

Bloque atencional "*multi-head*"

Este bloque analiza todos los datos de la secuencia y encuentra relaciones entre ellos. Para esto, se hace uso de tres representaciones alternativas del vector original denominadas "*Queries*", "*Keys*" y "*Values*" en *h* espacios lineales denominados "*multi-head*" en el paper original. Las diferentes representaciones las obtienen tres redes neuronales especializadas en su cálculo. Los espacios lineales, en cambio, permiten realizar los cálculos desde diferentes perspectivas evitando la preponderancia de un solo dato para con los otros. Al final, los resultados calculados en cada espacio lineal se concatenan en los vectores resultantes.

Los vectores "*Queries*" y "*Keys*" son utilizados para calcular el grado de asociación entre pares de palabras mediante una multiplicación entre ellos. El resultado se escala de acuerdo con el tamaño de cada vector y luego mediante una función *softmax* se obtiene la probabilidad de cada valor entre 0 y 1. Un valor cercano a cero indica que la relación es poco relevante y un valor cercano a 1 indica que es muy relevante. Luego, se multiplica la matriz resultante por el vector "Values" con esto se logra obtener un nuevo vector que contiene la información de la relevancia de cada dato con respecto al resto.

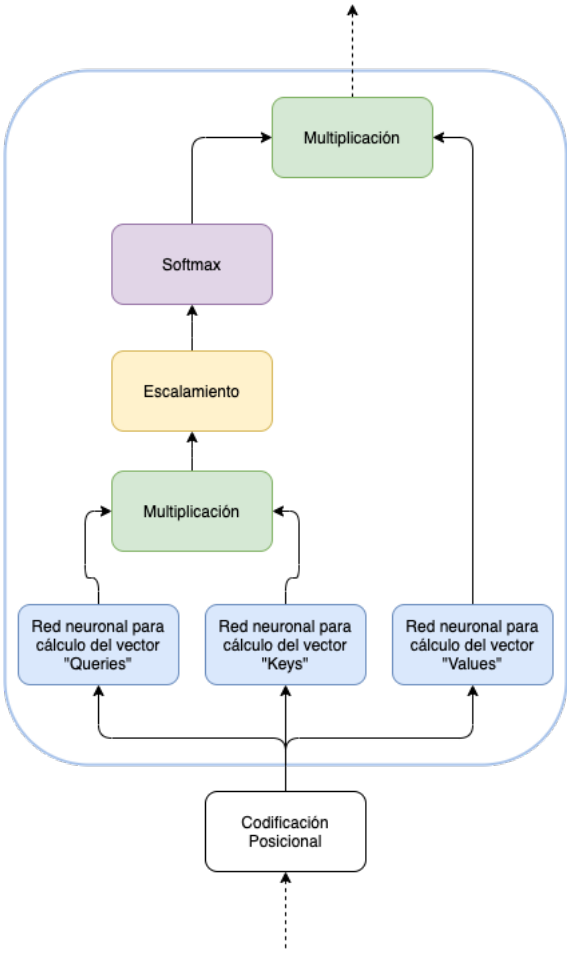


Figura 2.2 Diagrama de bloques del "Bloque atencional".

El resultado del proceso anterior es un vector que contiene la información de la relevancia de cada elemento de la serie para con el resto. Al utilizar múltiples bloques atencionales se logra obtener relaciones incluso entre grupos de datos.

La representación matemática del bloque atencional se muestra en la ecuación 2.8.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.8}$$

Bloque residual

La salida del "*Bloque atencional*" y la salida del "Blo*que de codificación posicional*" ingresan a un nuevo bloque denominado "*Bloque residual*", aquí se suman los dos vectores y se normaliza el resultado antes de ingresarlos al siguiente bloque. Todos los bloques residuales de la arquitectura tienen la misma estructura y cumplen la misma función.

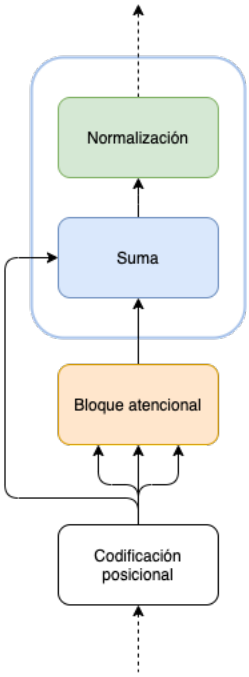


Figura 2.3 Diagrama de bloques del "Bloque residual".

*Red neuronal "*feed forward*"*

El último componente del "*Codificador*" es una red neuronal que toma los vectores con la información atencional de los bloques anteriores y los consolida en una única salida. La salida de esta red y su entrada se envían a un nuevo "*Bloque residual*" con la finalidad de normalizar la información antes de ser enviada los siguientes bloques.

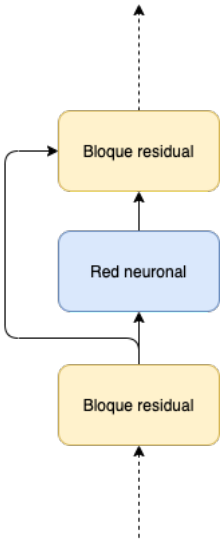


Figura 2.4 Diagrama de bloques de la "Red neuronal" de un bloque "Codificador".

2.2.2.2 Decodificación

La etapa de "*Decodificación*" utiliza una secuencia de "*Datos objetivo*" para entrenar el modelo en obtener la predicción deseada. En este entrenamiento es fundamental el uso de los vectores generados en la etapa de "*Codificación*".

A continuación, revisaremos los bloques que componen la etapa de "*Decodificación*" y analizaremos con más detalle el tratamiento que se le da a los "*Datos objetivo*" y a los vectores obtenidos desde la etapa anterior.

Codificación de datos objetivo

Este bloque se encarga de codificar en valores numéricos la información de la secuencia de "Datos objetivo", en este bloque se genera un diccionario de datos de salida que se utilizará luego en el momento de realizar la predicción. Los vectores resultantes se pasarán al siguiente bloque para agregar la información posicional.

Codificación posicional

Al igual que en la información que alimenta la etapa de "*Codificación*", en esta etapa también se debe codificar la ubicación de cada dato en la serie. El algoritmo utilizado es el mismo que se usa en la etapa antes mencionada.

Decodificadores

La capa de decodificación cuenta con bloques de "*Decodificadores*" que funcionan de manera similar a los bloques "*Codificadores*" de la etapa de "*Codificación*"; con algunas diferencias que las detallaremos a continuación mientras describimos su estructura. El paper original considera 6 bloques de este tipo en su diseño.

*Bloque atencional "*multi-head*" con enmascaramiento*

Este bloque tiene la misma función que el bloque atencional descrito en la etapa de "*Codificación*" incluyendo también la separación de los cálculos en *h* espacios lineales, con la diferencia de que en esta capa se utiliza un enmascaramiento para evitar que el algoritmo "*vea a futuro*", dado que el entrenamiento se realiza de manera secuencial en la serie, es importante que al predecir el dato x_n, la red solo tenga acceso a las probabilidades de los datos anteriores a este.

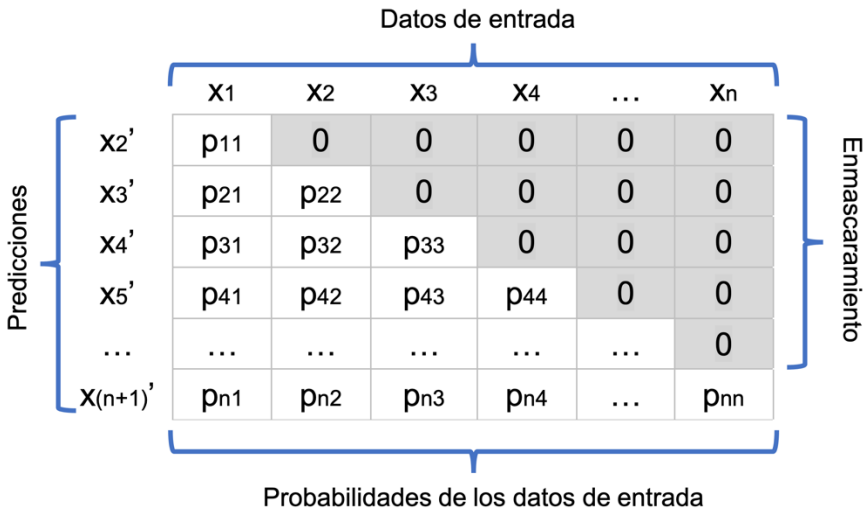


Figura 2.5 Enmascaramiento de las probabilidades del bloque atencional.

La figura 2.5 muestra como el enmascaramiento se logra colocando un 0 en las probabilidades de los datos ubicados en una posición mayor o igual a la del dato a predecir, así se evita que el entrenamiento se vea afectado por datos ubicados "*en el futuro*" de la serie de datos.

Bloques residuales

Los bloques residuales que se encuentran en esta etapa, tres en total, tienen una estructura y función idéntica a la descrita en los bloques residuales de la etapa de "Codificación".

Bloque atencional "*multi-head*"

Este bloque atencional, al igual que los descritos anteriormente, extrae las relaciones más relevantes en el conjunto de datos de entrada. Por ende, utiliza los vectores *Q*, *K* y *V en h espacios lineales*, sin embargo, la diferencia radica en que los vectores *Q* y *K* se calculan en base a la información resultante de la etapa de "*Codificación*" mientras que el vector *V* se calcula con la información obtenida desde el "*Bloque atencional con enmascaramiento*" descrito anteriormente, así se logra entrenar el modelo con la información de los "*Datos objetivo*".

Red neuronal "*feed forward*"

En este caso también, el último componente del "*Decodificador*" es una red neuronal que toma los vectores con la información atencional de los bloques anteriores y los consolida en una única salida. La salida de esta red y su entrada se envían, al igual que en los "*Decodificadores*", a un "*Bloque residual*" con la finalidad de normalizar la información antes de ser enviada los siguientes bloques

Capas lineal y Softmax

Esta capa contiene una red neuronal encargada de transformar los vectores de entrada en un solo vector con una cantidad de elementos igual a la cantidad de valores a predecir. El vector resultante se ingresa a una capa *Softmax* que se encarga de calcular las probabilidades de cada uno de los elementos del vector. La posición con mayor probabilidad será el resultado de la predicción. Para convertir esta posición se utiliza el diccionario obtenido en la capa de "*Codificación de datos objetivo*".

2.2.3 Implementación de series temporales con redes neuronales tipo "*transformer*"

Para efectos del presente trabajo, realizaremos una implementación de series temporales utilizando la arquitectura de redes neuronales llamada "*transformers*" con la finalidad de aprovechar las características que tienen al analizar grandes cantidades de datos. En la sección "*Arquitectura de redes neuronales tipo Transformer*" analizamos al detalle su estructura y propiedades.

Las redes "*transformer*" se han aplicado principalmente en el procesamiento de lenguaje natural (NLP por sus siglas en inglés) por su capacidad de procesamiento en paralelo, lo que permite encontrar relaciones entre los datos independientemente de su ubicación en la secuencia. Sin embargo, en una serie temporal, la ubicación del dato es parte de la información que debe considerarse al analizar la secuencia, por esta razón, es importante codificar la ubicación temporal del dato como parte de la información a analizar. Esto se logra en el momento de construir el token que analizará la red neuronal. Más adelante, en el capítulo 3, analizamos más a detalle la construcción de los tokens y la codificación de la información temporal en ellos.

2.3 Fuentes de datos relacionadas al problema

Tradicionalmente los datasets para la predicción del rendimiento académico estudiantil se componen de archivos planos generados por procesos que compilan la información desde diferentes fuentes, tales como sistemas académicos, sistemas LMS, redes sociales, entre otros. Es una práctica común la "*anonimización*" de estos para proteger la privacidad de las personas relacionadas a los datos contenidos en los archivos. Esta anonimización es particularmente importante cuando las personas involucradas son menores de edad.

La información utilizada para este trabajo proviene de la base de datos de una entidad educativa privada de educación primaria y secundaria. Los datos corresponden al periodo desde el 2017 hasta el 2021, y por tratarse de información sensible, los atributos relacionados a la identificación de los estudiantes y docentes han sido anonimizados

para efectos de este estudio, para ello se ha firmado el correspondiente acuerdo de confidencialidad con la entidad que nos proporcionó la información.

2.4 Librerías y software a utilizar

En el presente trabajo utilizaremos el siguiente conjunto de herramientas informáticas:

2.4.1 Extracción de datos para el entrenamiento

La base de datos en la que está almacenada la información es PostgreSQL versión 10, ejecutándose en servidores linux CentOS 7 en la nube. Los datos se extraerán en la forma de archivos CSV y el contenido será el resultado de la ejecución de una sentencia SQL creada específicamente para este trabajo. El procedimiento que utilizaremos se muestra en el siguiente enlace de la documentación oficial de PostgreSQL:

https://www.postgresql.org/docs/13/sql-copy.html

2.4.2 Diseño y entrenamiento del modelo

Para diseñar y entrenar el modelo se utilizará Python 3 con el apoyo de librerías especializadas como numpy, matplotlib, seaborn, pandas, keras, entre otras.

2.4.3 Desarrollo de la herramienta de software interactiva

El desarrollo de la herramienta interactiva para la consulta de resultados se realizará utilizando la siguiente arquitectura:

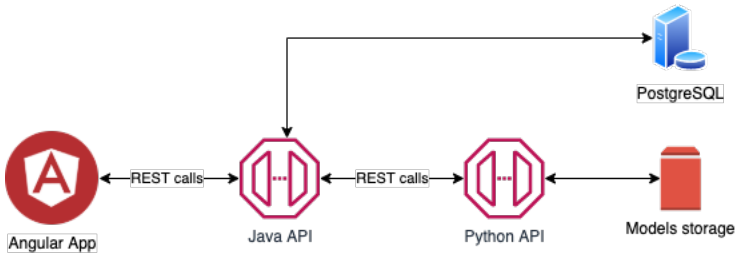


Figura 2.6 Arquitectura de la herramienta de software interactiva.

En la arquitectura de la herramienta interactiva destacan los siguientes componentes:

Tabla 2.1 Componentes de la arquitectura de la herramienta de software interactiva.

Componente	Descripción
PostgreSQL	Base de datos que almacena la información académica de la institución. Ésta es la fuente desde donde se extraerán el dataset que utilizaremos en el entrenamiento del modelo de predicción.
Models storage	Almacenamiento de los modelos de predicción entrenados.
Python API	API REST desarrollada en Python, para la interacción con el modelo.
Java API	Api desarrollada en Java para la interacción con la Python API, con la base de datos PostgreSQL y con la aplicación web.
Angular APP	Aplicación web interactiva desarrollada en Angular para la consulta de resultados.

Para el desarrollo de esta arquitectura se utilizarán las siguientes librerías y paquetes de software:

Tabla 2.2 Librerías y paquetes de software a utilizar en la arquitectura de la herramienta de software interactiva.

Componente	Librerías y paquetes de software
PostgreSQL	PostgreSQL versión 10 para Linux
Models storage	Almacenamiento en el sistema de archivos
Python API	Python versión 3 con Django Framework
Java API	Java versión 11 con Spring Framework 5
Angular App	Angular 12 con PrimeNG 12

CAPÍTULO 3

3. DISEÑO E IMPLEMENTACIÓN

3.1 Exploración y validación de datos y fuentes.

En el numeral 1.7 describimos el dataset con el que se realiza el presente trabajo. El conjunto de datos se compone de información referente a: calificaciones, niveles de estudio, edades, géneros, comportamientos y separación temporal entre actividades. En este apartado describiremos las acciones realizadas para la exploración de la información contenida en el dataset y la selección de los atributos que utilizaremos para la definición y entrenamiento del modelo de predicción.

3.1.1 Valores inexistentes

En la carga de información, existieron atributos que vinieron sin un valor específico, para efecto de este trabajo asumiremos que los atributos con esta característica tienen el valor NaN. La tabla 3.1 muestra la lista de atributos con este tipo de valores.

Tabla 3.1 Valores inexistentes.

Atributo	Cantidad	Porcentaje
edad_estudiante	91,052	10.45 %
dias_entre_actividades	38,219	4.39 %
dias_entre_actividades_mismo_tipo	89,154	10.24 %

Podemos apreciar que el atributo edad_estudiante tiene la mayor cantidad de datos inexistentes con 91,052 registros que representan el 10.45 % del total de registros existentes en el dataset, seguida por los atributos dias_entre_actividades_mismo_tipo y dias_entre_actividades.

3.1.2 Atributos con valores atípicos

Adicionalmente, se realizó el análisis de atributos con valores fuera de las categorías o rangos de valores esperados. El detalle de los valores esperados por cada atributo se muestra en el numeral 1.7. Los resultados obtenidos son los siguientes.

Tabla 3.2 Valores máximos y mínimos de los atributos numéricas.

Atributo	Esperados		Existentes		Cantidad de registros no válidos
	Mín.	Máx.	Mín.	Máx.	
calificacion	0.00	10.00	0.01	10.00	
nivel	1	5	1	5	
grado	1	13	1	13	
edad_estudiante	5	78	-3	1,812	3,510
edad_docente	18	78	27	48	
comportamiento_estudiante	0.00	1.00	0.20	1.00	
comportamiento_curso	0.00	1.00	0.80	1.00	
dias_entre_actividades	0	365	0	30	
dias_entre_actividades_mismo_tipo	0	365	0	30	

Como se puede apreciar en la tabla 3.2, el único atributo que tiene valores fuera del rango esperado es edad_estudiante con 3,510 registros no válidos, esto representa el 0.4 % del total de registros existentes en el dataset.

En lo que respecta a los atributos categóricos, no se encontraron registros con valores diferentes a los valores esperados.

3.1.3 Acciones correctivas sobre los datos atípicos encontrados

En vista de que el único atributo con valores atípicos fue edad_estudiante, se procedió a asignar el valor NaN a los valores atípicos. Es importante resaltar que en los análisis posteriores que involucran a este atributo no se consideran los valores inexistentes. A pesar de que en el dataset utilizado en el presente trabajo no se encontraron datos atípicos en la edad del docente (edad_docente), en el script se incluyeron instrucciones para limitar la edad de los docentes a un rango de entre 18 y 78 años. Se hizo esto debido a que el script se utilizará en el procesamiento de datasets que provendrán desde otras fuentes, lo cual podría ocasionar inconsistencias en esta información.

