UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Sistema informático para la predicción de certificado y abandono en entornos educativos en línea

Autor: Guiomar Herrero Gajate
Tutor: Ruth Cobos Pérez

junio 2021

Guiomar Herrero Gajate
Sistema informático para la predicción de certificado y abandono en entornos educativos en línea

Guiomar Herrero Gajate
C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

AGRADECIMIENTOS

Me gustaría agradecer a Ruth Cobos la oportunidad de desarrollar este proyecto.

A mi familia y amigos, por apoyarme y estar siempre ahí, sin vosotros esto no habría sido posible.

A Sorin y a Unai por ayudarme cuando lo he necesitado.

Resumen

Los MOOCs son cursos masivos online que cada vez cuentan con más popularidad. Estos cursos generan una gran cantidad de datos sobre las interacciones de los estudiantes con el curso. Esta información puede ser utilizada para predecir si los estudiantes van a abandonar el curso u obtener su certificación utilizando técnicas de clasificación con Aprendizaje Automático.

La Analítica del Aprendiaje o Learning Analytics (LA) es un tema de estudio emergente que se centra en recopilar, analizar y visualizar estos datos con el objetivo de entender y optimizar el aprendizaje de los estudiantes y su entorno.

El objetivo de este Trabajo de Fin de Grado es proponer un estudio de predicción de abandono para un MOOC asíncrono y el desarrollo de un sistema informático, el cual se basa en una herramienta llamada *edX-MAS+ (edX-MAS+: Model Analyzer System for edX MOOC)* con tecnología actual e integrarlo al sistema *edX-LIMS (edX-LIMS: Learning Intervention Monitoring Service for edX MOOCs)*, sistema que proporciona información de los estudiantes del MOOC asíncrono *WebApp* mediante el uso de *Dashboards* y que cuenta con una interfaz gráfica donde se podrán ver los resultados obtenidos.

El sistema permitirá al usuario entrenar y comparar modelos predictivos con algoritmos de Aprendizaje Automático, tomando como entrada las interacciones de actividad del usuario, es decir, los estudiantes del curso online, con la plataforma (lo que llamaremos indicadores de actividad) para predecir el abandono y obtención del certificado.

Palabras clave

Cursos Masivos Abiertos Online, Analítica de Aprendizaje, Apredizaje Automático, Dashboard, Modelo Predictivo

Abstract

MOOCs (Massive Open Online Courses) courses are becoming more and more popular. These courses generate a large amount of data concerning the learner's interaction. This information can be used to predict if the student is going to acquire his certificate or dropout using classification techniques with Machine Learning.

Learning Analytics (LA) is an emerging topic of research that focuses on collection, analysis and visualization of those information in order to understand and optimize the student's learning process and environment.

The objective of this Final Degree Proyect is to propose a study of the dropout prediction for an asynchronous MOOC and to develop a computer system based on the program *edX-MAS+ (edX-MAS+: Model Analyzer System for edX MOOC)* with modern technology and to integrate it to the *edX-LIMS (edX-LIMS: Learning Intervention Monitoring Service for edX MOOCs)* system, which provides information about the students of the asynchronous MOOC *WebApp* through the use of *Dashboards* and that has a graphical interface where the user will see his results.

The system will allow the user to train and to compare predictive models with Machine Learning algorithms, taking as an input the student's interactions with the platform (what we will call indicators' activity) in order to predict his dropout and certificate.

Keywords

Masive Open Online Courses, Learning Analytics, Machine Learning, Dashboard, Predictive Model

Índice

Listas

Lista de ecuaciones

Lista de figuras

Lista de tablas

1

Introducción

En esta sección se encuentra el entorno general del proyecto, la motivación para realizarlo y los objetivos del mismo.

1.1. Motivación del proyecto

Pese a que el uso de la palabra ordenador data del siglo diecisitete, no fue hasta doscientos años más tarde que se empezó a conceptualizar el término como lo conocemos hoy en día.

Desde sus inicios, los ordenadores han sido máquinas usadas para la guerra, cuyas personas a cargo eran a menudo catedráticos y profesores contratados por los gobiernos para desarrollar la tecnología requerida [5]. Gracias a esto, los ordenadores siempre han estado ligados a la enseñanza, por lo que antes o después, esto iba a acabar repercutiendo en los modelos educativos.

Este avance tecnológico ha implicado el desarrollo de los cursos de aprendizaje en línea, destacando los Masive Open Online Courses (MOOCs) entre ellos, que desde 2012 se han convertido en un fenómeno global [6].

Los MOOCs o cursos masivos de aprendizaje online, son cursos donde cualquier persona puede inscribirse de manera gratuita. Dado que no hay un profesor que controle y motive a los estudiantes, estos deben tener una gran motivación [7] tanto para inscribirse, como para continuar la educación de manera más autodidacta que en una enseñanza tradicional.