3.1.4 Atributos adicionales

Se procedió a crear atributos adicionales para analizar la información contenida en el dataset. Una vez terminado el análisis se eliminarán aquellos que no sean utilizables en la generación del modelo. Se crearon los siguientes atributos:

Tabla 3.3 Atributos creados para el análisis de la información.

Atributo	Descripción
en_pandemia	Define si los datos corresponden a un periodo de pandemia causada por el COVID-19 o no. La fecha que se utiliza para el límite es: 13 de marzo del 2020.
semana_recepcion	Número de semana en el periodo lectivo al que corresponda la actividad. La semana se calcula dividiendo para 7 la cantidad de días transcurridos entre el inicio del periodo lectivo y la fecha de la actividad. De esta división se extrae la parte entera para luego incrementarle 1.
mes_recepcion	Número de mes en el periodo lectivo al que corresponda la actividad. La numeración de meses se realizó considerando el mes de abril como el mes inicial y terminando en el mes de marzo como mes número 12. Esto con la finalidad de representar el periodo lectivo en un la secuencia de meses. Abril: 1 Mayo: 2 Junio: 3 Julio: 4 Agosto: 5 Septiembre: 6 Octubre: 7 Noviembre: 8 Diciembre: 9 Enero: 10 Febrero: 11 Marzo: 12
calificacion_previa	Calificación obtenida en una actividad inmediatamente anterior a la actual. Para la primera actividad este valor esta vacío.
calificacion_promedio	Calificación promedio de las actividades anteriores. Para la primera actividad este atributo esta vacío.

3.1.5 Prueba de normalidad

Para la prueba de normalidad de los atributos se utilizó el test de *Kolmogorov Smirnov* por tratarse de un conjunto considerable de datos. La prueba concluyó en que todos los atributos no tienen una distribución normal. Esto se debe a la gran cantidad de datos que tiene el dataset.

3.1.6 Estadística descriptiva

Tabla 3.4 Parámetros de estadística descriptiva.

Atributo	Media	Mediana	Mínimo	Máximo
calificacion	8.64	9.20	0.01	10.00
nivel	3.09	3.00	1.00	5.00
grado	6.29	6.00	1.00	13.00

edad_estudiante	10.66	10.00	5.00	36.00
edad_docente	37.86	38.00	27.00	48.00
comportamiento_estudiante	0.95	1.00	0.20	1.00
comportamiento_curso	0.95	0.97	0.80	1.00
dias_entre_actividades	6.26	3.00	0.00	30.00
dias_entre_actividades_mismo_tipo	9.09	6.00	0.00	30.00
semana_recepcion	20.06	19.00	1.00	46.00
mes_recepcion	5.99	6.00	1.00	12.00

Tabla 3.5 Parámetros de estadística descriptiva (continuación).

Atributo	Grado de divergencia	Varianza	Desviación estándar	Coeficiente de variación
calificacion	9.99	3.18	1.78	0.21
nivel	4.00	1.04	1.02	0.33
grado	12.00	8.88	2.98	0.47
edad_estudiante	31.00	8.64	2.94	0.28
edad_docente	21.00	27.64	5.26	0.14
comportamiento_estudiante	0.80	0.01	0.09	0.10
comportamiento_curso	0.20	0.00	0.05	0.05
dias_entre_actividades	30.00	47.73	6.91	1.10
dias_entre_actividades_mismo_tipo	30.00	74.80	8.65	0.95
semana_recepcion	45.00	136.31	11.68	0.58
mes_recepcion	11.00	7.54	2.75	0.46

Las tablas 3.4 y 3.5 muestran los valores de algunos parámetros estadísticos básicos para los atributos numéricos del dataset, podemos destacar que los atributos relacionados a la separación temporal entre las actividades (dias_entre_actividades y dias_entre_actividades_mismo_tipo) presentan una alta variabilidad, en contraste con los atributos relacionados al comportamiento del estudiante, los cuales presentan una baja variabilidad. El resto de los atributos presentan un coeficiente de variación intermedio.

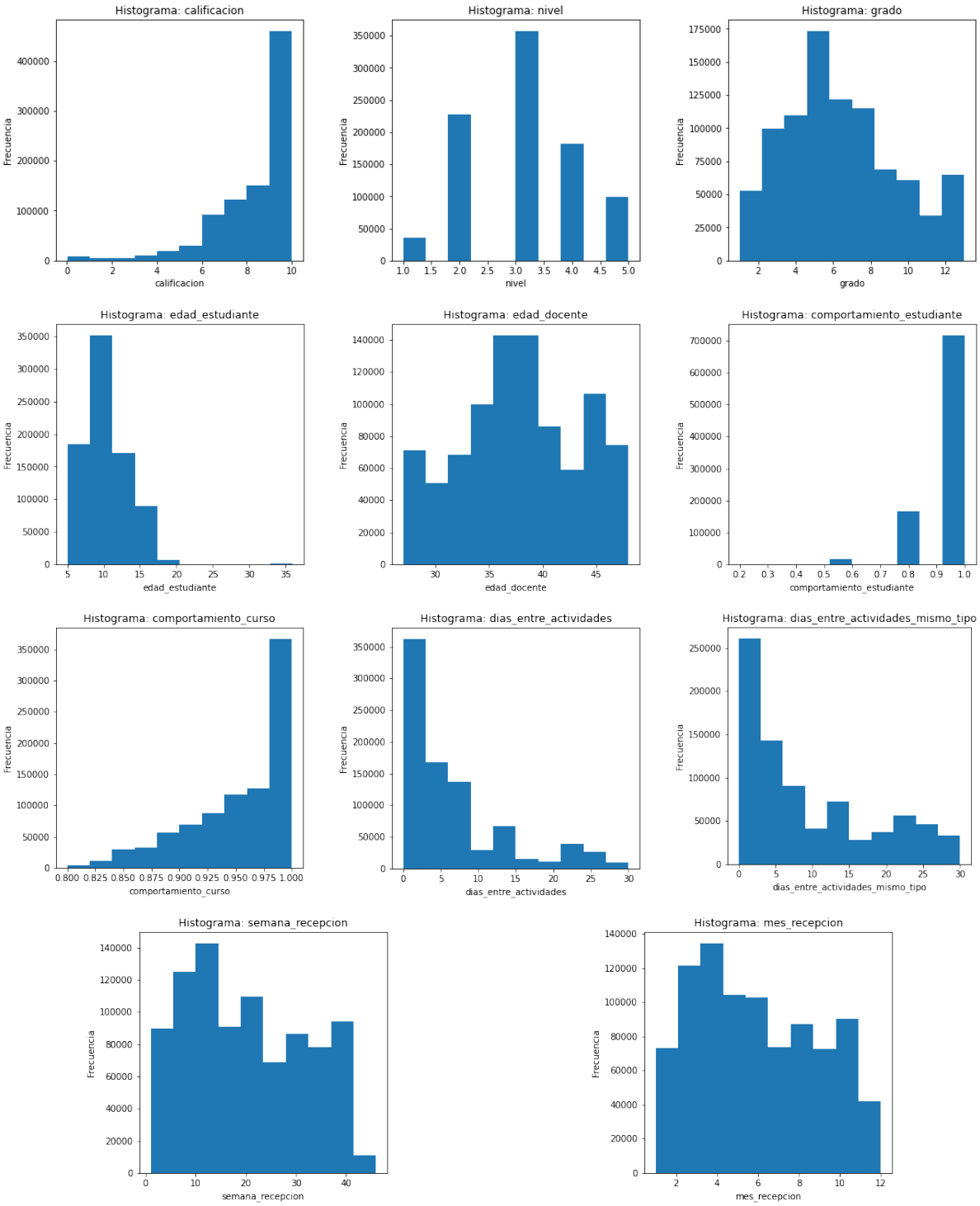


Figura 3.1 Histogramas de los atributos numéricos.

La figura 3.1 muestra los histogramas de los atributos numéricos, podemos observar que la variable que nos interesa predecir tiene un marcado sesgo en las calificaciones más altas, algo similar sucede en los atributos relacionados al comportamiento. En los

atributos relacionados a la separación temporal entre actividades podemos observar lo contrario, el sesgo es hacia la menor cantidad de días. En lo que respecta a la distribución temporal de las actividades (semana_recepcion y mes_recepcion), podemos observar un comportamiento uniforme.

3.1.7 Correlación entre atributos numéricos

Para analizar la correlación entre atributos numéricos generamos un gráfico de correlación, el cual se muestra en la figura 3.2.

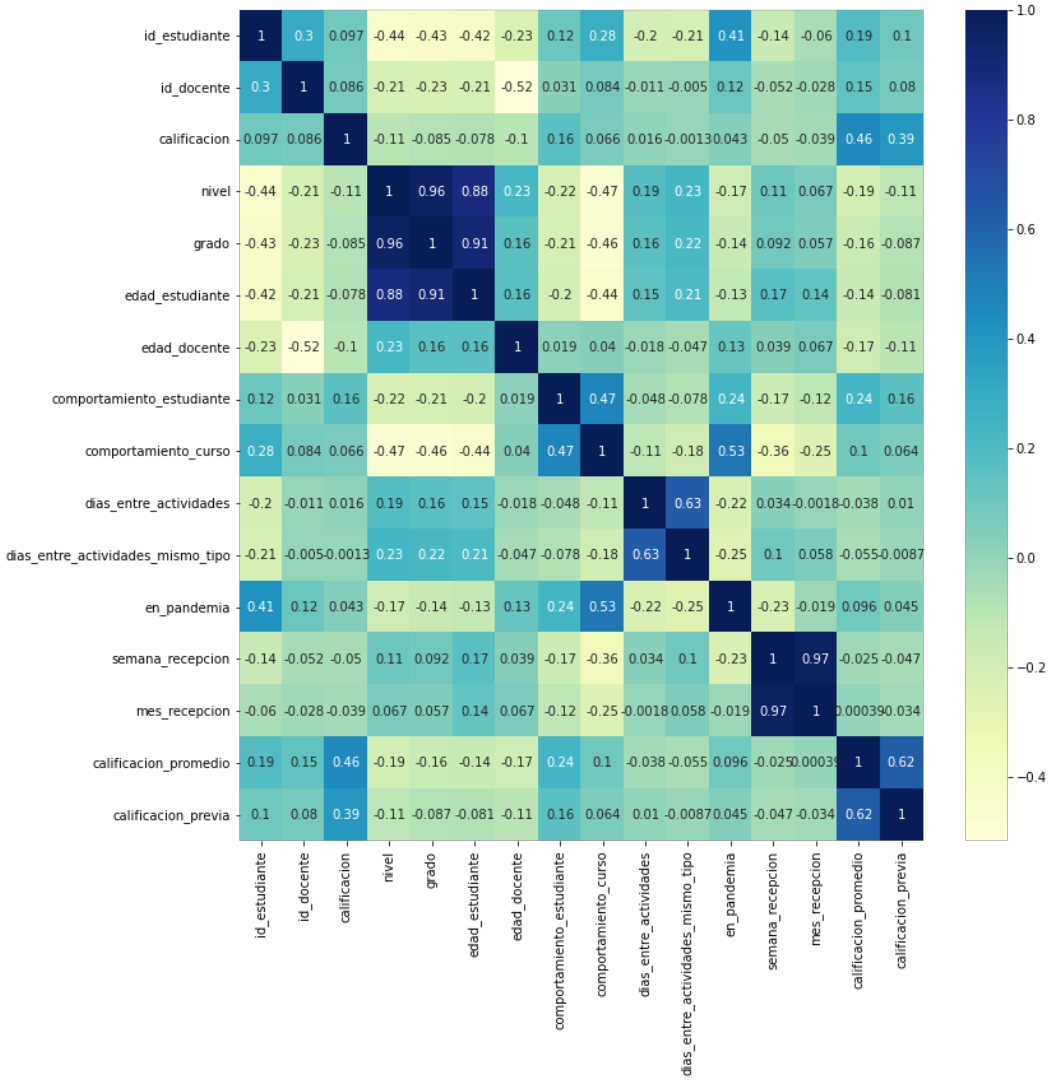


Figura 3.2 Gráfico de correlación entre atributos numéricos.

En este gráfico podemos apreciar una marcada correlación entre los siguientes atributos:

Tabla 3.6 Correlación más fuertes entre atributos.

Atributos	Coeficiente de correlación
nivel y grado	0.97
mes_recepcion y semana_recepcion	0.97
edad_estudiante y grado	0.92
edad_estudiante y nivel	0.88
dias_entre_actividades_mismo_tipo y dias_entre_actividades	0.64
id_docente y edad_docente	-0.50
comportamiento_curso y comportamiento_estudiante	0.49
comportamiento_curso y en_pandemia	0.49
id_estudiante y en_pandemia	0.43

3.1.8 Correlaciones con la variable a predecir: calificación

Los atributos que presentan una correlación significativa con el atributo a predecir son calificacion_promedio y calificacion_previa con 0.46 y 0.39 respectivamente, lo que indica que el atributo calificacion tiene una fuerte correlación con las calificaciones obtenidas por el estudiante en actividades previas. También podemos apreciar que la correlación con otros atributos es menor, aunque no se puede despreciar, como es el caso del atributo comportamiento_estudiante con un coeficiente de correlación de 0.17.

En lo que respecta al resto de atributos debemos resaltar que existen correlaciones muy altas, como es el caso de nivel y grado, o edad_estudiante y grado, lo que nos permite depurar la cantidad de atributos que se utilizarán para la generación del modelo de predicción.

En base a las correlaciones mas significativas, listadas en la tabla 3.6, procedimos a eliminar los atributos: id_estudiante, nivel, edad_estudiante, dias_entre_actividades_mismo_tipo, mes_recepcion**,** comportamiento_curso y id_docente. La figura 3.3 muestra el gráfico de correlación luego de eliminar los atributos correlacionados.

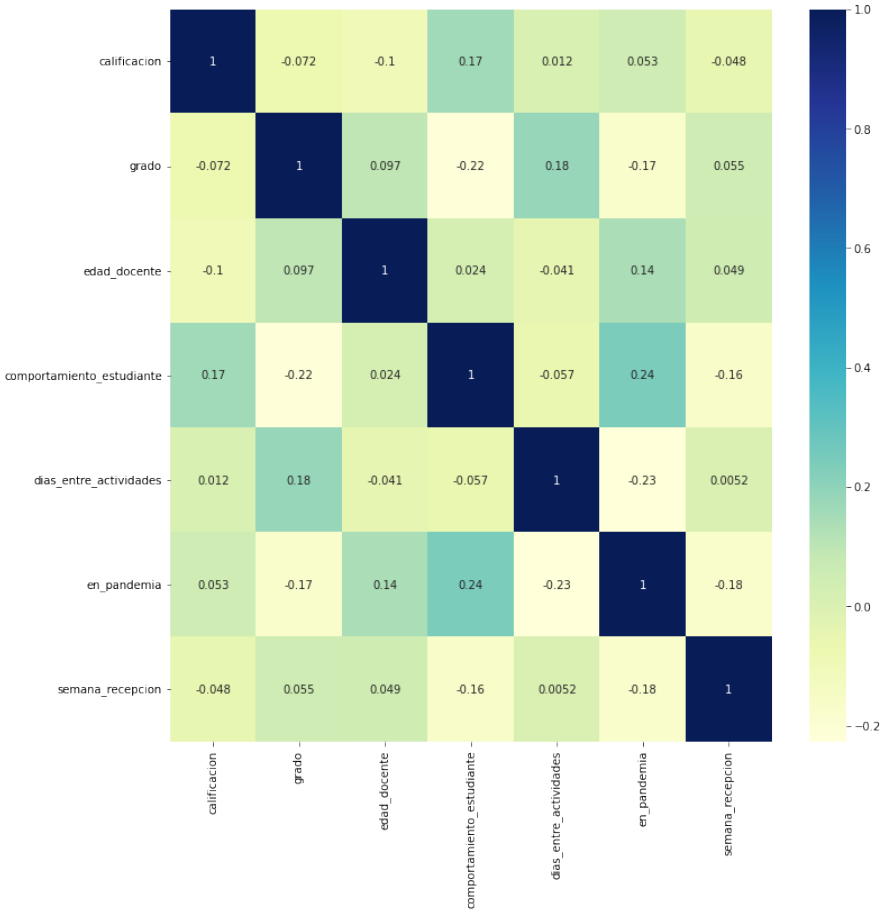
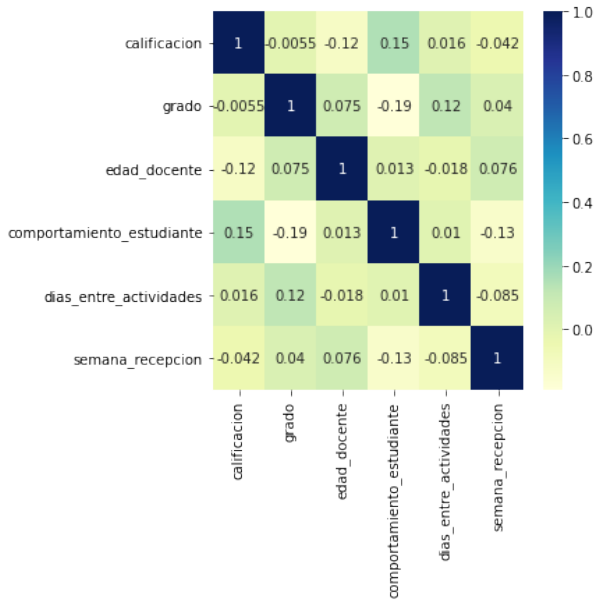


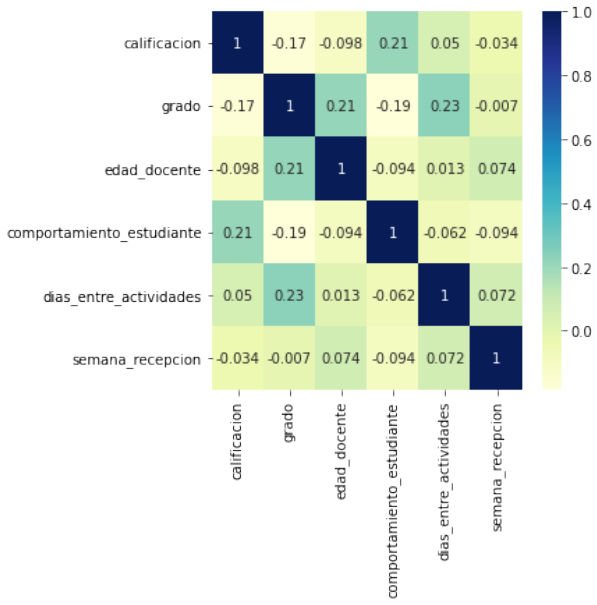
Figura 3.3 Gráfico de correlación sin atributos fuertemente correlacionados.