En este tipo de cursos hay una gran disponibilidad de recursos, vídeos, foros de preguntas o problemas interactivos que entre otros, ayudan a los estudiantes a conseguir un aprendizaje más fluido y libre, enfatizando los temas que resulten más complejos de manera individual. Además, permiten una gran cantidad de estudiantes inscritos en ellos al mismo tiempo.

A modo de consecuencia, se genera una gran cantidad de datos sobre las interacciones de los distintos estudiantes con los cursos. Para que todos estos datos sean útiles, han de ser procesados y tratados de forma que se puedan entender las necesidades de los estudiantes, personalizar la ayuda a estos y adaptar el contenido de los MOOCs.

La Learning Analytics (LA) es una herramienta de estudio que se centra en medir, recolectar, analizar y generar datos nuevos sobre estudiantes y sus interacciones con los cursos online con el propósito de entender y optimizar el aprendizaje y los entornos de este [8]. LA cuenta con varias fases; la primera es la extracción y procesamiento; la segunda, es de selección y tratamiento, donde también se hacen las predicciones, y por último, la etapa de uso y mejoras, en la que se utilizan los datos obtenidos y se perfecciona el análisis [9].

Algunas plataformas como edX [10], recopilan MOOCs y proporciona los datos de los estudiantes a las distintas entidades académicas. La Universidad Autónoma de Madrid (UAM) es una de ellas, que posee cursos en la plataforma UAMx [11], como *Introducción al desarrollo de Aplicaciones Web en línea* [12], MOOC sobre el que se realizará este proyecto.

1.2. Objetivos

Después del análisis de la motivación, se procederá a exponer los objetivos del proyecto.

El objetivo global del proyecto es desarrollar un sistema informático para la predicción de certificado y abandono en un sistema MOOC asíncrono. Dicho sistema informático se basa en la herramienta *edX-MAS+* [3, 13], hecha sobre *Model Analyzer System para edX MOOC (edX-MAS)* [14, 15], el sistema informático se realizará con la misma tecnología del sistema *Learning Intervention Monitoring Service for edX MOOCs (edX-LIMS)* [4,16], para que se puedan integrar en un sistema Web. Estas herramientas *edX-MAS*, *edX-MAS+* y *edX-LIMS* han sido desarrolladas en anteriores TFGs, todos ellos bajo la tutorización de la tutora del actual TFG.

O-1.– Investigar sobre los MOOCs y entender su importancia para la educación actual.

O-2.– Conocer el estado del arte de la Analítica de Aprendizaje y de Aprendizaje Automático.

O-3.– Investigar sobre la herramienta *edX-MAS+*.

- **O-3.1.–** Entender el alcance de dicha herramienta mediante la documentación existente.

O-4.– Investigar sobre la herramienta *edX-LIMS*.

- **O-4.1.–** Entender el alcance de dicha herramienta mediante la documentación existente.
- **O-4.2.–** Instalar la aplicación e interactuar con ella.

O-5.– Creación de un algoritmo para predecir el abandono de los estudiantes en un MOOC asíncrono.

- **O-5.1.–** Explorar los datos recibidos para encontrar patrones comunes entre los estudiantes que abandonan el curso, haciendo un estudio previo de ello.
- **O-5.2.–** Programar el algoritmo de detección del abandono.

O-6.– Diseño y desarrollo del sistema informático para la predicción de certificado y abandono en entornos educativos en línea.

- **O-6.1.–** Permitir la actualización de datos para poder seguir usando la herramienta de cara al futuro.
- **O-6.2.–** Automatizar la creación de las nuevas tablas en la base de datos.

O-6.3.– Permitir la descarga de todos los datos mostrados en la aplicación en *csv*.

O-6.4.– Integración del nuevo sistema a la herramienta *edX-LIMS*.

O-6.5.– Desplegar el sistema en un servidor web.

O-6.6.– Realizar las pruebas necesarias para comprobar el correcto funcionamiento del sistema.

O-7.– Proponer posibles mejoras y trabajo futuro.

1.3. Organización del documento

Esta memoria se compone de los siguientes capítulos:

En el *primer capítulo* se ha introducido la motivación del proyecto y los objetivos de éste.

En el *segundo capítulo* se plantea el estado del arte sobre los MOOCs, analítica de aprendizaje y modelos de predicción. Así como el contexto previo, en el que se describen las heramientas *edX-LIMS* y *edx-MAS+* haciendo especial énfasis a las partes que se van a utilizar para este proyecto. También se comentará la tecnología que se va a utilizar.

En el *tercer capítulo* se detalla el estudio previo y proceso seguido para clasificar el abandono.

En el *cuarto capítulo* se plantea el desarrollo del módulo, detallando las fases, procedimientos y pruebas seguidas.

En el *quinto capítulo* se explicarán los resultados obtenidos, mostrando la interfaz gráfica del módulo.

En el *sexto capítulo* se incluyen las conclusiones y trabajo futuro que se podría realizar.

Como *anexos* se incluyen las tareas de la técnica *Thinking Aloud*, un ejemplo de primer uso del nuevo módulo y un ejemplo de cálculo del tiempo que tardan los algortimos en entrenar los modelos.

2

Estado del arte

En este capítulo se va a plantear el estado del arte de Learning Analytics (LA), los MOOCs, la predicción de datos y el contexto de partida. También se va a hacer un análisis del entorno tecnológico que se va a utilizar.

2.1. MOOCs

Los MOOCs son cursos abiertos que pueden contar con miles de usuarios y que comparten unas características en común. El primer MOOC fue creado en 2008, pero no fue hasta 2012 cuando se popularizaron, denominando ese año como *The year of the MOOC* [6, 17]. Gracias al avance tecnológico y la accesibilidad a los dispositivos electrónicos, estos cursos cuentan con miles de estudiantes inscritos.

Los MOOCs comparten ciertas características. Tienen que tener un carácter masivo, donde pueden matricularse miles de estudiantes desde cualquier país. Todo su contenido tiene que ser ordenado y estar disponible desde la web del curso, no necesitando recursos adicionales para poder superarlo [15]. Otra parte fundamental de estos cursos es el *conectivismo*, en el que los estudiantes se pueden ayudar, reforzando los conocimientos mutuos [18].

Desde su creación, ha habido muchas especulaciones de cómo evolucionaría la educación en el futuro, y se predijeron diferentes escenarios. Estos vaticinaban que la mayoría de la educación sería online [19]. Sin embargo, aunque el número de estudiantes y cursos no deja de crecer, estas espectativas no se han cumplido por el momento. En 2020 hay 180 millones de estudiantes cursando más de 16300 cursos en 950 universidades [20]. Esto supone 60 millones más de estudiantes, 2800 cursos nuevos y casi 50 universidades más que en 2019 [21]. Estas subidas en la demanda de cursos online en 2020 se deben principalmente al COVID-19, que ha hecho aumentar la demanda de MOOCs drásticamente [1], como se puede observar en la figura 2.1.

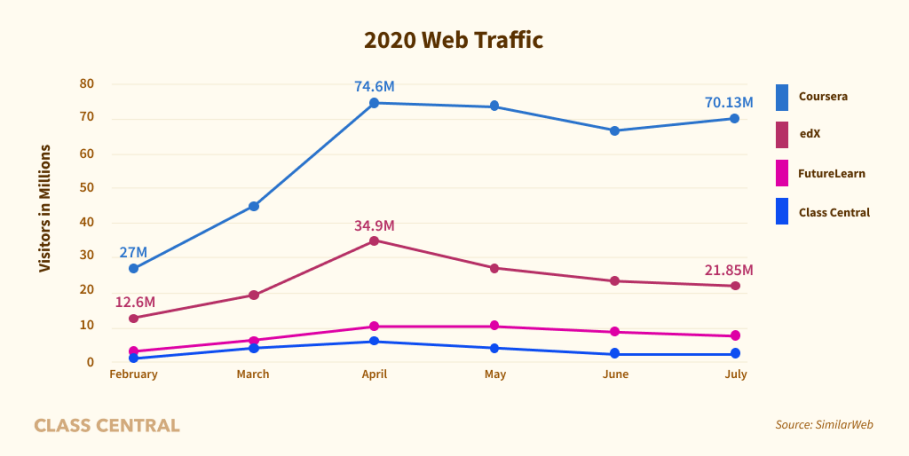

Figura 2.1: Crecimiento de estudiantes en MOOCs durante la Pandemia COVID-19, comparativa con las principales plataformas de cursos online (extraído de [1])

2.2. Learning Analytics y modelos de predicción

La **Analítica de Aprendizaje** o **Learning Analytics** fueron definidas por primera vez en 2011 [8] y hoy en día cuentan con una gran popularidad y grupos de investigación dedicandose a ellas. Su importancia se debe, entre otras cosas, al potencial que tienen las LA para mejorar la calidad de la educación [22].

Las LA surgen de la necesidad de analizar la cantidad masiva de datos que producen los recursos de aprendizaje electrónico, puesto que estos datos tienen gran importancia educacional para entender como aprenden los estudiantes y es imposible analizarlos a mano [23].