Podemos apreciar que ya no existen atributos con correlaciones fuertes entre ellos, sin embargo, seguimos sin tener atributos que muestren una correlación significativa con la variable a predecir.

Cuando aplicamos el factor de la pandemia para filtrar la correlación entre atributos podemos apreciar que existe un marcado efecto en los atributos del dataset. Este efecto se ve especialmente remarcado en los atributos dias_entre_actividades, edad_docente, grado, comportamiento_estudiante y calificacion. Esto permite intuir, que el atributo en_pandemia tiene un efecto directo (aunque no muy significativo) en la variable a predecir, e indirecto a través del resto de atributos.



Correlación entre atributos sin pandemia



Correlación entre atributos con pandemia

Figura 3.4 Gráfico de correlación entre atributos con y sin pandemia.

3.1.9 Verificación de independencia entre variables categóricas

Con la finalidad de realizar un análisis de independencia mediante tablas de contingencia, se realizó una segmentación de las variables continuas en variables categóricas, los atributos creados para este efecto tienen el prefijo “escala_” en su nombre seguida del nombre de la variable continua. Las segmentaciones realizadas son las siguientes:

Tabla 3.7 Segmentación de variables continuas.

Variable	Rangos
calificación	DAR: Calificación mayor o igual a 9. AAR: Calificación mayor o igual a 7 y menor a 9. PAAR: Calificación mayor a 4 y menor a 7. NAAR: Calificación menor o igual a 4.
nivel	0: Inicial 1: Preparatoria 2: EGB Elemental 3: EGB Media 4: EGB Superior 5: Bachillerato

edad_estudiante y edad_docente	Las dos variables se categorizaron de acuerdo con los siguientes rangos: Primera Infancia: menor o igual a 5 años. Infancia: mayor a 5 años y menor o igual a 11 años. Adolescencia: mayor a 11 años y menor o igual a 18 años. Adulto joven: mayor a 18 años y menor o igual a 26 años. Adulto: mayor a 26 años y menor o igual a 40 años. Adulto maduro: mayor a 40 años y menor o igual a 59 años. Adulto mayor: mayor a 59 años.
comportamiento_estudiante y comportamiento_curso	Las dos variables se categorizaron de acuerdo con los siguientes rangos: Muy satisfactorio: mayor o igual al 80% de la nota. Satisfactorio: mayor o igual al 60% y menor al 80%. Poco satisfactorio: mayor o igual al 40% y menor al 60%. Mejorable: mayor o igual al 20% y menor al 40%. Insatisfactorio: menor al 20%.
dias_entre_actividades y dias_entre_actividades_mismo_tipo	Corto plazo: menor o igual a 10 días. Mediano plazo: mayor a 10 días y menor o igual a 20 días. Largo plazo: mayor a 20 días.

Luego se procedió a realizar una tabla de contingencia entre las variables categóricas y las segmentaciones realizadas para luego utilizar un test de independencia *chi-cuadrado*. El resultado de este proceso se muestra en la tabla 3.8, en donde constan seis dependencias detectadas y para cada una de ellas, además de las variables involucradas, se muestran la función de punto porcentual (*ppf*), el estadístico de prueba de *chi-cuadrado* (*stat*) y el valor de *p* (*p-value*).

Tabla 3.8 Resultados de la prueba de independencia entre variables categóricas.

Atributo 1	Atributo 2	ppf	stat	p
genero_docente	escala_edad_docente	3.841459	13.041583	0.000305
genero_docente	en_pandemia	3.841459	12.663052	0.000373
genero_estudiante	genero_docente	3.841459	12.358913	0.000439
genero_estudiante	escala_edad_docente	3.841459	5.173473	0.022934
en_pandemia	escala_edad_docente	3.841459	4.884090	0.027105
genero_estudiante	en_pandemia	3.841459	4.087305	0.043206

Esta información nos servirá para la selección de los atributos que se utilizarán para el entrenamiento del modelo de predicción.

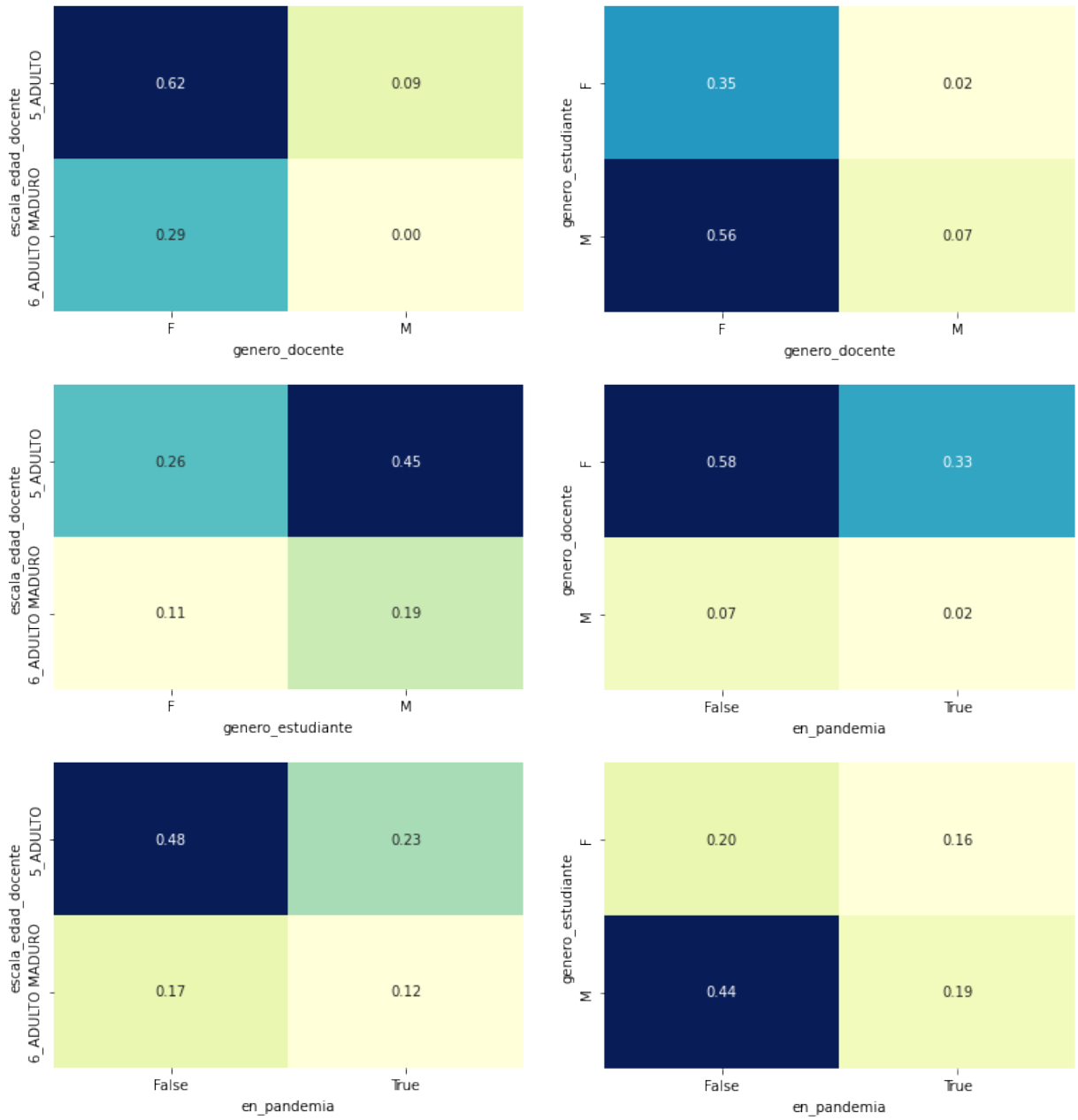


Figura 3.5 Gráfico de correlación de tablas de contingencia.

La figura 3.5 muestra gráficos de dependencia de los atributos que son dependientes entre si según la prueba de *chi-cuadrado*. Como se puede apreciar, la relación de dependencia se da entre variables de dos valores y la relación se explica por la características particulares de los datos de la unidad educativa. Por ejemplo, la relación entre: genero_docente y escala_edad_docente nos dice que la mayoría de las docentes, en esta unidad educativa en particular, son de genero femenino y adultas, sin embargo,

la correlación entre estas dos variables no es lo suficientemente significativa como para omitir una de ellas.

Por otra parte, las relaciones de dependencia con el atributo en_pandemia son ignoradas debido a que este atributo nos permite segmentar el comportamiento de los datos entre los dos escenarios: antes y después de la pandemia.

En lo que respecta a las columnas edad_estudiante y nivel se omitirán por su fuerte correlación con la columna grado, por lo que las relaciones evidenciadas en las columnas dependientes derivadas de ellas se solucionarán en el dataset final.

Por último, podemos observar que ninguna variable categórica presenta una correlación significativa con las categorías creadas para la calificación: escala_calificacion.

3.1.10 Selección de los atributos a utilizar

Con la información referente a las correlaciones entre atributos, procedimos a seleccionar aquellos que se utilizarán en el entrenamiento del modelo de predicción. El principal criterio de eliminación es la alta correlación entre atributos, luego la generalización del modelo. Por estos motivos se omitieron los siguientes atributos:

Tabla 3.9 Omisión de atributos del dataset para el entrenamiento del modelo.

Atributo omitido	Motivo
nivel	Por su alta correlación con grado.
mes_recepcion	Por su alta correlación con semana_recepcion.
edad_estudiante	Por su alta correlación con grado y nivel.
dias_entre_actividades_mismo_tipo	Por su alta correlación con dias_entre_actividades.
comportamiento_curso	Por su alta correlación con comportamiento_estudiante.

Adicionalmente se eliminaron la mayoría de los atributos generados para el análisis de correlaciones, tanto los que se usaron para clasificar las variables continuas, como los que permitieron determinar la correlación entre una calificación y sus calificaciones anteriores. No se eliminó el atributo escala_edad_docente debido a que se detectó una correlación leve entre la edad del docente y la calificación asignada al estudiante.

Por último, para definir los atributos que se utilizarán en el entrenamiento del modelo, se omitieron los atributos id_estudiante y id_docente, porque identifican individualmente al estudiante y al docente respectivamente, y pueden limitar la generalización del modelo.

Como resultado de esta exploración tenemos la siguiente lista de atributos seleccionados para el entrenamiento del modelo:

Tabla 3.10 Atributos seleccionados para el entrenamiento del modelo.

Atributo seleccionado
calificacion
area
grado
tipo_actividad
fecha_recepcion
genero_estudiante
genero_docente
escala_edad_docente
comportamiento_estudiante
dias_entre_actividades
en_pandemia

El dataset con los atributos seleccionados se exportó a un archivo plano tipo *CSV* para luego utilizar esta información en el entrenamiento del modelo de predicción.

3.2 Prototipos de algoritmos, modelos, y módulos del sistema

En el presente trabajo se plantea predecir el rendimiento académico de los estudiantes mediante la predicción de sus calificaciones, para ello, en el numeral anterior hemos aplicado técnicas estadísticas para extraer los atributos y valores que nos permitan realizar estas predicciones. Con estos atributos hemos definido un modelo de inteligencia artificial que se basa en los conceptos de redes neuronales y series temporales con la finalidad de realizar las predicciones.

3.2.1 Análisis de series temporales

En este apartado revisaremos algunos ejemplos de datos para explorar la factibilidad de aplicar modelos tradicionales de series temporales en el proceso de predicción de

calificaciones. Para ello graficaremos la evolución de las calificaciones para tres estudiantes diferentes en una misma asignatura, luego la evolución de calificaciones en tres asignaturas diferentes para un mismo estudiante, en todos los casos visualizaremos también la auto correlación de los datos.

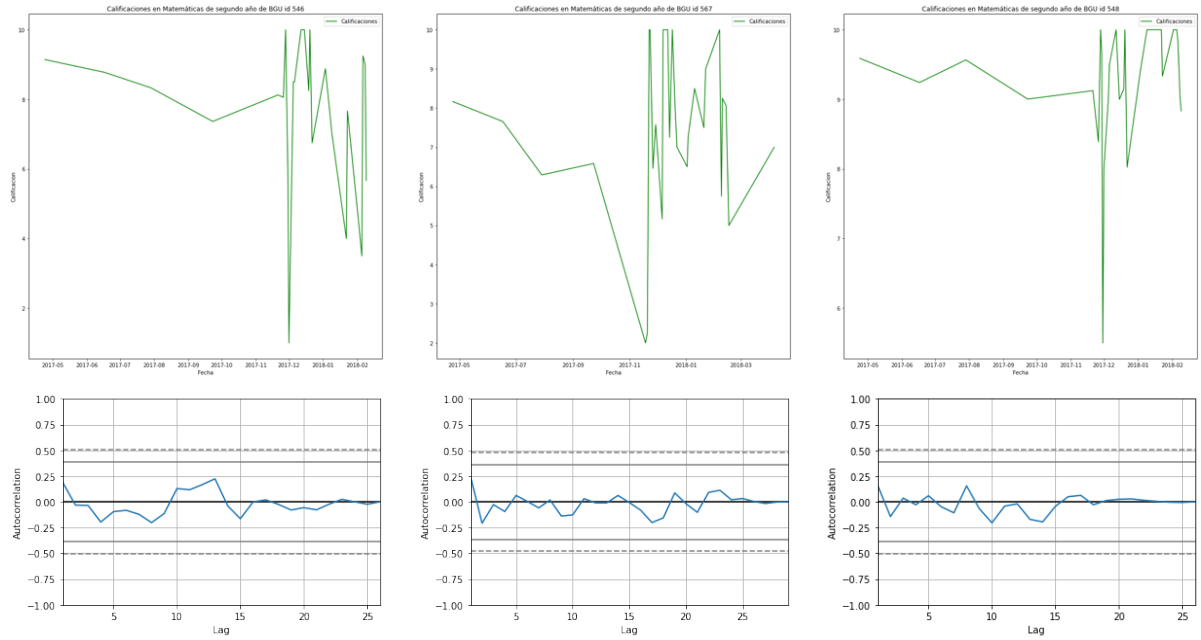


Figura 3.6 Evolución de las calificaciones en la asignatura de Matemáticas en segundo año de BGU para tres estudiantes diferentes.

La figura 3.6 muestra la evolución de las calificaciones en una misma asignatura para diferentes estudiantes en el mismo curso. Podemos notar que no se evidencia una estacionalidad, tendencia o ciclo; atributos característicos de una serie temporal. Tampoco existe una correlación entre los datos, tal como lo evidencia el gráfico de auto correlación ubicado en la parte inferior de cada serie de datos.

Por otra parte, en la figura 3.7 muestra la evolución de las calificaciones en diferentes asignaturas para un mismo estudiante. Al igual que en la figura 3.6, podemos notar que tampoco se evidencian las características de una serie temporal, también podemos visualizar que tampoco existe correlación entre los datos.

La falta de las características principales de las series temporales dificulta la aplicación de técnicas tradicionales de predicción aplicables a ellas. Esto nos orienta a explorar otras alternativas que nos permitan cumplir con nuestros objetivos.

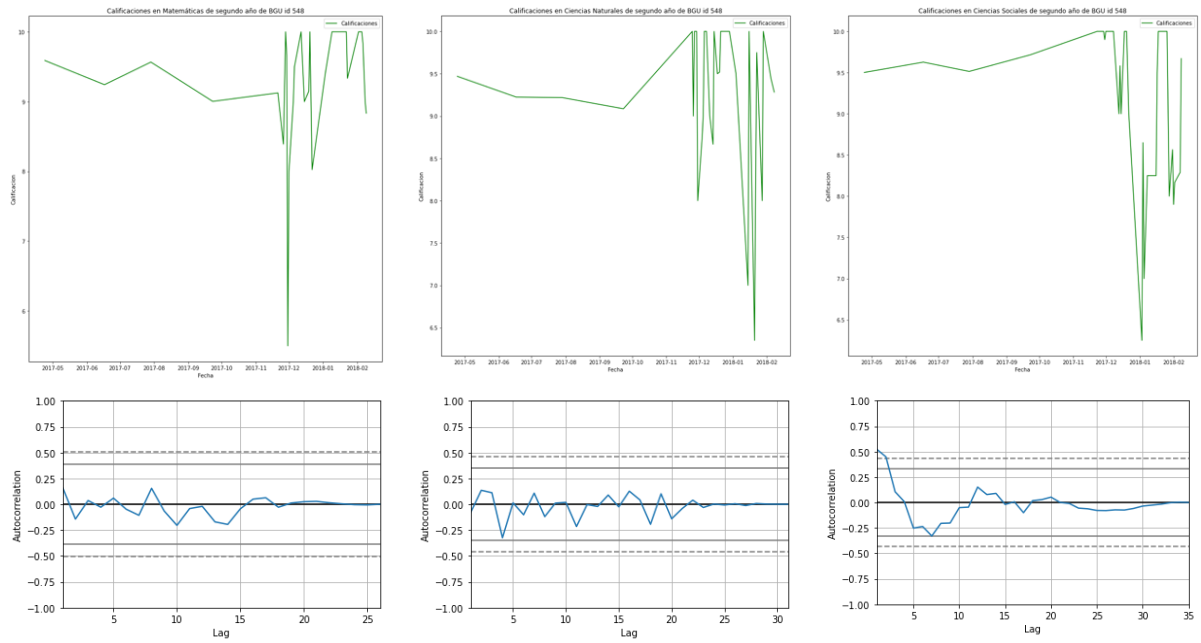


Figura 3.7 Evolución de las calificaciones en tres asignaturas diferentes para el mismo estudiante.

Al analizar las correlaciones del resto de variables con la variable a predecir pudimos apreciar que existía una correlación significativa con las calificaciones anteriores, se analizó la correlación con la calificación inmediatamente anterior y con el promedio, determinando que la correlación más fuerte de las dos es con la variable que contiene el promedio de las calificaciones anteriores; esto último significa que las calificación actual se explica más con el comportamiento en conjunto de datos anteriores antes que con la calificación inmediatamente anterior, lo cual hace necesario que en el modelo se pueda considerar el comportamiento de todas las calificaciones anteriores en su conjunto para predecir la calificación actual.

Por otra parte, debemos considerar que otros atributos también aportan con cierta información (aunque no tan significativamente como las calificaciones anteriores) en el comportamiento de la variable calificacion, como por ejemplo la variable comportamiento_estudiante con un factor de correlación de 0.16 o la variable edad_docente con un factor de -0.1. La tabla 3.11 muestra la correlación entre la variable a predecir, calificacion, y el resto de las variables numéricas.