La red de investigadores Society for Learning Analytics Research (SOLAR) [24], es una de las más influyentes a día de hoy en el área de LA, uniendo a investigadores en este área alrededor del mundo [22]. En España está Spanish Network Of Learning Analytics (SNOLA) [25].

Estas redes establecen líneas de investigación atendiendo a las necesidades actuales de la educación y de los puntos de vista de estos investigadores. Entre estos objetivos se encuentran predecir variables de aprendizaje, predecir el abandono de estudiantes en MOOCs y la visualización de resultados para los profesores [26], entre otros.

Los **modelos de predicción** son algoritmos matemáticos que toman ciertas variables de entrada (como pueden ser los indicadores de actividad de usuarios) y predicen otras variables (como el aprobado o el abandono). Los algoritmos más utilizados hoy en día incluyen los mostrados en la figura 2.2, de los cuales se van a explicar resumidamente los cinco primeros:

Regresión lineal o logística: La lineal se utiliza para predecir valores concretos, la logística para clasificar entre dos opciones. Ambas utilizan un modelo estadístico que aproxima los datos de entrada a una recta.

Árboles de decisión o Random Forest: Algoritmos de aprendizaje supervisado. Los árboles de decisión se construyen con nodos, ramas y hojas que clasifican los resultados dependiendo de características y valores. *Random Forest* está formado por un grupo árboles de decisión diferentes, mejorando los resultados y eficiencia [27].

Gradient Boosting Machines: Junta escalonadamente diferentes algoritmos de regresión [28]

Redes neuronales convulocionales: Es un tipo de red neuronal en la que por lo menos hay una capa convolucional [29]. Una convolución es una operación matemática entre dos funciones que crea una tercera función.

Aproximaciones bayesianas: Modelos estadísticos basados en el Teorema de Bayes [30]

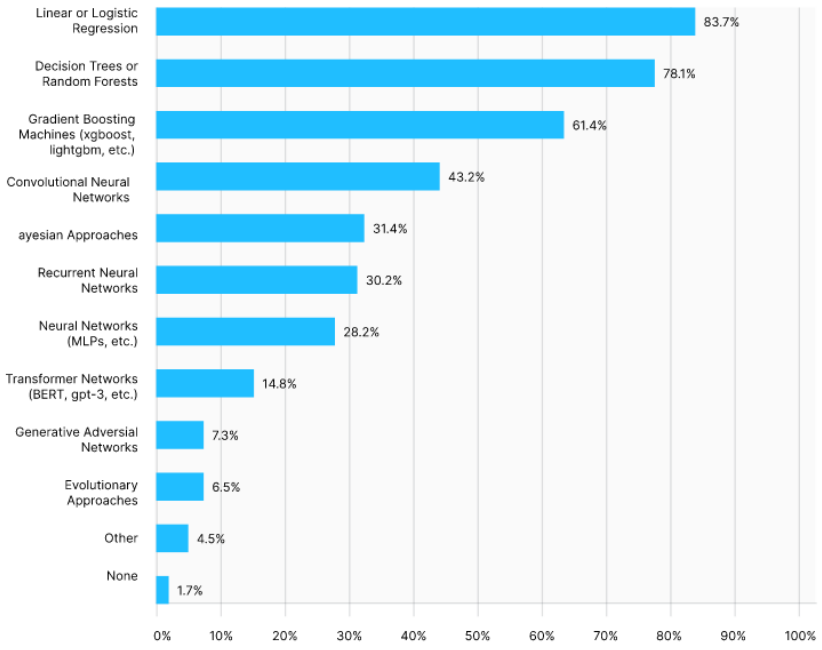

Figura 2.2: Métodos y algoritmos de Machine Learning más utilizados en 2020 (extraído de [2])

Para comprobar la eficacia de los algoritmos se va a usar la matriz de confusión, medidas F1-score, Area Under the Curve (AUC) y otras métricas de evaluación, que permiten la visualización de los resultados de clasificación de los algoritmos de aprendizaje supervisado [31].

Matriz de confusión: Figura 2.1. Muestra el número de aciertos que ha tenido el algoritmo mostrando cuatro cifras:

Veradero Positivo (VP) : El algoritmo predice 0 y realmente es 0

Falso Positivo (FP) : El algoritmo predice 0 pero realmente es 1

Falso Negativo (FN) : El algoritmo predice 1 pero realmente es 0

Veradero Negativo (VN) : El algoritmo predice 1 y realmente es 1

Precisión: Cuántos elementos de esta clase han sido correctamente clasificados, ecuación

2.1.

Exhaustividad: Cuántos elementos de esta clase hay reales entre el total, ecuación 2.2.

F1-score: Media armónica entre precisión y exhaustividad.

Exactitud: Cúantos elementos totales se han clasificado correctamente, ecuación 2.3.

$$precision = \frac{VP}{VP + FP} \tag{2.1}$$

$$recall = \frac{VP}{VP + FN} \tag{2.2}$$

$$accuracy = \frac{VP + VN}{VP + FN + FP + VN} \tag{2.3}$$

		Predicción		
		Positivo	Negativo	Total
Observación	Positivo	VP	FN	$VP + FN$
	Negativo	FP	VN	$FP + VN$
	Total	$VP + FP$	$FN + VN$	N

Tabla 2.1: Matriz de confusión, siendo N el total de elementos utilizados

2.3. Contexto

En esta sección se van a analizar las herramientas de partida.

2.3.1. Herramienta edX-MAS+

El sistema *edX-MAS+* [3, 13], basado en el sistema *edX-MAS* [14, 15] es una aplicación web en la que se puede importar una edición de un MOOC y generar modelos para predecir la obtención de certificado o el abandono. También cuenta con una sugerencia del mejor algoritmo e interfaz gráfica.

En *edX-MAS+* se utilizaban los siguientes algoritmos de predicción para certificado y abandono:

- Bayesian GLM
- Boosted Logistic Regression
- CART Decision Tree

- eXtreme Gradient Boosting
- k-Nearest Neighbors
- Naive Bayes
- Neural Network
- Random Forest
- Stochastic Gradient Boosting
- Support Vector Machine

Además, la herramienta se organiza en tres módulos principales con intención de separar la importación de datos, predicciones y visualización:

- Módulo de importación de cursos
- Módulo de generación de modelos
- Módulo de visualización y exportado de resultados

La aplicación está programada con las siguientes tecnologías:

- Python para extraer, limpiar y procesar los datos
- Se utiliza R para el análisis estadístico y la interfaz del usuario
- Base de datos relacional PostgreSQL y gestor pgAdmin

Y se ha probado para tres cursos y seis ediciones de MOOCs síncronos.

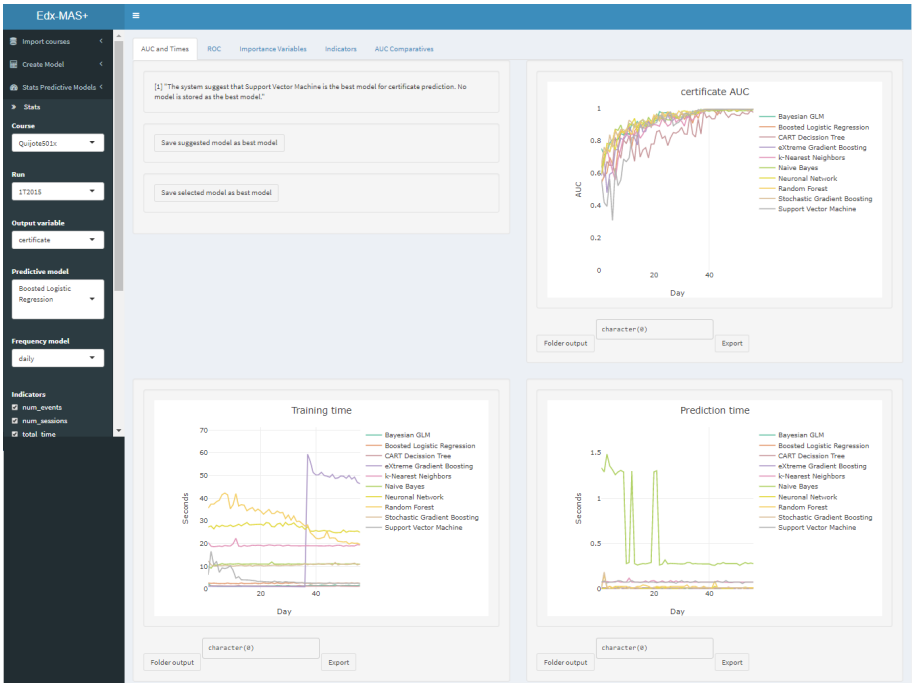

Figura 2.3: Interfaz gráfica de edX-MAS+ (extraído de [3])

El problema de esta herramienta es que ha quedado obsoleta, y a día de hoy no se puede acceder a ella desde la web. De aquí viene la principal motivación de crear el nuevo sistema informático que se propone en este TFG, con tencnología actual e integrándolo a *edX-LIMS*.

2.3.2. Herramienta edX-LIMS

La plataforma *edX-LIMS* [4, 16] es una aplicación web hecha excluisivamente para el curso de *WebApp* [32] que permite visualizar las estadísticas de los estudiantes inscritos en este curso, de tal manera que se puede ver el progreso de cada uno. Esta herramienta está basada en *edX-LIS* [12].