Tabla 3.11 Factores de correlación entre la variable a predecir y el resto de las variables numéricas.

Variable	Factor de correlación con la variable a predecir
edad_docente	-0.1
comportamiento_estudiante	0.17
dias_entre_actividades	0.012
semana_recepcion	-0.048
calificacion_promedio	0.46
calificacion_previa	0.39

En lo que respecta a las variables categóricas, para analizar la dependencia entre ellas, se realizó una categorización de la variable a predecir, para luego verificar la independencia de variables mediante un test *chi-cuadrado*. El resultado de este proceso se muestra en la tabla 3.12.

Tabla 3.12 Factores de correlación entre la variable a predecir y las variables categóricas.

Variable	ppf	stat	p
en_pandemia	7.81473	0.01136	0.99968
genero_estudiante	7.81473	0.00635	0.99987
genero_docente	7.81473	0.00165	0.99998
grado	24.99579	0.05646	1.00000
tipo_actividad	40.11327	0.02258	1.00000
area	32.67057	0.03373	1.00000

Podemos apreciar que en ninguno de los casos el estadístico de prueba (*stat*) supera el valor de la función de punto porcentual (*ppf*), por lo que no podemos concluir que existe una dependencia evidente entre las variables categóricas y la variable a predecir. Sin embargo, tal como se muestra en la figura 3.8, podemos apreciar que existen ciertos sesgos en las calificaciones dependiendo del valor de estas variables categóricas, como, por ejemplo: en el caso de la variable en_pandemia se muestra que los estudiantes obtuvieron mejores calificaciones (*DAR*) cuando esta variable es verdadera, o en el caso de la variable genero_docente, en donde vemos que las mejores calificaciones son otorgadas por las docentes de género femenino.

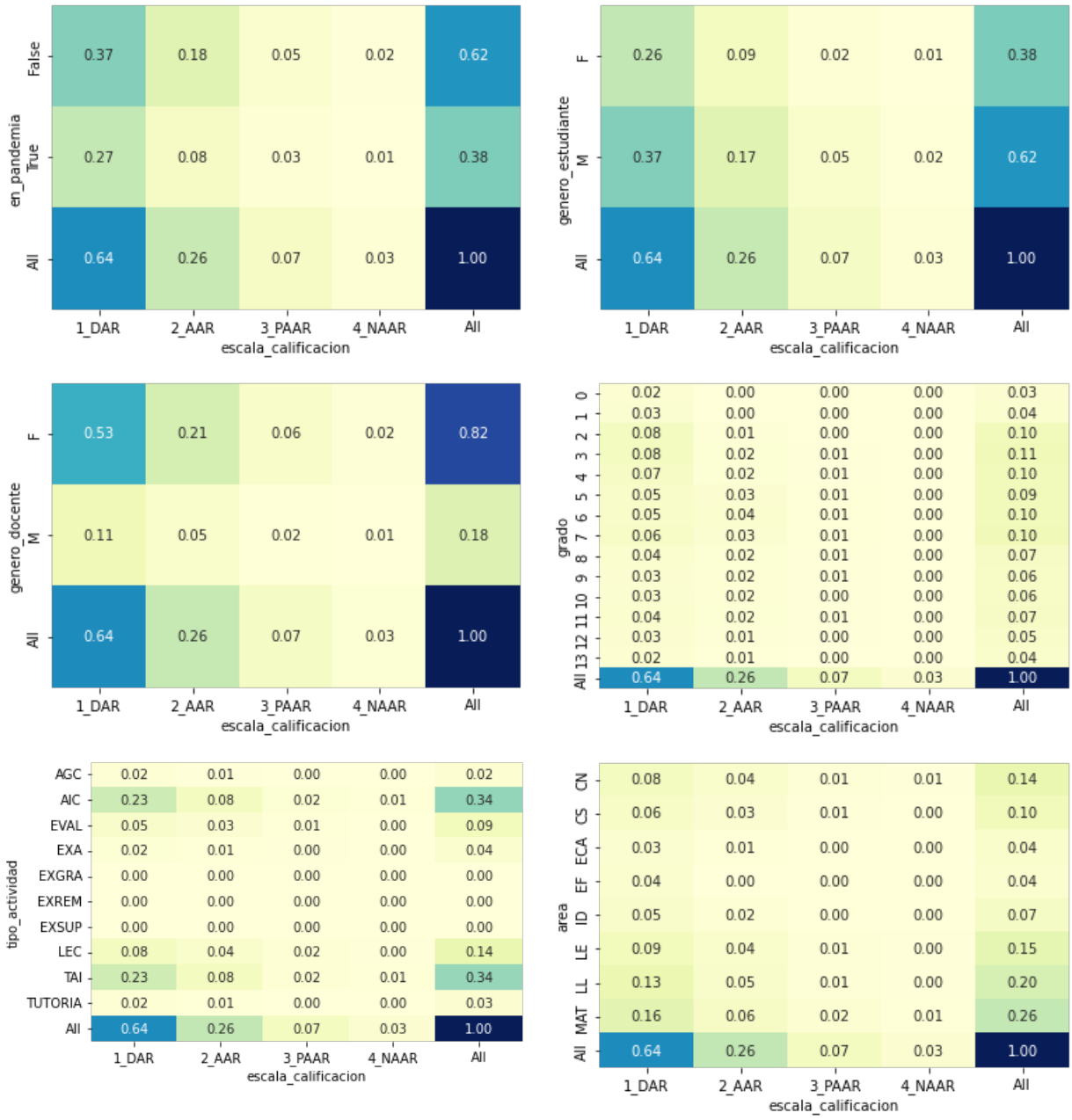


Figura 3.8 Tablas de correlación entre la variable a predecir y las variables categóricas.

Lo anterior nos permite concluir que las variables seleccionadas aportan con información para la obtención del valor de la variable a predecir, por lo que un modelo que considere todas estas variables permitirá obtener una mayor exactitud en las predicciones realizadas.

3.2.2 Selección del modelo de predicción

En el análisis de series temporales realizado en el apartado anterior se concluye que la información de las calificaciones, contenidas en el dataset, no muestra significativamente los atributos necesarios para realizar predicciones utilizando las técnicas tradicionales aplicables a este tipo de información, incluso, los gráficos de correlación evidencian que no existe una correlación evidente entre los datos. Esto hace necesaria la exploración de otras alternativas para realizar las predicciones requeridas en este trabajo.

En el capítulo 2, analizamos varios estudios que utilizaron diversas metodologías para realizar predicciones del rendimiento académico estudiantil, entre ellas sobresale el trabajo de Amirah et al. (2015), en donde mediante el uso de una red neuronal obtuvieron una exactitud del 98% o el de Sivasakthi (2017), quien logró una exactitud del 93% con el uso de una red neuronal MLP. El alto nivel de exactitud que puede lograr una red neuronal, según los dos trabajos antes mencionados, sumado a la facilidad que tienen las redes neuronales para analizar, detectar y memorizar patrones entre múltiples variables de entrada, hacen de éstas las mejores candidatas para la construcción del modelo de predicción en este trabajo. Sin embargo debemos considerar que, cuando se trabaja con series de tiempo, en donde la cantidad de datos históricos puede ser alto, la mayoría de redes neuronales tienen problemas para relacionar valores que se encuentran muy distantes entre sí en la línea del tiempo, a este problema se le denomina "*desvanecimiento de gradiente*", y nos referimos a él en el capítulo 2; también en ese mismo capítulo mencionamos que la arquitectura de red neuronal denominada "*Transformers*" soluciona el problema del "*desvanecimiento de gradiente*", y es por esta razón que para el presente trabajo utilizaremos este tipo de arquitectura de redes neuronales en la construcción del modelo de predicción.

3.2.3 Modelo de predicción

Para la implementación del modelo de predicción utilizamos la librería "keras" de Python, utilizando un componente denominado "keras_transformer". La definición de los hiper-parámetros es la siguiente:

Tabla 3.13 Parámetros para la definición del modelo de predicción.

Parámetro	Descripción	Valor
token_num	Número máximo de tokens de entrada. Cada token es un posible valor de la secuencia de datos.	Se obtiene tomando el tamaño máximo entre los diccionarios de los "Datos de entrada" y de los "Datos objetivo".
embed_dim	Tamaño de las capas neuronales de codificación de datos.	32 neuronas
encoder_num	Número de codificadores	2 bloques
decoder_num	Número de decodificadores	2 bloques
head_num	Cantidad de espacios lineales utilizados para el cálculo de los vectores Q, K y V.	4 espacios lineales
hidden_dim	Tamaño de las redes neuronales "feed forward".	128 neuronas
dropout_rate	Tasa de eliminación para evitar el sobreajuste del modelo.	0.20 equivalente al 20% de los datos.
use_same_embed	Indica si se utilizará una sola capa de codificación de datos tanto para los "Datos de entrada" y para los "Datos objetivo". Si este valor es falso se utilizarán capas distintas.	Falso

```
# Crear la red transformer

model = get_model(
    token_num = max(len(source_token_dict), len(target_token_dict)),
    embed_dim = 32,
    encoder_num = 2,
    decoder_num = 2,
    head_num = 4,
    hidden_dim = 128,
    dropout_rate = 0.20,
    use_same_embed = False
)

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

```

Figura 3.9 Definición del modelo de predicción.

La figura 3.9 muestra la definición del modelo utilizado para la predicción con los parámetros descritos en la tabla 3.13. Se puede apreciar que para el entrenamiento se utiliza un optimizador "*adam*" y para el cálculo de la pérdida utiliza la función "*sparse_categorical_crossentropy*" debido a que existen más de dos valores (clases) a predecir.

3.2.3.1 Entrenamiento

El proceso de entrenamiento requiere información en el formato esperado por el modelo, para ello se transformó el dataset resultante del proceso de "*Exploración y validación de datos y fuentes*" a un conjunto de tokens con el formato adecuado para el entrenamiento. El dataset se compone de 2,370,727 registros y luego de la transformación se obtuvieron 1,935,226 tokens para el entrenamiento.

"*Token embedding*", transformación de los tokens de entrenamiento

El proceso de transformación genera un token por cada estudiante, grado y área de estudio, identificando también las calificaciones obtenidas antes y durante la pandemia del COVID-19. En la figura 3.10 se ilustra este proceso.

calificacion	area	grado	tipo_actividad	fecha_recepcion	genero_estudiante	genero_docente	comportamiento_estudiante	dias_entre_actividades	en_pandemia
1000	11001	11104	11201	2017-04-24	11302	11302	11600	11710	12102
1000	11001	11104	11201	2017-06-17	11302	11302	11600	11730	12102
1000	11001	11104	11201	2017-11-30	11302	11302	11600	11710	12102
1000	11001	11104	11201	2017-11-30	11302	11302	11600	11710	12102
1000	11001	11104	11201	2017-12-22	11302	11302	11600	11710	12102
...	...	...	...	...	...	...	...	...	...
1000	11008	11108	11205	2021-09-02	11302	11302	11600	11710	12101
800	11008	11108	11205	2021-09-14	11302	11302	11600	11720	12101
1000	11008	11108	11207	2021-09-07	11302	11302	11600	11710	12101
1000	11008	11108	11207	2021-09-09	11302	11302	11600	11710	12101
800	11008	11108	11207	2021-09-23	11302	11302	11600	11710	12101

```
[array([12102, 11104, 11001, 11302, 11302, 11405, 11202, 12201, 11600,
        11710,   900, 11202, 12208, 11600, 11730,  1000, 11202, 12214,
        11600, 11720,   700, 11202, 12214, 11600, 11720,   900, 11202,
        12214, 11600, 11720,  1000, 11202, 12214, 11600, 11720,  1000,
        11202, 12222, 11600, 11730,  1000, 11202, 12222, 11600, 11730,
         1000, 11202, 12238, 11600, 11730,   800, 11202, 12238, 11600,
        11730,   800, 11203, 12214, 11600, 11720,  1000]),
 array([12102, 11104, 11001, 11302, 11302, 11405, 11202, 12208, 11600,
        11730,  1000, 11202, 12214, 11600, 11720,   700, 11202, 12214,
        11600, 11720,   900, 11202, 12214, 11600, 11720,  1000, 11202,
        12214, 11600, 11720,  1000, 11202, 12222, 11600, 11730,  1000,
        11202, 12222, 11600, 11730,  1000, 11202, 12238, 11600, 11730,
          800, 11202, 12238, 11600, 11730,   800, 11203, 12214, 11600,
        11720,  1000, 11203, 12232, 11600, 11710,   900]),
 array([12102, 11104, 11001, 11302, 11302, 11405, 11202, 12214, 11600,
        11720,   700, 11202, 12214, 11600, 11720,   900, 11202, 12214,
        11600, 11720,  1000, 11202, 12214, 11600, 11720,  1000, 11202,
        12222, 11600, 11730,  1000, 11202, 12222, 11600, 11730,  1000,
        11202, 12238, 11600, 11730,   800, 11202, 12238, 11600, 11730,
          800, 11203, 12214, 11600, 11720,  1000, 11203, 12232, 11600,
        11710,   900, 11203, 12232, 11600, 11710,  1000]),
 array([12102, 11104, 11001, 11302, 11302, 11405, 11202, 12214, 11600,
        11720,   900, 11202, 12214, 11600, 11720,  1000, 11202, 12214,
        11600, 11720,  1000, 11202, 12222, 11600, 11730,  1000, 11202,
        12222, 11600, 11730,  1000, 11202, 12238, 11600, 11730,   800,
        11202, 12238, 11600, 11730,   800, 11203, 12214, 11600, 11720,
         1000, 11203, 12232, 11600, 11710,   900, 11203, 12232, 11600,
        11710,  1000, 11203, 12232, 11600, 11710,  1000]),
```

Figura 3.10 Transformación del dataset con los atributos seleccionados a un conjunto de tokens para el entrenamiento del modelo.

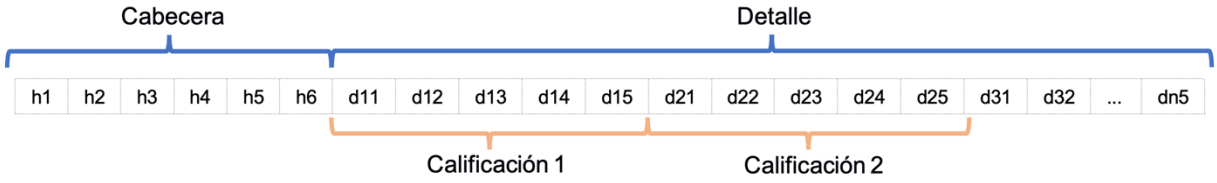


Figura 3.11 Estructura del token.

Cabecera

Cada token se compone de una cabecera y un detalle, esta estructura se muestra en la figura 3.11. La cabecera la representan los 6 primeros elementos del token y contiene los siguientes valores:

Tabla 3.14 Estructura de la cabecera del token.

Elemento	Contenido	Valores
h1	Define si la calificación fue obtenida en un periodo de pandemia o no.	12101: No 12102: Si
h2	Grado de estudio del estudiante cuando obtuvo las calificaciones.	111NN, donde NN es el grado de estudio según la tabla 3.14.
h3	Área de estudio en la que obtuvieron las calificaciones.	110NN, donde NN es el código asignado al área de estudio según la tabla 3.15.
h4	Género del estudiante.	11301: Masculino 11302: Femenino
h5	Género del docente.	11301: Masculino 11302: Femenino
h6	Escala de edad del docente.	114NN, donde NN es el código asignado al rango de edad del docente según la tabla 3.16.

Tabla 3.15 Códigos numéricos para el grado de estudio.

Código	Nivel de estudio
00	Inicial
01 – 10	Primero año hasta décimo año de Educación General Básica (EGB).
11	Primer año de Bachillerato General Unificado (BGU).
12	Segundo año de BGU.
13	Tercer año de BGU.

Tabla 3.16 Códigos numéricos para el área de estudio.

Código	Área de estudio
01	Educación cultural y artística
02	Educación física
03	Ciencias Naturales
04	Ciencias Sociales
05	Lengua y Literatura
06	Matemáticas
07	Lengua Extranjera
08	Interdisciplinar

Tabla 3.17 Códigos numéricos para los rangos de edad.

Código	Área de estudio
00	Sin edad definida
01	Primera infancia, menor o igual a 5 años.
02	Infancia, mayor a 5 y menor o igual a 11 años.
03	Adolescencia, mayor a 11 y menor o igual a 18 años.
04	Adulto joven, mayor a 18 y menor o igual a 26 años.
05	Adulto, mayor a 26 y menor o igual a 40 años.
06	Adulto maduro, mayor a 40 y menor o igual a 59 años.
07	Adulto mayor, mayor a 59 años.

Detalle

El detalle del token lo componen las calificaciones obtenidas por el estudiante y se divide en grupos de 5 elementos, cada grupo se estructura de la siguiente manera y representa a una calificación obtenida por el estudiante:

Tabla 3.18 Estructura de cada grupo del detalle del token.

Elemento	Contenido	Valores
dn1	Define el tipo de actividad académica.	112NN, donde NN es el tipo de actividad según la tabla 3.18.
dn2	Semana del periodo académico.	122NN, donde NN es el número de semana, el rango va desde 01 hasta 99.
dn3	Calificación del comportamiento del estudiante.	11500 mas el valor en base 100 de la calificación obtenida por el estudiante en su

		comportamiento, así, si un estudiante obtiene 100 puntos en comportamiento (nota máxima) el valor de este elemento es 11600.
dn4	Días plazo para realizar la actividad	117NN, donde NN es la cantidad de días de plazo, el rango va desde 01 hasta 99
dn5	Calificación obtenida.	Calificación obtenida por el estudiante multiplicada por 100. Los valores van desde 1 hasta 1000, donde 1000 equivale a una nota de 10.00.

Tabla 3.19 Códigos asignados a los tipos de actividad.

Código	Tipo de actividad
01	TAI: Trabajos Académicos Independientes (Tareas).
02	AIC: Actividades individuales en clase.
03	AGC: Actividades grupales en clase.
04	LEC: Lecciones.
05	EVAL: Evaluaciones parciales.
06	TUTORIA: Tutorías.
07	EXA: Exámenes quimestrales.
08	EXGRA: Exámenes de gracia.
09	EXSUP: Exámenes supletorios.
10	EXREM: Exámenes remediales.