La interfaz gráfica de *edX-LIMS* cuenta con diferentes vistas, cuya organización se puede observar en la figura 2.4.

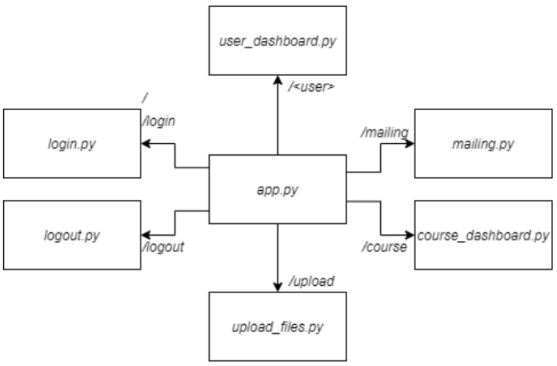

Figura 2.4: Estructura de navegación edX-LIMS (extraído de [4])

User dashboard: Muestra el dashboard personalizado para el usuario.

Login: Pantalla de inicio de sesión para el administrador

Upload files: Subservicio de extracción y preprocesado de datos, donde se introducen:

- **Logs del curso:** Ficheros en formato JSON con las acciones realizadas por los usuarios proporcionado por edX.
- **Cerfificaciones de los usuarios:** Notas de los estudiantes calculadas de manera semanal.
- **Perfiles de usuarios:** Datos de cada usuario para relacionar el curso con el usario.

Mailing: Módulo que para la redacción y envío de correos electrónicos que permite filtrar por indicadores de estudiante y guardar plantillas.

Course dashboard: Vista principal, donde se muestran las estadísticas generales de todos los usuarios, se puede acceder al dashboard personalizado de cada estudiante y las estadísticas de uso por parte de los usuarios del propio dashboard.

La tecnología utilizada para esta plataforma es:

- Python
- El framework Dash [33]

- Base de datos no relacional MongoDB

Además se cuenta con las colecciones mostradas en la siguiente lista en la base de datos.

app_users: Usuarios registrados en la application.

course_info: Información del curso.

course_structure: Información de la estructura del curso.

course_users : Información de los usuarios del curso verificados.

email_sendings: Registro de los correos enviados.

email_templates: Correos predefinidos del sistema.

final_indicators: Indicadores de los usuarios medidos para un día concreto.

num_problems_student_unit: Cúantos problemas de cada unidad ha hecho cada estudiante.

num_problems_unit: Número de problemas disponibles que tiene cada unidad.

problems_to_be_done: Problemas no resueltos por los estudiantes.

seen_problems: Registro de los resultados de los problemas hechos.

seen_videos: Registro de los vídeos y tiempo que ha visto un estudiante.

success_viability: Nota máxima que puede obtener un estudiante teniendo en cuenta los intentos realizados y resultados.

undone_attainable_grade: Valor máximo que puede obtener cada estudiante por cada subunidad del curso.

undone_attainable_grade_avgs: Lo mismo que el anterior pero por unidad.

user_filters: Filtros guardados por el administrador o profesor

user_grade_avgs: Nota semanal de cada usuario por unidad.

user_grades: Nota semanal de cada usuario por subunidad.

user_tracking: Registros de interacción del usuario con el dashboard

En la tabla *final_indicators* se guardan las interacciones de los usuarios con los recursos para cada día concreto, sin acumular datos. Estos son los siguientes indicadores para cada estudiante y día:

Number of Events: Número total de eventos registrados (un evento corresponde a un log)

Number of Sessions: Número de sesiones realizadas en el curso

Video time: Tiempo de uso de recursos de vídeo

Problem time: Tiempo de uso de recursos de problemas

Navigation time: Tiempo de uso de recursos de navegación

Forum time: Tiempo de uso de los recursos del foro

Total time: Tiempo total de uso

Forum Events: Número de eventos registrados de actividades de foro

Navigation Events: Número de eventos registrados de actividades de problema

Problem Events: Número de eventos registrados de actividades de problema

Video Events: Número de eventos registrados de actividades de vídeo

Consecutive Inactivity Days: Días consecutivos sin actividad en la plataforma edX

Connected Days: Días consecutivos de conexión en la plataforma edX

Different Videos: Vídeos diferentes vistos

Different Problems: Problemas diferentes realizados

El sistema cuenta con una interfaz gráfica hecha con Dash [33] (imagen 2.5), en la que el usuario puede navegar por las distintas vistas (mostradas en la imagen 2.4)

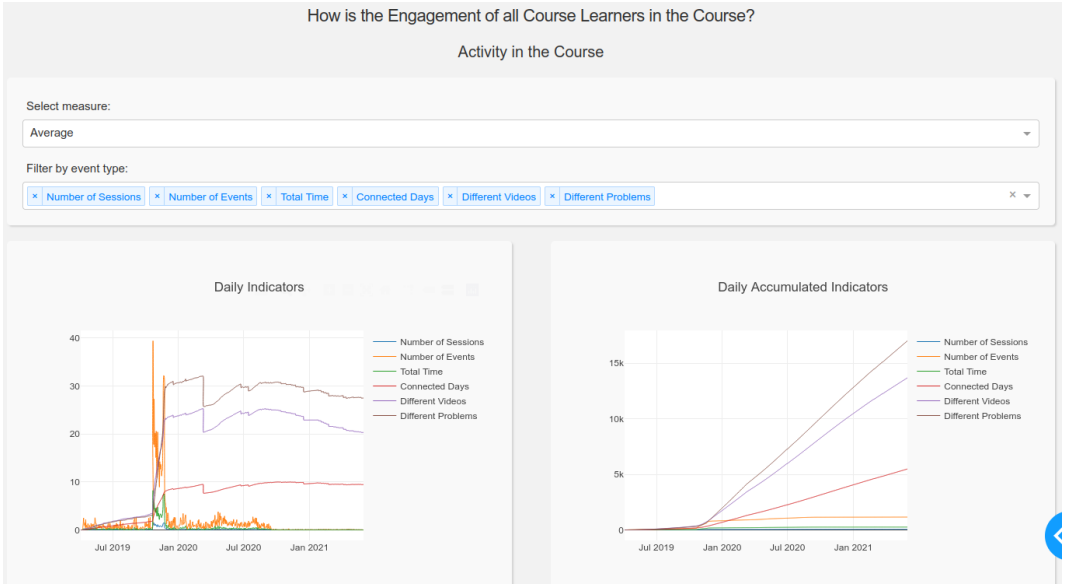

Figura 2.5: Interfaz gráfica de edX-LIMS (extraído de [4])

El sistema informático que se va a crear en este TFG contará con la misma tecnología que edX-LIMS y también funcionará exclusivameente para el curso de *WebApp*, puesto que se va a integrar a esta herramienta.

2.3.3. MOOC Introducción al desarrollo de aplicaciones web

El curso de *Introducción al desarrollo de aplicaciones web (WebbApp)* [32] es un MOOC disponible en la plataforma edX [10] desde el día 9 de abril de 2019. Desde esa fecha se han inscrito más de 37000 usuarios. Este curso es asíncrono, lo que indica que aunque está pensado para poder ser completado en cinco semanas, los usuarios pueden organizar su progreso en el tiempo que quieran.

En este curso, los estudiantes se pueden inscribir como *audit* o *verified*, solo en este segundo tipo de modalidad pueden participar en tareas calificadas. A día 1 de junio de 2021, este curso cuenta con un total de 1190 inscritos verificados, cuyos datos son los que se utilizarán para el desarrollo de este proyecto.

Los estudiantes disponen de los recursos de vídeos, problemas, páginas de explicaciones y foro de preguntas. La interacción de los estudiantes con estos recursos genera los indicadores de estos, mostrados en la sección 2.3.2.

Este curso se divide en cinco unidades y sus correspondientes subunidades:

- Unidad 1
  - 1.1 La World Wide Web
  - 1.2 Aplicaciones web
  - 2.1 HTML: para crear documentos Web
- Unidad 2
  - 2.2 Formularios: para crear la interfaz de comunicación con la aplicación Web
  - 2.3 CSS: para definir el aspecto de nuestros documentos Web
- Unidad 3
  - 3.1 Introducción al lenguaje de programación Python
  - 3.2 Implementación del servidor Web: Python + Flask
  - 3.3 Flask: respondiendo a peticiones del cliente
  - 3.4 Lenguaje de Templates
- Unidad 4
  - 4.1 El lenguaje JSON
  - 4.2 Gestión de sesiones
- Unidad 5
  - 5.1 JavaScript
  - 5.2 Document Object Model (DOM)
  - 5.3 Validación de formularios
  - 5.4 DHTML
  - 5.5 AJAX
  - 5.6 jQuery

Los estudiantes verificados, para conseguir su certificado necesitan obtener un 50 % de la nota ponderada de las cinco unidades, siendo un 0 el mínimo, un 1 el máximo y un 0.5 la obtención de certificado.

2.4. Tecnología

En esta sección se analizan las tecnologías utilizadas para el desarrollo del sistema de predicción.

Uno de los problemas de los MOOCs es la gran cantidad de datos que estos generan, por lo que su almacenamiento y acceso es una parte fundamental a la hora de tratar los datos.