Generación de los tokens para el entrenamiento

Una vez obtenido el arreglo con los tokens se procedió a generar dos arreglos denominados “source_tokens” y “target_tokens” que serán utilizados en la etapa de “*Codificación*” y “*Decodificación*” del modelo respectivamente.

El contenido del arreglo source_tokens es el conjunto completo de tokens sin el último elemento de cada token. Y el contenido del arreglo target_tokens es el último elemento de cada token.

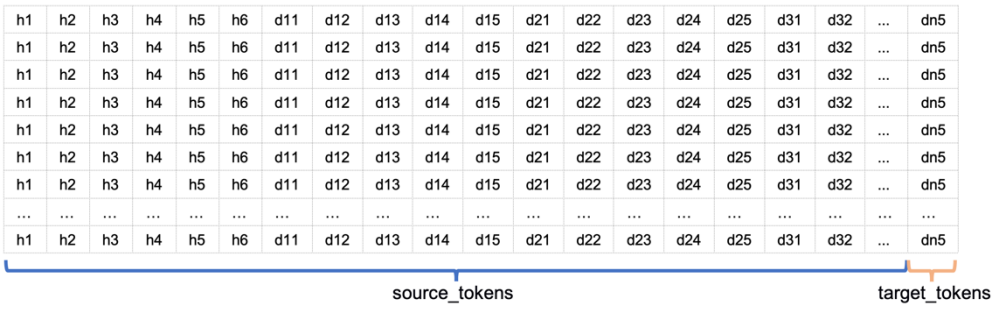


Figura 3.12 Segmentación del arreglo de tokens.

Diccionarios de datos

Previo al entrenamiento del modelo se realizó la adecuación de cada token agregando elementos que definan el inicio y fin de un token, además de elementos que representen la ausencia de información. Esto debido a que el proceso de entrenamiento requiere tokens de igual tamaño por lo que es necesario agregar elementos sin información en aquellos tokens con un tamaño inferior al requerido. Los valores de cada uno de estos elementos se muestran en la tabla 3.20.

Tabla 3.20 Códigos asignados a elementos de delimitación de tokens.

Código	Tipo de actividad
-1	PAD_TOKEN: Elemento que representa la ausencia de información.
-2	START_TOKEN: Representa el inicio de un token.
-3	END_TOKEN: Representa el fin de un token.

Una vez agregados los elementos de delimitación y normalizado el tamaño de los tokens se generaron los arreglos requeridos para el proceso de entrenamiento y validación. Los arreglos obtenidos se listan en la tabla 3.21.

Tabla 3.21 Arreglos generados para el entrenamiento.

Arreglo	Contenido
source_token_dict	Diccionario con los elementos existentes en el arreglo source_token.
target_token_dict	Diccionario con los elementos existentes en el arreglo target_token.
target_token_dict_inv	Es el mismo diccionario target_token_dict, pero invertido, con la finalidad de utilizarlo para obtener el resultado de la predicción.
encoder_input	Arreglo con los tokens que se utilizarán como entrada en el entrenamiento del modelo en la etapa de “Codificación”.

decoder_input	Arreglo con los tokens que se utilizarán como entrada en el entrenamiento del modelo en la etapa de “Decodificación”.
output_decoded	Arreglo con los tokens que se utilizarán para validar la exactitud del modelo.

Entrenamiento del modelo

El entrenamiento del modelo se realizó utilizando la plataforma de *Google Collaboratory*, utilizando un script mediante el cual se almacena paulatinamente los modelos con menor pérdida a medida que avanza el entrenamiento.

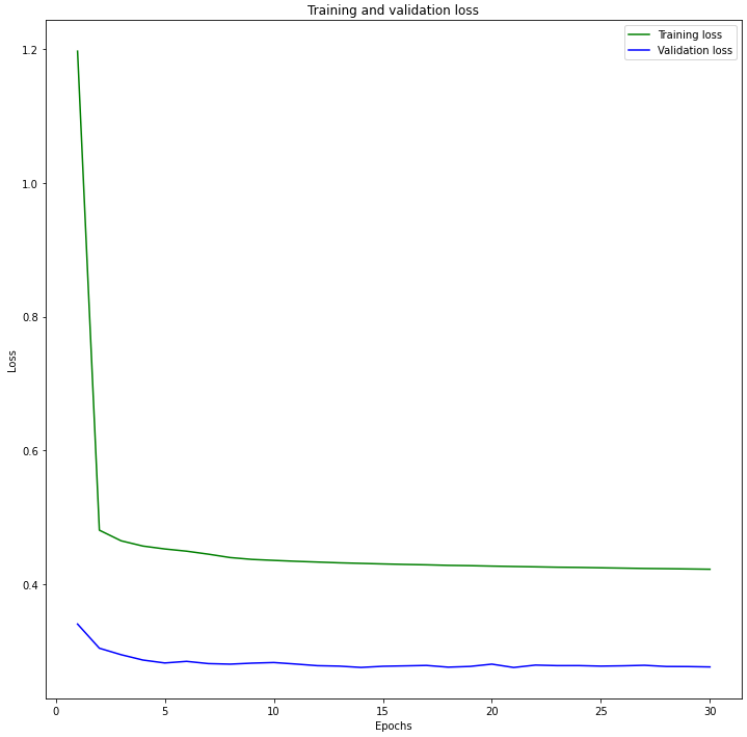


Figura 3.13 Evolución de la función de pérdida y validación en el proceso de entrenamiento.

Para el procesar el entrenamiento se ingresó al modelo los tokens transformados mediante el proceso de “*Token embedding*”. El resto de los bloques de la estructura del modelo “*Transformer*” que se muestran en la figura 2.1 se ejecutan automáticamente en el proceso de entrenamiento y están definidos como *hiper-parámetros* al construir el modelo (ver “*Definición del modelo*”, antes en esta sección).

El entrenamiento se realizó utilizando lotes 5000 tokens, utilizando el 20% de los datos para validación y el 80% para entrenamiento.

El modelo entrenado se almacenó en formato HDF5 de Keras, para luego ser utilizado en las pruebas de predicción y su uso en producción.

3.2.3.2 Predicción de calificaciones

Para realizar la predicción de calificaciones se requiere pasar al modelo un token con una estructura similar a la que se utilizó para el entrenamiento (ver "*Token embedding, transformación de los tokens de entrenamiento*" antes en esta sección), esto se logra realizando el proceso de "*Token embedding*" sobre la serie de datos que se pasará al modelo entrenado.

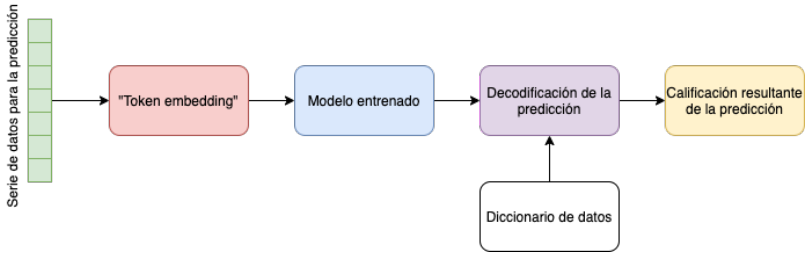


Figura 3.14 Proceso de predicción de calificaciones.

El modelo entrenado toma la serie de datos convertida y con ella realiza una predicción. La predicción se decodifica utilizando el diccionario de datos creados para el entrenamiento del modelo (ver "*Diccionarios de datos*" antes en esta sección).

3.2.4 Arquitectura de la solución

Para la definición, entrenamiento y uso del modelo de predicción hemos definido una arquitectura que se muestra en la figura 3.15. En ella podemos apreciar dos etapas: la definición y entrenamiento del modelo y la utilización del modelo.

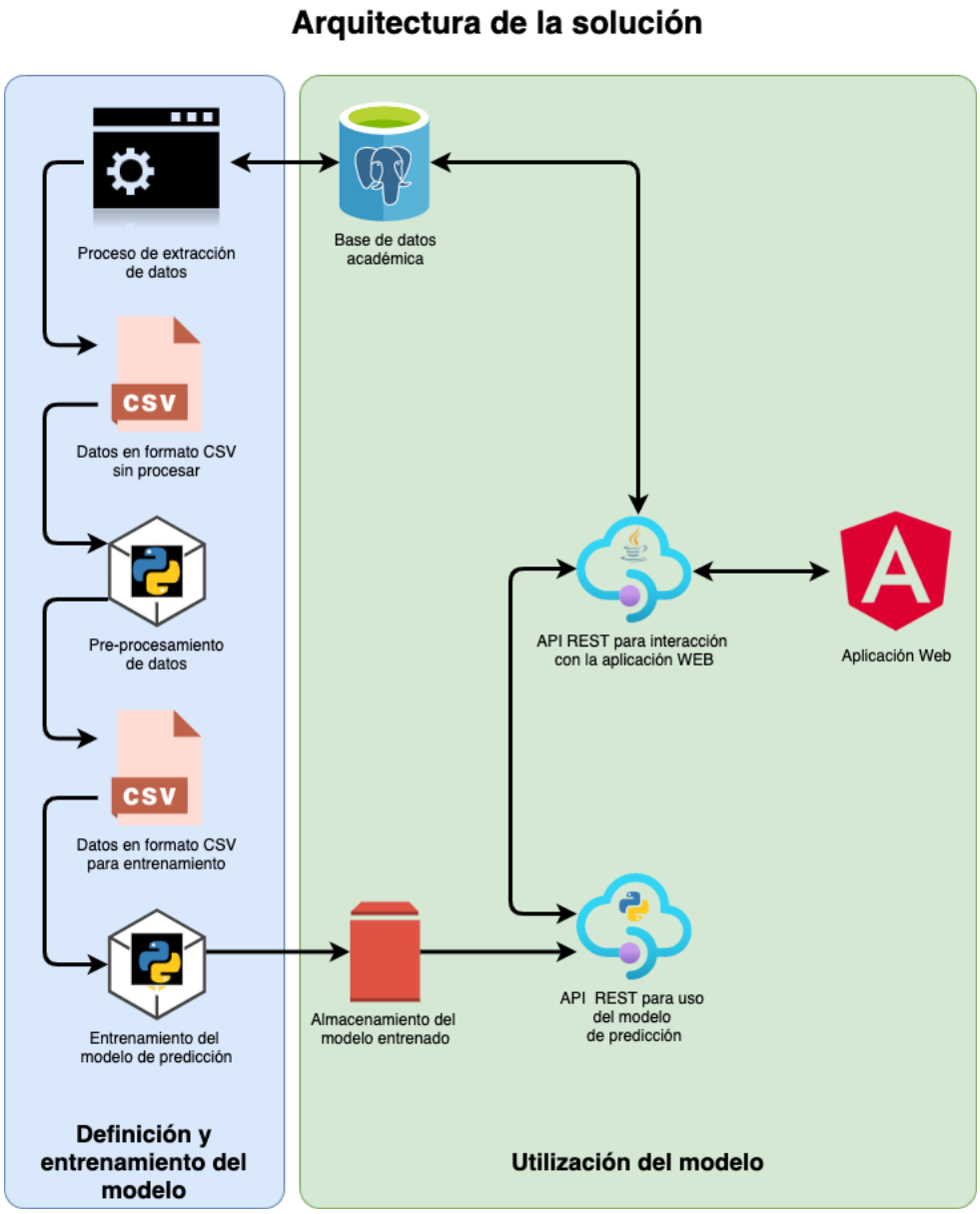


Figura 3.15 Arquitectura de la solución, vista por etapas.

3.2.4.1 Definición y entrenamiento del modelo

En la definición y entrenamiento del modelo interactúan los siguientes procesos:

Tabla 3.22 Procesos de la etapa de definición y entrenamiento del modelo.

Proceso	Descripción
Proceso de extracción de datos	El proceso de extracción de datos se realiza mediante una consulta a la base de datos académica creada utilizando SQL. El resultado de la

	consulta se envía a un archivo en formato CSV. La estructura del archivo y su contenido se detalla en numeral "*1.7 Dataset*".
Pre-procesamiento de datos	El pre-procesamiento se realizó utilizando scripts de Python. En este paso se realizó la exploración y validación de datos, se extrajeron los atributos más relevantes y se generaron los tokens y diccionarios necesarios para el entrenamiento del modelo de predicción.
Entrenamiento del modelo de predicción	Este paso también se realizó utilizando scripts de Python, aquí construimos el modelo de predicción, lo entrenamos y lo almacenamos para ser utilizado en la generación de predicciones. En la sección "Modelo de Predicción" se encuentran más detalles sobre su definición y entrenamiento.

3.2.4.2 Utilización del modelo

En esta etapa se hace uso del modelo entrenado para realizar predicciones mediante una interfaz web. Para ello se desarrollaron dos APIs que interactúan entre si y con la aplicación web. La figura 3.16 muestra los componentes de esta etapa y su interacción.

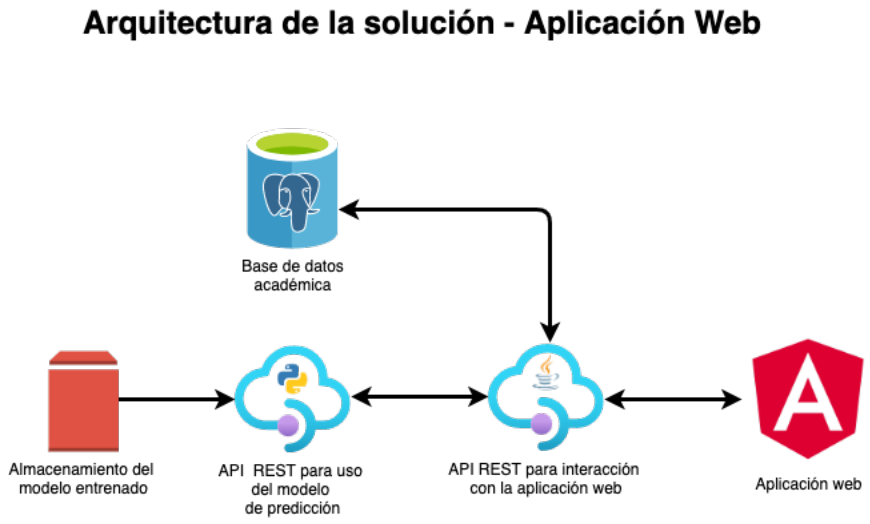


Figura 3.16 Arquitectura de la solución, componentes que interactúan con la aplicación web.

API REST para uso del modelo de predicción

Esta API se desarrolló con Django Framework, una librería de Python, y contiene un solo servicio web asociado al método POST del protocolo HTTP. Este servicio web acepta un objeto JSON como parámetro con el contenido que se muestra en la tabla 3.23.

Tabla 3.23 Estructura del parámetro del servicio web POST /predictions/.

Atributo	Descripción
pandemic	Indica si se debe considerar el efecto de la pandemia o no, los valores son: S : Si N : No
level	Nivel de estudio, los valores son: 0 : inicial 1 : primer año 2 : segundo año 3 : tercer año 4 : cuarto año 5 : quinto año 6 : sexto año 7 : séptimo año 8 : octavo año 9 : noveno año 10 : decimo año 11 : primer año de bachillerato 12 : segundo año de bachillerato 13 : tercer año de bachillerato
area	Área de estudio, los valores son: ECA : Educación cultural y artística EF : Educación física CN : Ciencias Naturales ES : Estudios Sociales LL : Lengua y Literatura MAT : Matemáticas LE : Lengua Extranjera ID : Interdisciplinar
studentGen	Género del estudiante, los valores son: F : Femenino M : Masculino
teacherGen	Género del docente, los valores son: F : Femenino M : Masculino
teacherAgeScale	Escala de edad a la que pertenece el docente, los valores son: 0 : No determinado 1 : Primera infancia (igual o menor a 5 años) 2 : Infancia (entre 6 y 11 años) 3 : Adolescencia (entre 12 y 18 años) 4 : Adulto joven (entre 19 y 26 años) 5 : Adulto (entre 27 y 40 años) 6 : Adulto maduro (entre 41 y 59 años) 7 : Adulto mayor (igual o mayor a 60 años)

previousScores	Arreglo con las calificaciones obtenidas anteriormente en el área de estudio. La tabla 3.23 muestra la estructura de los elementos de este arreglo.

Tabla 3.24 Estructura de los "scores" (calificaciones) enviados al servicio web POST /predictions/.

Atributo	Descripción
activityType	Tipo de actividad, los valores son: TAI : Trabajos Académicos Independientes (Tareas) AIC : Actividades Individuales en Clases AGC : Actividades Grupales en Clase LEC : Lecciones EVAL : Evaluaciones TUTORIA : Tutorías EXA : Examen EXGRA : Examen de gracia EXSUP : Examen supletorio EXREM : Examen remedial
academicWeek	Número de semana académica, los valores comienzan en 1 para la primera semana hasta la semana 46. Este límite debido a que es el número máximo de semanas existentes en el dataset que se usó para el entrenamiento.
studentBehaviour	Calificación del comportamiento del estudiante, el rango de valores es de 0 a 100.
daysBetweenActivities	Rango de días desde la última calificación hasta la actual. Los valores son: 10: De 0 a 10 días 20: de 11 a 20 días 30: de 21 días en adelante.
score	Calificación, el rango de valores va de 0 a 10.

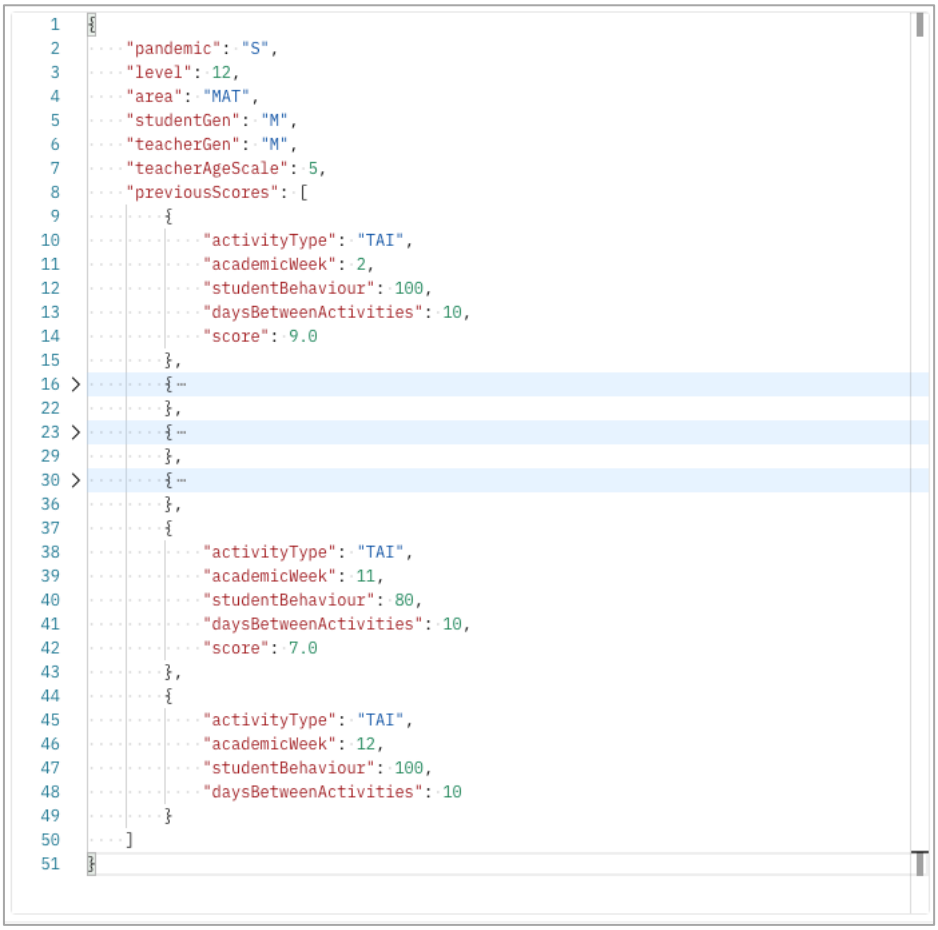


Figura 3.17 Ejemplo de la estructura JSON enviada como parámetro al servicio web POST /predictions/.

La figura 3.17 muestra un ejemplo de la estructura que espera el servicio web para ejecutar las predicciones, nótese que desde la línea 2 a la 7 contiene los atributos mencionados en la tabla 3.23, mientras que desde la línea 10 a la 14 muestra los atributos detallados en la tabla 3.24, por ende, el atributo “previousScores” es un arreglo que contiene elementos con similar estructura, los cuales se repiten desde la línea 9 a la 43. El elemento que se detalla en las líneas 45 hasta la 48 no contiene el atributo “score”, justamente porque ese último elemento detalla los atributos adicionales de la predicción que se va a realizar. En otras palabras, el último elemento del arreglo “previousScores” siempre detallará los atributos individuales de la calificación a predecir.

El detalle de los procesos que se ejecutan al hacer un llamado al servicio POST /predictions/ se muestra en la figura 3.18, debemos resaltar que el objeto JSON recibido

pasa por un proceso de "*input embedding*" para transformar los parámetros recibidos a un token que pueda ser ingresado al modelo para obtener la predicción. El resultado de la ejecución del modelo pasa por un proceso de decodificación para obtener el resultado de la ejecución.

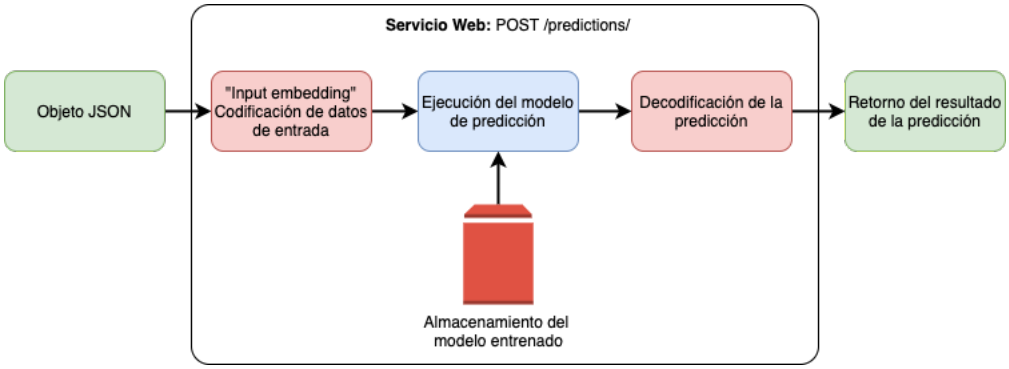


Figura 3.18 Procesos del servicio web para ejecución del modelo de predicción.

El resultado de la predicción es un objeto JSON con la estructura que se muestra en la tabla 3.25 y la figura 3.19 muestra un ejemplo da la respuesta del servicio web de predicción.

Tabla 3.25 Estructura del la respuesta del servicio web de predicción.

Atributo	Descripción
predictionScore	Predicción de la calificación, el rango de valores es de 0 a 10.
predictionClass	Clase asignada a la calificación, la equivalencia se muestra en la tabla 3.25.

```
{
    "predictionScore": 9.0,
    "predictionClass": "DAR"
}
```

Figura 3.19 Ejemplo de la respuesta del servicio web de predicción.

Tabla 3.26 Clases asignadas a la calificación, según la Ley Orgánica de Educación Intercultural.

Clase	Equivalencia
DAR	Domina los aprendizajes requeridos, calificación mayor o igual a 9 y menor o igual a 10.
AAR	Alcanza los aprendizajes requeridos, calificación menor a 9 y mayor o igual a 7.
EPAAR	Está próximo a alcanzar los aprendizajes requeridos, calificación menor a 7 y mayor a 4.
NAAR	No alcanza los aprendizajes requeridos, calificación menor o igual a 4.

API REST para interacción con la aplicación web

Para la operación de la aplicación web se desarrolló un API utilizando Java y Spring Framework, el API interactúa con la base de datos académica para recuperar los datos de los estudiantes sobre los que se realiza la consulta. Luego hace uso del servicio web de predicción para realizar las predicciones necesarias de acuerdo con el rango de predicción requerido. Los servicios web desarrollados en esta API se clasifican en dos clases: los servicios web para gestión de objetos y los servicios web para la generación de predicciones.

Servicios web para gestión de objetos

Estos servicios web se ajustan a la interfaz IRestCRUDController. Esta interfaz define algunos servicios de índole genérica que permiten la creación, recuperación, actualización y borrado de objetos persistentes (CRUD por sus siglas en inglés). En su implementación se debe indicar la clase de los objetos que se gestionará y el tipo de datos del ID de los objetos. La tabla 3.27 detalla los servicios web que define la interfaz antes mencionada.

Tabla 3.27 Servicios web definidos por la interfaz IRestCRUDController.

Servicio web	Descripción
GET /search	Realiza una búsqueda de objetos en base al criterio que acepta como parámetro.
GET /get-by-id	Busca un objeto en base a su ID.
GET /	Retorna todos los objetos existentes en el repositorio de la clase que se está gestionando. Este método no está implementado para todos los objetos, por motivos de seguridad.
POST /	Almacena el objeto enviado como parámetro.
POST /save-list	Almacena una lista de objetos enviada como parámetro.

DELETE /delete-by-id	Elimina un objeto en base a su ID. Se realiza una eliminación lógica en la base de datos. Este método no se implementa en el presente trabajo por seguridad del repositorio de datos.

Los objetos gestionados mediante este tipo de servicios web se listan en la tabla 3.28.

Tabla 3.28 Clases de los objetos gestionados por los servicios web implementados con la interfaz IRestCRUDController.

Clase	Descripción	Ruta
Activity	Actividades académicas.	/activities
ActivityType	Tipos de actividades académicas.	/activity-types
Area	Áreas de estudio.	/areas
Level	Niveles de estudio.	/levels
Student	Estudiantes.	/students

La ejecución de los servicios web retornará como respuesta un código de estado HTTP 200 (OK) en caso de que la ejecución sea exitosa, y un código diferente en caso de algún error.

Cuando la ejecución sea exitosa, al código de estado acompañará una respuesta en formato JSON con el objeto u objetos resultantes de la ejecución de la ejecución, a excepción del servicio de eliminación, el cual solo retornará el código de estado indicando que la eliminación fue exitosa o tuvo un error.

Servicios web para la generación de predicciones

Para la generación de predicciones se construyó un conjunto de servicios web que se encuentran en la ruta: /predictions. La lista de los servicios web implementados se muestra en la tabla 3.29 y todos ellos se ejecutan bajo la ruta antes mencionada.

Tabla 3.29 Servicios web para la ejecución de predicciones.

Ruta	Descripción
/by-area-and-activity-type	Genera las predicciones de calificaciones en base al ID del estudiante, al área de estudio, el tipo de actividad y un rango de fechas.
/by-student	Genera las predicciones de calificaciones en base al ID del estudiante y a un rango de fechas.
/by-area	Genera las predicciones de calificaciones en base al ID del estudiante, el área de estudio y a un rango de fechas.

/by-activity-type	Genera las predicciones de calificaciones en base al ID del estudiante, a un tipo de actividad y a un rango de fechas.

Fara facilitar su comprensión y uso, se ha documentado el API utilizando el componente de Spring Framework llamado SpringDoc, el cual hace una implementación de OpenAPI 3 para generar la documentación de manera automática. SpringDoc también despliega una herramienta de visualización de la documentación creada mediante la herramienta Swagger-UI. La tabla 3.30 muestra las rutas de acceso a los componentes más importantes de la documentación, todas las rutas son relativas a la raíz de la implementación del servidor del API.

Tabla 3.30 Rutas de acceso a la documentación del API para la generación de predicciones.

Ruta	Descripción
/api/public/doc/v1	Documento en formato JSON de la documentación del API. Este documento cumple con la especificación OpenAPI 3.
/api/public/doc/v1/index.html	Herramienta de navegación sobre la documentación del API.

La figura 3.25 muestra un ejemplo de la herramienta de navegación en la documentación implementada utilizando Swagger.

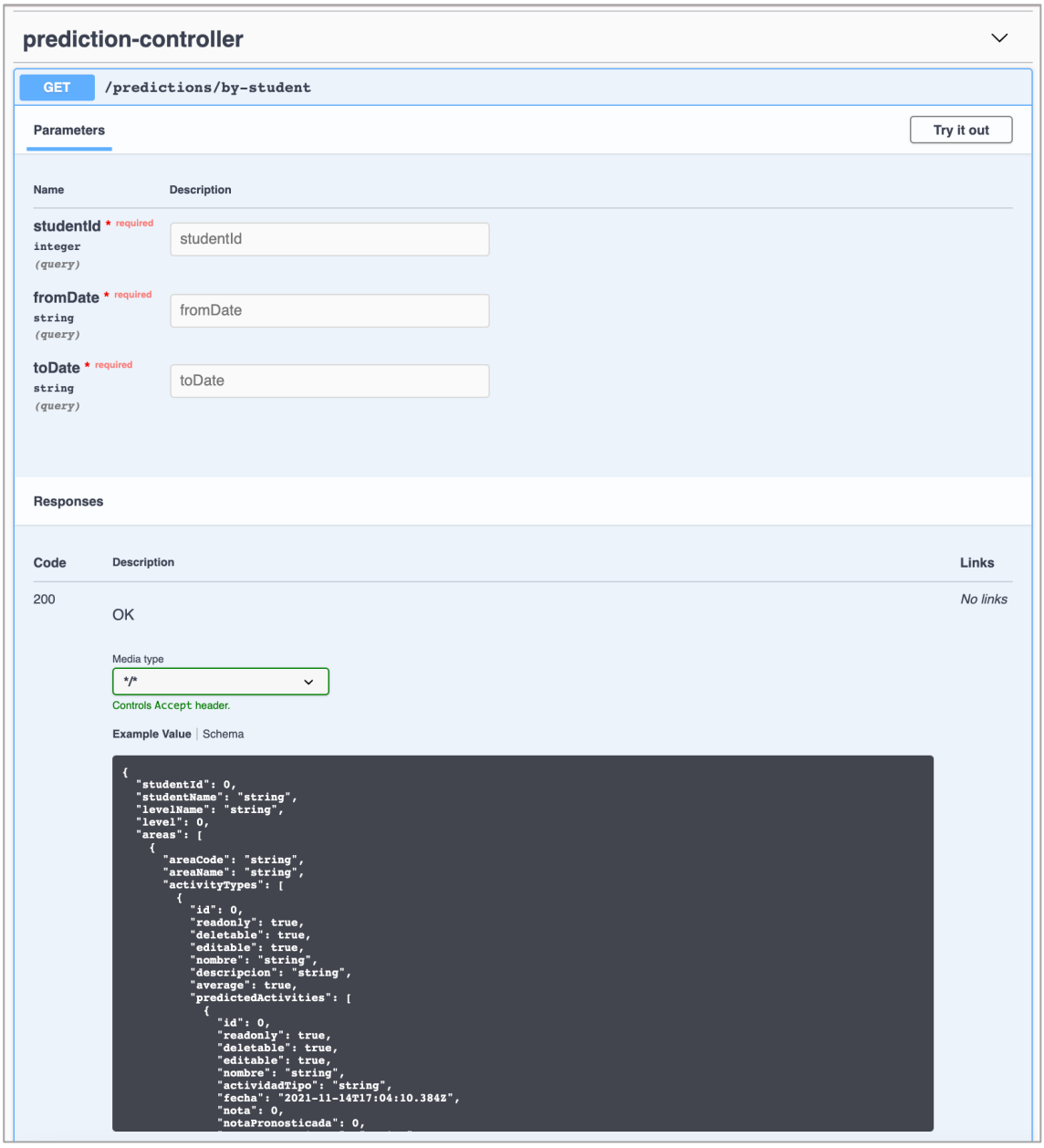


Figura 3.20 Ejemplo de la herramienta de navegación en la documentación del API para la generación de predicciones.

Aplicación web

Para realizar las predicciones se elaboró un prototipo de la aplicación web que se pondrá en producción. Este prototipo permite acceder a la información de la base de datos académica de la Unidad Educativa que nos facilitó la información para la elaboración del presente trabajo.

La aplicación web se desarrolló como una SPA (Aplicación de una sola página por sus siglas en inglés) utilizando Angular Framework y hace uso del API REST descrito en el apartado anterior.

3.3 3.3 Infraestructura para procesamiento y almacenamiento

El procesamiento del entrenamiento del modelo se realizó utilizando la plataforma colaborativa de Google "*Google Collaboratory*". Para la ejecución del modelo en producción se implementaron los siguientes servidores en virtuales en la nube, todos ellos utilizan el sistema operativo CentOS 8:

Tabla 3.31 Infraestructura de procesamiento.

Servidor	Características	Contenido
hachiko.korigen.com	IP: 134.209.42.6 RAM: 4GB Disco: 50Gb vCPUs: 2	Base de datos académica, proceso de extracción de datos.
kobi.korigen.com	IP: 159.203.64.190 RAM: 8GB Disco: 160Gb vCPUs: 4	API REST para predicción, API REST para la aplicación web, almacenamiento del modelo, procesos de depuración de datos y de entrenamiento del modelo de predicción.
kaiser.korigen.com	IP: 134.209.168.225 RAM: 1GB Disco: 25Gb vCPUs: 1	Aplicación web para la generación de predicciones.

El dominio "*korigen.com*" es de propiedad del autor y es utilizado para la provisión de servicios en la nube tipo SaaS.

3.4 Plataformas y prototipos de visualización

La visualización de los resultados se realiza mediante una aplicación web desarrollada para el efecto. En el presente apartado detallamos su estructura y mostramos ejemplos de su operación.

En esta sección incluimos imágenes en las que se censura la información personal de los estudiantes por tratarse de menores de edad y en cumplimiento con el acuerdo de confidencialidad firmado con la Unidad Educativa proveedora de la información.

3.4.1 Acceso a la aplicación

La primera interfaz que se muestra al ejecutar la aplicación es la pantalla de ingreso de usuario y clave de acceso, la gestión de usuarios y privilegios se realiza dentro de la aplicación de gestión académica. La figura 3.21 muestra un ejemplo de la pantalla de control de acceso.

Figura 3.21 Pantalla de control de acceso a la aplicación web.

3.4.2 Operación general de las interfaces para predicción

Todas las interfaces que se han desarrollado para este prototipo tienen una estructura similar. La figura 3.22 muestra la estructura general de las interfaces y la tabla 3.32 contiene una descripción de cada bloque.

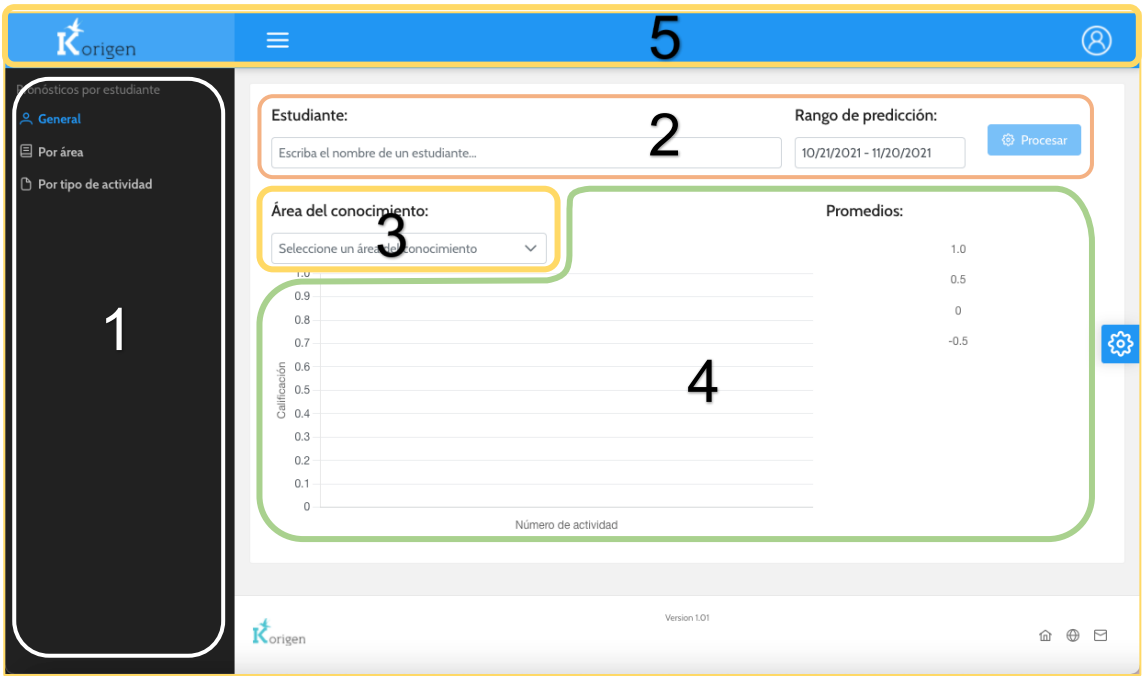


Figura 3.22 Estructura general de las interfaces de consulta de predicciones.

Tabla 3.32 Componentes de las interfaces gráficas para predicción.

Bloque	Descripción
1	Menú de opciones, permite acceder a las interfaces gráficas para realizar las predicciones.
2	Parámetros para la consulta de las predicciones, dependiendo de la interfaz en este bloque se mostrarán los parámetros necesarios para realizar la predicción. Una vez ingresados se presiona el botón “Procesar” para realizar la predicción. Como resultado se obtendrá un conjunto de información que se puede filtrar en el bloque número 3.
3	Filtro de las predicciones realizadas, este bloque permite filtrar la información de las predicciones ya ejecutadas.
4	Gráficos, contiene dos gráficos con la siguiente información: Gráfico de líneas: Presenta una evolución en el tiempo de las calificaciones. Gráfico de radar: Presenta un estado de los promedios del estudiante en cada área del conocimiento.
5	Barra de título, muestra opciones para colapsar el menú y para salir de la aplicación.

3.4.3 Codificación de colores de la información en los gráficos

Los gráficos codifican las series de datos con los siguientes colores:

Tabla 3.33 Codificación de colores de los gráficos.

Color		Descripción
Azul		Serie de datos anteriores a la predicción.
Naranja		Predicción en el rango de tiempo solicitado.
Verde		Datos reales en el rango de la predicción.

3.4.4 Predicción general

La interfaz de predicción general permite revisar el promedio de calificaciones obtenidas por el estudiante. Los elementos que muestra esta interfaz son:

Tabla 3.34 Elementos de la interfaz de Predicción General.

Sección	Elemento	Descripción
Parámetros para la consulta de las predicciones	Estudiante	Permite seleccionar un estudiante sobre el que se realizará la predicción.
	Rango de predicción	Selecciona el rango de fechas de la predicción.
	Procesar	Envía los parámetros al servicio web para realizar la predicción.
Filtro de las predicciones realizadas	Área del conocimiento	Permite seleccionar un área del conocimiento para la visualización de las predicciones realizadas.

En la figura 3.23 podemos ver un ejemplo de la información generada en esta interfaz.

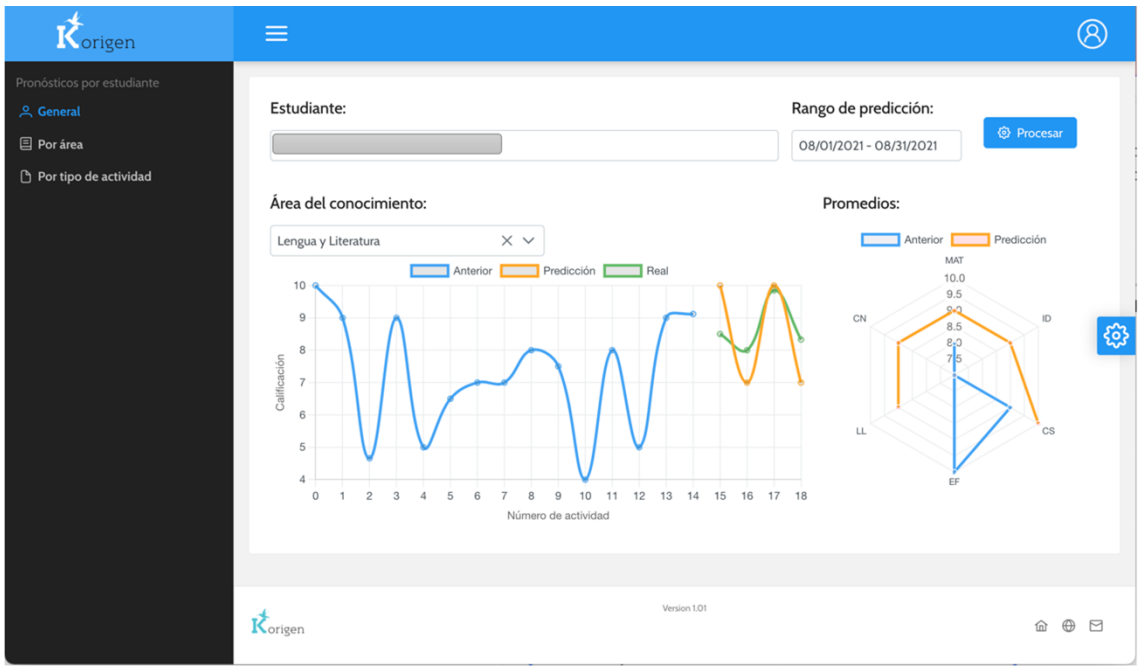


Figura 3.23 Ejemplo de la interfaz de predicción general.

En este ejemplo podemos apreciar la predicción para un estudiante en las actividades realizadas en el rango desde el 1 al 31 de agosto del 2021 en la asignatura de Lengua y Literatura.

3.4.5 Predicción por área

La interfaz de predicción por área permite revisar el promedio de calificaciones obtenidas por el estudiante en un área de conocimiento determinada. Los elementos que muestra esta interfaz son:

Tabla 3.35 Elementos de la interfaz de Predicción General.

Sección	Elemento	Descripción
Parámetros para la consulta de las predicciones	Estudiante	Permite seleccionar un estudiante sobre el que se realizará la predicción.
	Área del conocimiento	Permite seleccionar un área del conocimiento.
	Rango de predicción	Selecciona el rango de fechas de la predicción.
	Procesar	Envía los parámetros al servicio web para realizar la predicción.
Filtro de las predicciones realizadas	Tipo de actividad	Permite seleccionar un tipo de actividad para la visualización de las predicciones realizadas.

En la figura 3.24 podemos ver un ejemplo de la información generada en esta interfaz.

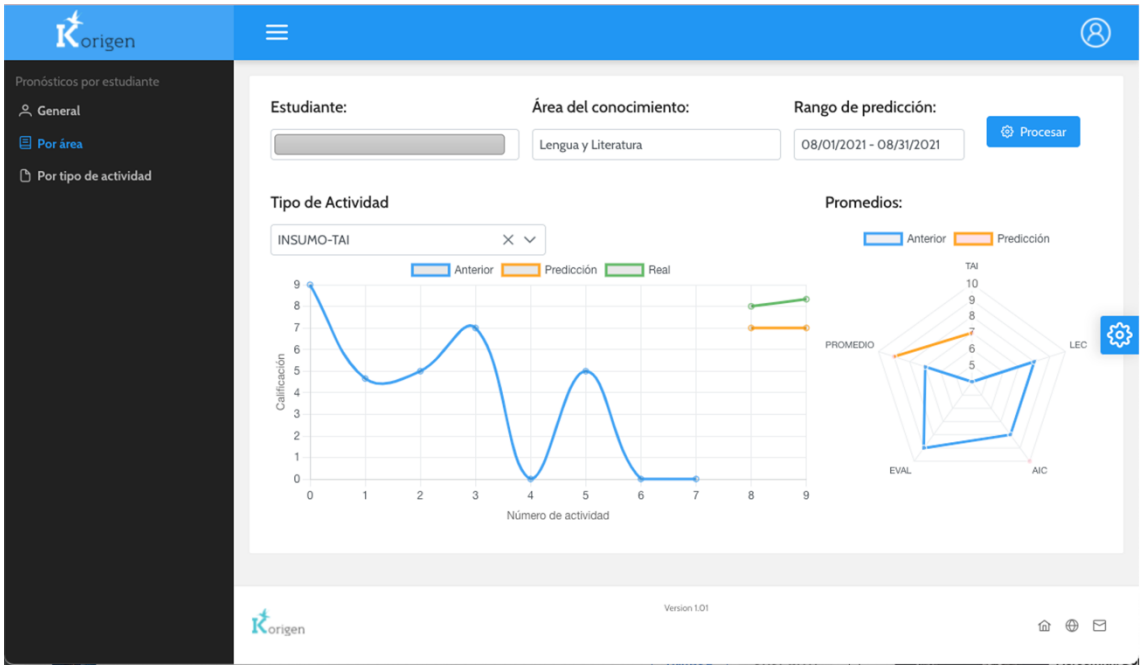


Figura 3.24 Ejemplo de la interfaz de predicción por área del conocimiento.

En este ejemplo podemos apreciar la predicción para un estudiante en las actividades realizadas en el rango desde el 1 al 31 de agosto del 2021 en la asignatura de Lengua y Literatura y para los tipos de actividad TAI (Trabajos académicos individuales).

3.4.6 Predicción por tipo de actividad

La interfaz de predicción por tipo de actividad permite revisar el promedio de calificaciones obtenidas por el estudiante en un tipo de actividad determinada. Los elementos que muestra esta interfaz son:

Tabla 3.36 Elementos de la interfaz de Predicción General.

Sección	Elemento	Descripción
Parámetros para la consulta de las predicciones	Estudiante	Permite seleccionar un estudiante sobre el que se realizará la predicción.
	Tipo de actividad	Permite seleccionar un tipo de actividad.
	Rango de predicción	Selecciona el rango de fechas de la predicción.
	Procesar	Envía los parámetros al servicio web para realizar la predicción.

Filtro de las predicciones realizadas	Área del conocimiento	Permite seleccionar un área del conocimiento para la visualización de las predicciones realizadas.

En la figura 3.25 podemos ver un ejemplo de la información generada en esta interfaz.

Figura 3.25 Ejemplo de la interfaz de predicción por tipo de actividad.

En este ejemplo podemos apreciar la predicción para un estudiante en las actividades realizadas en el rango desde el 1 al 31 de agosto del 2021 en el tipo de actividad TAI (Trabajos académicos individuales) y en la asignatura de Lengua y Literatura.

3.5 Métricas y comunicación de resultados

La tabla 3.37 muestra los principales indicadores que utilizamos para medir el resultado del presente trabajo.

Tabla 3.37 Indicadores para medir los resultados obtenidos.

Indicador	Descripción
Exactitud (accuracy)	Mide el nivel de exactitud del modelo con respecto a las predicciones realizadas.
Pérdida (loss)	Es la acumulación de errores entre el conjunto de datos entrenamiento y el de validación. Este valor sirve para tener una referencia del valor a minimizar en el entrenamiento del modelo.

La comunicación de resultados la realizamos mediante las interfaces gráficas detalladas en el numeral *3.4 Plataformas y prototipos de visualización*.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS

4.1 Recolección de datos y estrategias para validación del proyecto

La información utilizada para el presente trabajo se extrajo de la base de datos de una Unidad Educativa particular de educación primaria y secundaria en la provincia de Santa Elena. Para extraer la información se construyó una sentencia SQL que al ejecutarse envía el resultado a un archivo en formato CSV. Este archivo luego es pasado por un proceso de depuración previo a utilizarlo en el entrenamiento del modelo. El dataset resultante se analiza con profundidad en el capítulo 3 de este trabajo.

La base de datos de la Unidad Educativa está en constante evolución, por lo que se ha definido una arquitectura que permite implementar una estrategia de entrenamiento continuo del modelo en base a la información que se genera diariamente.

El proyecto se evaluó en base a las métricas que se definieron al final del capítulo 3 y los valores de estas métricas se obtuvieron como resultado del entrenamiento del modelo.

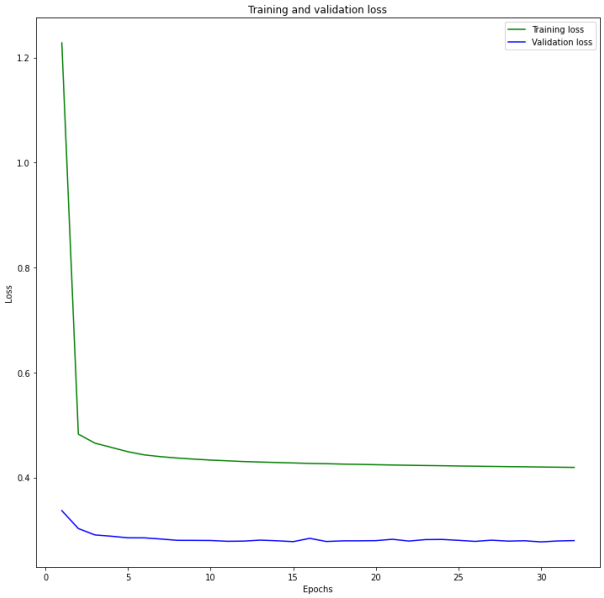


Figura 4.1 Gráfico de la evolución del entrenamiento del modelo.

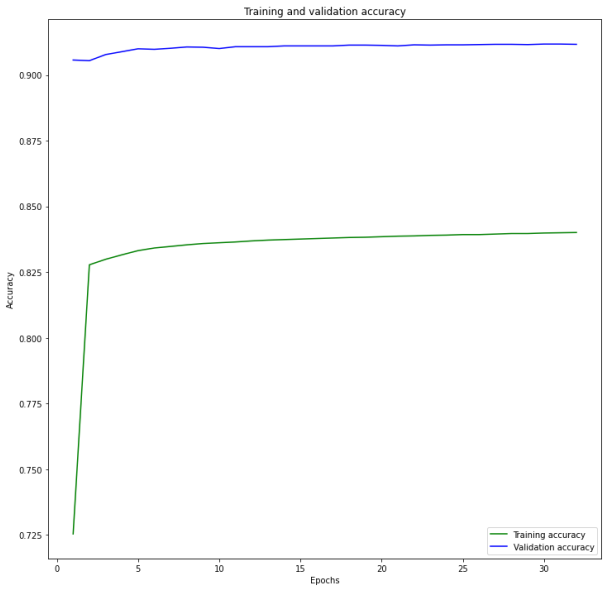


Figura 4.2 Gráfico de la evolución de la exactitud del modelo.

Los gráficos 4.1 y 4.2 ilustran la evolución del entrenamiento del modelo, en lo que corresponde a la pérdida ("*loss*") y la exactitud ("*accuracy*"). La tabla 4.1 muestra los resultados de las métricas obtenidas en las primeras 32 épocas del entrenamiento.

Tabla 4.1 Resultados de las métricas en los primeros 32 entrenamientos.

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	1.2281	0.7253	0.3371	0.9057
2	0.4827	0.8278	0.3029	0.9054
3	0.4657	0.8299	0.2908	0.9078
4	0.4575	0.8316	0.2883	0.9089
5	0.4493	0.8332	0.2853	0.9100
6	0.4432	0.8342	0.2852	0.9098
7	0.4398	0.8348	0.2830	0.9102
8	0.4372	0.8354	0.2805	0.9107
9	0.4352	0.8359	0.2804	0.9106
10	0.4334	0.8362	0.2802	0.9101
11	0.4320	0.8365	0.2787	0.9108
12	0.4306	0.8369	0.2790	0.9108
13	0.4296	0.8372	0.2809	0.9108
14	0.4287	0.8374	0.2797	0.9111
15	0.4279	0.8376	0.2780	0.9111
16	0.4270	0.8378	0.2843	0.9111

17	0.4265	0.8380	0.2783	0.9111
18	0.4257	0.8382	0.2795	0.9114
19	0.4252	0.8383	0.2796	0.9114
20	0.4246	0.8385	0.2799	0.9113
21	0.4240	0.8387	0.2825	0.9111
22	0.4235	0.8388	0.2791	0.9115
23	0.4230	0.8390	0.2819	0.9114
24	0.4226	0.8391	0.2822	0.9115
25	0.4221	0.8393	0.2805	0.9115
26	0.4217	0.8393	0.2785	0.9116
27	0.4213	0.8395	0.2808	0.9117
28	0.4209	0.8397	0.2790	0.9117
29	0.4206	0.8397	0.2797	0.9116
30	0.4201	0.8399	0.2775	0.9118
31	0.4198	0.8400	0.2793	0.9118
32	0.4194	0.8401	0.2800	0.9117

Podemos apreciar en la tabla que en las primeras 32 épocas el modelo obtuvo una exactitud de 0.8401 con una pérdida de 0.4194. En lo que respecta a la validación, la exactitud es de 0.9117 con una pérdida de 0.28,

4.2 Puesta en marcha y funcionamiento

Si bien es cierto que el prototipo se desarrolló utilizando los servicios de *Google Collaboratory*, debemos considerar que la puesta en marcha del prototipo para su explotación en producción se realizó utilizando servicios en la nube.

Una vez desplegados los servidores se procedió a instalar los siguientes componentes en cada uno de ellos:

Tabla 4.2 Distribución de los componentes de la solución en los servidores implementados.

Servidor	Componentes instalados	Implementado con…
hachiko.korigen.com	Base de datos académica	PostgreSQL 10
	Proceso de extracción de datos	Shell y Cron de Linux
kobi.korigen.com	Almacenamiento del modelo entrenado	Sistema operativo
	Pre-Procesamiento de datos	Python 3
	Entrenamiento del modelo de predicción	Python 3

	API REST para interacción con el servidor Web	Java 8
	API REST para uso del modelo de predicción	Python 3 y Django
kaiser.korigen.com	Aplicación web	Apache web server

4.3 Pruebas de funcionalidad

La figura 4.3 muestra un ejemplo de predicción para un estudiante en la asignatura de ciencias naturales sobre actividades planificadas para el periodo entre el 1 y 30 de septiembre del 2021. Con la finalidad de apreciar la predicción, en este caso se ocultó la línea de predicción.

Figura 4.3 Ejemplo de una predicción general de calificaciones.

La figura 4.4 muestra las predicciones realizadas para el mismo estudiante.

Figura 4.4 Ejemplo de una predicción general de calificaciones.

Podemos apreciar en este caso que la única predicción significativamente desproporcionada es la número 56, mientras que el resto de las predicciones se ajusta bastante bien a los datos reales. También se puede apreciar que la predicción del promedio final del estudiante se ajusta significativamente bien a los datos reales.

Por otra parte, en la figura 4.5 se muestra la ejecución de la predicción general para un segundo estudiante. Podemos apreciar que, en este caso, a pesar de que la predicción tiene un margen considerable de error, el comportamiento de las calificaciones se predice con bastante similitud.

Figura 4.5 Ejemplo de una predicción general de calificaciones.

En lo que respecta a los indicadores para evaluar el performance del modelo, se obtuvieron los siguientes resultados:

Tabla 4.3 Valores de los indicadores para evaluación del trabajo.

Indicador	Valor
Exactitud (accuracy)	0.8374
Pérdida (loss)	0.4283

A pesar de que el indicador de pérdida refleja aun un conjunto de errores importante, debemos resaltar que la exactitud del modelo es bastante aceptable, aunque aun no llega a los niveles de exactitud del estado del arte, los cuales están por encima del 0.9 (90%).

4.4 Análisis costo/beneficio

Para analizar el costo/beneficio de implementar la herramienta de software propuesta en el presente trabajo, debemos considerar que se integrará como un módulo adicional que agregará valor a una aplicación que se ofrece en la nube bajo la modalidad de "*Software As A Service*" (SaaS por sus siglas en inglés), por lo tanto, el presente análisis se realiza

desde la perspectiva del incremento esperado en las ventas como producto de la integración de este componente adicional.

4.4.1 Costos

Los costos iniciales relacionados al presente trabajo se reflejan en la tabla 4.4 y se agrupan en dos categorías: desarrollo del módulo y el paso del prototipo a producción.

Tabla 4.4 Costos relacionados al trabajo.

Categoría	Valor ($)
Desarrollo	3,477.27
Paso del prototipo a producción	3,809.09
	7,286.36

Estos valores se consideran la inversión inicial y será recuperada con la venta del módulo a los nuevos clientes.

La tabla 4.5 muestra el costo de operación mensual del módulo en la nube en el ítem "*Operación mensual por los clientes actuales*". Este valor correspondiente al costo del incremento en los recursos de los servidores para la operación del módulo desarrollado. El ítem denominado "*Operación mensual por cada nuevo cliente*" se refiere al costo mensual del incremento necesario en los recursos de los servidores para soportar un nuevo cliente que utilice el módulo de predicción.

Tabla 4.5 Costos relacionados al trabajo.

Costo mensual	Valor ($)
Operación mensual por los clientes actuales	211.05
Operación mensual por cada nuevo cliente	20.00

4.4.2 Beneficios

El beneficio principal esperado con este trabajo y su adición a la aplicación comercializada en la nube es el incremento del portafolio de servicios y con ello la atracción de nuevos clientes además de la fidelización de los actuales. Desde esta perspectiva se realiza el análisis de los beneficios esperados desde tres escenarios:

pesimista, esperado y optimista; estos tres escenarios se detallan en la tabla 4.6, en donde se describe lo que implica cada uno de ellos, así como el incremento anual de clientes que contratará el módulo en cada escenario.

Tabla 4.6 Escenarios para el análisis del costo/beneficio.

Escenario	Descripción	Clientes
Pesimista	Ningún cliente nuevo contrata el módulo.	0
Esperado	Al menos el 50% de los clientes nuevos contrata el módulo.	2
Optimista	Todos los clientes nuevos contratan el módulo.	4

El promedio de nuevos clientes al año para la aplicación completa, en valores enteros, es de 4 y el presente análisis se realiza con la premisa de que este promedio se mantiene, a pesar de que con la implementación del nuevo módulo se espera un incremento en esta cantidad.

De la cartera actual de clientes, cuatro de ellos han manifestado su interés en la contratación del módulo de predicción académica una vez que se encuentre en producción, por razones de reserva los denominaremos: Cliente A, Cliente B, Cliente C y Cliente D respectivamente. La modalidad de facturación del servicio se indexa a la cantidad de estudiantes y el precio de lanzamiento del módulo será de $ 1,00 (un dólar) anual por estudiante. La tabla 4.7 muestra un resumen del análisis realizado:

Tabla 4.7 Resumen de ingresos esperados por la comercialización del módulo.

Cliente	Cantidad de estudiantes	Valor por estudiante	Total
Cliente A	863	1.00	863.00
Cliente B	631	1.00	631.00
Cliente C	184	1.00	184.00
Cliente D	962	1.00	962.00
			2,640.00

El promedio de estudiantes de las Unidades Educativas es de alrededor de 650, por lo que utilizaremos esta cifra para los cálculos de las proyecciones anuales de ingresos y gastos. Los resultados para cada escenario en 5 periodos consecutivos son:

Tabla 4.8 Resultados para los escenarios pesimista, esperado y optimista.

Escenario pesimista: Ningún cliente nuevo contrata el módulo.						
Periodo	0	1	2	3	4	5
Clientes nuevos		0	0	0	0	0
Ingresos		2,640.00	2,640.00	2,640.00	2,640.00	2,640.00
Costos		211.05	211.05	211.05	211.05	211.05
Resultado	-7,686.36	2,428.95	2,428.95	2,428.95	2,428.95	2,428.95

Escenario esperado: Al menos el 50% de los clientes nuevos contrata el módulo.						
Periodo	0	1	2	3	4	5
Clientes nuevos		2	4	6	8	10
Ingresos		3,940.00	5,240.00	6,540.00	7,840.00	9,140.00
Costos		251.05	291.05	331.05	371.05	411.05
Resultado	- 7,686.36	3,688.95	4,948.95	6,208.95	7,468.95	8,728.95

Escenario optimista: Todos los clientes nuevos contratan el módulo.						
Periodo	0	1	2	3	4	5
Clientes nuevos		4	8	12	16	20
Ingresos		5,240.00	7,840.00	10,440.00	13,040.00	15,640.00
Costos		291.05	371.05	451.05	531.05	611.05
Resultado	- 7,686.36	4,948.95	7,468.95	9,988.95	12,508.95	15,028.95

Con estos resultados realizamos el cálculo del "*Valor Actual Neto*" (VAN) y la "*Tasa Interna de Retorno*" (TIR), para cada escenario, con la finalidad de evaluar financieramente el proyecto. La tasa de retorno considerada para estos cálculos es del 10%. En la tabla 4.9 se muestran los valores de estos indicadores.

Tabla 4.9 Resultados para los escenarios pesimista, esperado y optimista.

Escenario	**VAN**	**TIR**
Pesimista	1,521.27	17%
Esperado	14,943.53	60%
Optimista	28,365.79	90%

Podemos apreciar que la TIR, aún en el escenario pesimista, es mayor a la tasa de retorno del proyecto (10%), esto nos indica que los beneficios esperados del proyecto son superiores a los costos incurridos para su desarrollo lo que nos permite concluir que su ejecución es completamente viable.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. El conjunto de atributos seleccionados por su significancia para la predicción de calificaciones (detallados en la tabla 3.10) es consistente con lo anticipado en el estado del arte, pues representan información demográfica, social y de calificaciones anteriores del estudiante, sin embargo, durante el análisis se encontró que la nota obtenida también es afectada por factores externos tales como: el tipo de actividad, su frecuencia, el docente y además, en este caso, por la pandemia del COVID-19.
2. La utilización de un modelo que implementa el procesamiento de series temporales mediante redes neuronales tipo “transformer” nos permitió aprovechar las características de este tipo de redes tales como el procesamiento en paralelo, la detección y aprendizaje de patrones, la memoria a largo plazo y el mecanismo de atención para mejorar el desempeño de la predicción. Esta característica se evidencia en la imagen 4.5, en donde se muestra que la predicción del comportamiento de las calificaciones es bastante aproximada a la realidad a pesar de que las calificaciones anteriores no evidencian un patrón definido. Este comportamiento demuestra la capacidad del modelo de interpretar la incidencia de todos los parámetros al realizar la predicción.
3. A pesar del escaso entrenamiento del modelo las predicciones alcanzadas evidencian su eficiencia al pronosticar el comportamiento de las calificaciones en forma coherente y muy cercana a la realidad. Esto es consistente con los resultados detallados en la tabla 4.1, en donde se refleja una exactitud (accuracy) del 84.01% en el entrenamiento y un 91.17% en la validación.
4. Los requerimientos técnicos necesarios para la implementación de los diversos componentes de software desarrollados en el presente trabajo, y detallados en la tabla 3.31, dificultan su instalación en cada unidad educativa, lo que genera una excelente oportunidad de provisión como una herramienta de tipo SaaS.

5. La herramienta de software desarrollada en este trabajo muestra el estado del rendimiento académico de un estudiante con corte a una fecha y permite pronosticar su comportamiento en un rango de tiempo, con base en las actividades planificadas con antelación. Incluye además una vista proyectada de su nivel académico en todas las áreas del conocimiento, de manera que los directivos institucionales pueden tener una visión global del estado cognitivo actual y futuro del estudiante y con ello tomar decisiones oportunas.

RECOMENDACIONES

1. Incrementar la cantidad de épocas de entrenamiento con la finalidad de reducir la pérdida y aumentar la precisión previo a la implementación del modelo en entornos de producción. Para ello es recomendable el uso de plataformas especializadas en el entrenamiento de modelos de machine learning.
2. Generar modelos independientes para cada entidad, entrenados con sus propios datos. Esto con la finalidad de que los modelos aprendan de las condiciones particulares de cada institución mejorando de esta manera la exactitud de sus predicciones. Sin embargo, debido a la gran cantidad de tiempo y recursos que requiere el entrenamiento de las redes neuronales, se recomienda utilizar técnicas de "Transfer Learning" y aprovechar el conocimiento almacenado en modelos pre-entrenados como base para los modelos individuales.
3. Realizar convenios con entidades que puedan proveer la información necesaria para la generación de los modelos pre-entrenados. Es importante que los convenios contengan cláusulas de confidencialidad y anonimización de datos que permitan acceder a la información de las entidades de manera confiable y segura.
4. Utilizar la metodología de selección de atributos detallada en el capítulo 3 para analizar e incluir nuevos ítems al modelo de predicción, con la finalidad de optimizar en forma progresiva la exactitud de las predicciones.

REFERENCIAS BIBLIOGRAFICAS

Libros

Amirah, M. S., Wahidah, H., Nur'aini, A. R. (2015). A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science.

Arteaga, J. A. B., & Ibarra, G. M. (2020). Factores que influyen en el aprendizaje del inglés de los bachilleres de Pasto, Colombia. Folios.

D. M. S. Anupama Kumar, Appraising the significance of self regulated learning in higher education using neural networks, International Journal of Engineering Research and Development Volume 1 (Issue 1) (2012) 09–15

Lau, E. T., Sun, L., & Yang, Q. (2019). Modelling, prediction and classification of student academic performance using artificial neural networks. SN Applied Sciences.

Muñoz, M. L., Alonso, R. D. H., & Marqués, N. M. (2019). Estudio de nuevos modelos de Deep Leaning para el análisis y comprensión de grandes cantidades de datos.

Saa, A. A., Al-Emran, M., & Shaalan, K. (2019). Factors affecting students' performance in higher education: a systematic review of predictive data mining techniques. Technology, Knowledge and Learning, 24(4), 567-598.

Tsiakmaki, M., Kostopoulos, G., Kotsiantis, S., & Ragos, O. (2020). Transfer learning from deep neural networks for predicting student performance. Applied Sciences.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

Artículos presentados en conferencias

Fierro, A. A. (2021). Predicción de series temporales con redes neuronales (Doctoral dissertation, Universidad Nacional de La Plata).

Hellas, A., Ihantola, P., Petersen, A., Ajanovski, V. V., Gutica, M., Hynninen, T., ... & Liao, S. N. (2018, July). Predicting academic performance: a systematic literature review. In Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education.

Iyanda, A. R., Ninan, O. D., Ajayi, A. O., & Anyabolu, O. G. (2018). Predicting Student Academic Performance in Computer Science Courses: A Comparison of Neural Network Models. International Journal of Modern Education & Computer Science

Sivasakthi, M. (2017, November). Classification and prediction based data mining algorithms to predict students' introductory programming performance. In 2017 International Conference on Inventive Computing and Informatics (ICICI) (pp. 346-350). IEEE.

Libros en línea

Srivastava, K. (2021, April 25). Student academic performance during covid | Towards Data Science. Medium. https://towardsdatascience.com/evaluation-and-prediction-of-students-academic-performance-during-covid-19-40bb2b90141b

Instituto Nacional De Estadistica Y Censos, I. N. E. C. (2013). Estimaciones de proyecciones de población 2010 INEC. Instituto Nacional de Estadistica y Censos. https://www.ecuadorencifras.gob.ec/documentos/web-inec/Poblacion_y_Demografia/Proyecciones_Poblacionales/presentacion.pdf

Páginas web

¿Qué es el cloud computing? (2018, March 14). Redhat. https://www.redhat.com/es/topics/cloud

¿Qué es Front end y para qué sirve? (2020, December 31). NeoAttack. https://neoattack.com/neowiki/front-end/

¿Qué es un SaaS? (2019, July 25). Redhat. https://www.redhat.com/es/topics/cloud-computing/what-is-saas

¿Qué es una API de REST? (2020, May 8). Redhat. https://www.redhat.com/es/topics/api/what-is-a-rest-api

¿Qué es una API? (2017, October 31). Redhat. https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces

Attal, M. (2021, December 13). Machine Learning: definición, funcionamiento, usos. Formación en ciencia de datos | DataScientest.com. https://datascientest.com/es/machine-learning-definicion-funcionamiento-usos

Attal, M. (2022, January 6). ¿Qué es el Transfer Learning? Formación en ciencia de datos | DataScientest.com. https://datascientest.com/es/que-es-el-transfer-learning

Castillo, J. A. (2020, February 2). Token, Token Ring – qué es, definición y para qué sirven. Profesional Review. https://www.profesionalreview.com/2020/02/21/token-token-ring-que-es/

Collaguazo, D. (2019, January 8). ¿Qué es el Procesamiento de Lenguaje Natural y cómo ponerlo en práctica con recursos abiertos? Inter-American Development Bank. https://blogs.iadb.org/conocimiento-abierto/es/que-es-el-procesamiento-de-lenguaje-natural-y-como-ponerlo-en-practica-con-recursos-abiertos/

CRUD - Glosario | MDN. (2022, February 23). MSDN Web Docs. https://developer.mozilla.org/es/docs/Glossary/CRUD

Descripción general de diferentes optimizadores para redes neuronales. (2020, November 26). ICHI.PRO. https://ichi.pro/es/descripcion-general-de-diferentes-optimizadores-para-redes-neuronales-247308806799555

Edix, R. (2021, September 13). Framework: qué es, para qué sirve y algunos ejemplos. Edix España. https://www.edix.com/es/instituto/framework/

El modelo de redes neuronales. (n.d.). IBM. https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model

Función de funciones de pérdida (MicrosoftML) - SQL Server Machine Learning Services. (2021, October 13). Microsoft. https://docs.microsoft.com/es-es/sql/machine-learning/r/reference/microsoftml/loss?view=sql-server-ver15

Qué es una web SPA | Single Page Applications. (2020, March 12). Incentro. https://www.incentro.com/es-ES/blog/que-es-web-simple-page-applications

GLOSARIO

API

na API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones. (¿Qué es una API?, 2017)

API REST

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST). (¿Qué es una API de REST?, 2020)

Cloud computing

El cloud computing es la ejecución de las cargas de trabajo en las nubes, las cuales son entornos de TI que extraen, agrupan y comparten recursos flexibles en una red. (¿Qué es el cloud computing?, 2018)

CRUD

CRUD (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, pero también pude aplicar a funciones de un nivel superior de una aplicación como soft deletes donde la información no es realmente eliminada, sino es marcada como eliminada a través de un estatus. (CRUD - Glosario | MDN, 2022)

Framework

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software. (Edix, 2021)

Front end

El Front end es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos y la que permite a los visitantes navegar por las diferentes páginas mientras lo deseen. (¿Qué es Front end y para qué sirve?, 2020)

Función de pérdida

Una función de pérdida mide la discrepancia entre la predicción de un algoritmo de aprendizaje automático y la salida supervisada y representa el costo de equivocarse. (Función de funciones de pérdida (MicrosoftML) - SQL Server Machine Learning Services, 2021)

HTTP

El protocolo de transferencia de hipertexto (HTTP por sus siglas en inglés) fue diseñado para la comunicación de clientes con servidores mediante el intercambio de documentos elaborados con HTML, aunque actualmente se usa para transferir más tipos de información. HTTP contiene, en su definición, algunos verbos que indican la acción que se desea realizar con un recurso en particular, estos verbos son aprovechados por la arquitectura REST para la interacción entre los productores y consumidores.

JSON

La notación de objetos Javascript (JSON por sus siglas en inglés), es un conjunto de convenios para la representación e intercambio de datos en formato texto que se caracteriza por se ligero y fácil de interpretar.

LMS

Los sistemas de gestión de aprendizaje (LMS por sus siglas en inglés) son herramientas que facilitan la interacción entre estudiantes y docentes mediante una plataforma

informática en la que se definen, distribuyen y controlan actividades relacionadas al proceso de enseñanza / aprendizaje.

Machine learning

El Machine Learning o aprendizaje automático es un campo científico y, más particularmente, una subcategoría de inteligencia artificial. Consiste en dejar que los algoritmos descubran “patterns”, es decir, patrones recurrentes, en conjuntos de datos. Esos datos pueden ser números, palabras, imágenes, estadísticas, etc. (Attal, 2021)

Procesamiento de lenguaje natural (NLP)

El Procesamiento del Lenguaje Natural (PLN), en inglés Natural Language Processing, es un campo de las ciencias de la computación e ingeniería que se ocupa de facilitar la interacción humana con las máquinas a través del uso del lenguaje natural o lenguaje humano. El Procesamiento del Lenguaje Natural ocurre es a través de un proceso en el cual la máquina, que solamente entiende un lenguaje binario de ceros y unos, es entrenada para entender el lenguaje humano. (Collaguazo, 2019)

Optimizador

Los optimizadores actualizan los parámetros de peso para minimizar la función de pérdida. La función de pérdida actúa como guía para el terreno, indicando al optimizador si se está moviendo en la dirección correcta para llegar al fondo del valle, el mínimo global. (Descripción general de diferentes optimizadores para redes neuronales, 2020)

Red neuronal

Una red neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas. (El modelo de redes neuronales, n.d.)

REST

La arquitectura de transferencia de estado representacional (REST por sus siglas en inglés) fue definida en el 2020 en la tesis doctoral de Roy Fielding y su principal

característica es que utiliza HTTP para comunicar un productor (servidor) con un consumidor (cliente).

SaaS

El software como servicio (SaaS) es una forma de cloud computing que ofrece a los usuarios una aplicación en la nube junto con toda su infraestructura de TI y plataformas subyacentes. (¿Qué es un SaaS?, 2019)

Singe page application

Una web SPA o single page application se refiere a una forma de desarrollo web en la que la página web está contenida en un único archivo. De esta forma, se carga todo en HTML y, mientras naveguemos por la página, irá solicitando el contenido al servidor. (Qué es una web SPA | Single Page Applications, 2020)

Token

Representa un objeto o símbolo (seria esa su traducción al español), que puede ser tanto software como hardware que representa la capacidad o derecho de realizar una operación. (Castillo, 2020)

Transfer Learning

El Transfer Learning, o aprendizaje transferido en español, se refiere al conjunto de métodos que permiten transferir conocimientos adquiridos gracias a la resolución de problemas para resolver otros problemas. (Attal, 2022)