Las **bases de datos relacionales** se popularizaron a finales de los 80 [34] y su idea es organizar la información en puntos de datos que se conectan entre sí a través de identificadores. Las bases de datos relacionales usan tablas para almacenar los datos, y cada fila es un registro con un identificador único, llamado *clave* [35]. En este tipo de bases de datos se utiliza el lenguaje Structured Query Language (SQL) para realizar las consultas. En *edX-MAS+* se utiliza SQL.

Las **bases de datos no relacionales** (también llamadas NoSQL) se diferencian en que estas no cuentan con un identificador que pueda conectar unos datos con otros. Los datos se organizan mediante documentos. El propósito de estas bases de datos es poder almacenar grandes cantidades de datos, pues las bases de datos relacionales no están preparadas para ello [36]. En *edX-LIMS* se utiliza MongoDB como base de datos no relacional.

El nuevo sistema va a utilizar la misma tecnología que *edX-LIMS* puesto que se va a integrar a este. Se van a dejar de lado las bases de datos relacionales, ya que se va a almacenar una gran cantidad de datos y se necesitará escalabilidad. El lenguaje R también se va a sustituir por Python, y se va a desarrollar el sistema basado en la herramienta *edX-MAS+* con la nueva tecnología mencionada y modificada para que funcione con el curso asíncrono de *WebApp*.

Dash [26, 33] es un framework de código abierto sobre el que está construido *edX-LIMS*, en el que se pueden mostrar dashboards y que además está integrado con Python. Se va a continuar con el uso de dashboards para la visualización de resultados en la interfaz gráfica. Además de Python y MongoDB, se han utilizado las siguientes librerías:

pandas: [37] Para el uso de *DataFrames* y operaciones con ellos. Estos son fundamentales para la organización y tratamiento de datos.

numpy: [38] Utilizado para las operaciones matemáticas necesarias.

scikit-learn: [39] Para utilizar los algoritmos y métricas de aprendizaje automático con Python. Se ha elegido por ser la librería de Data Science más utilizada por los programadores de Python [2].

dash extensions: [40] Necesario para añadir los nuevos componentes de descarga de archivos *csv* en la página.

pymongo: [41] Como nexo de unión entre Python y MongoDB.

dash boostrap components: [42] Utilizado para el diseño de la interfaz gráfica.

3

Predicción de abandono

Un gran problema al que se enfrentan los MOOCs es la continua falta de interés o abandono que sufren los estudiantes [43]. Algunos cursos tienen una tasa de abandono de más del 90 % [44]. Esto hace que predecir el abandono de los estudiantes sea una labor fundamental.

Para estudiar esta predicción del abandono, se utiliza principalmente la participación del estudiante. Estos datos se sacan de las plataformas de cursos online y es útil para predecir a tiempo real qué estudiantes van a abandonar [45].

A partir de este punto, hay que diferenciar entre MOOCs síncronos y MOOCs asíncronos.

En los **MOOCs síncronos**, el profesor es el que lleva el ritmo del curso y el que establece cuántos días hay para cada tarea o tema. Este tipo de cursos solo están disponibles en una ventana limitada de tiempo, aunque puedan ser impartidos en múltiples ocasiones.

Por el contrario, en los **MOOCs asíncronos**, son los estudiantes los que han de planificar su progreso y organizar sus avances, debido a que estos cursos no tienen una fecha de finalización y están disponibles durante un tiempo ilimitado. Esta modalidad hace que tengan que ser más responsables y disciplinados, pues el control de avanazar en el curso depende únicamente de ellos.

En estos, la diferencia entre los que abandonan el curso y los que lo terminan, es que los segundos poseen aptitudes de Self-Regulated Learning (SRL) y son capaces de fijar objetivos, es decir, que son más propensos a guiarse en el aprendizaje [46].

Se ha demostrado que las habilidades SRL tienen una gran influencia a la hora de predecir el abandono [43]. Por otro lado, los estudiantes que perciben los resultados como consecuencia de su esfuerzo, son más propensos a continuar el curso que aquellos que achacan los resultados a la suerte o a “una buena racha” [47].

Al ser dos modelos diametralmente opuestos, el abandono de los cursos se predice de diferente manera. Para este, que es asíncrono, se va a optar por seguir el método detallado por P. M. Moreno [43, 48].

3.1. Estudio previo

Antes de calcular el abandono se ha decidido hacer un estudio previo del estado de los estudiantes del curso (un total de 1190).

El curso de WebApp [32] es un MOOC asíncrono, por lo que para calcular el abandono hay que seguir un procedimiento diferente al de los cursos síncronos.

El problema de un MOOC asíncrono es que un estudiante puede estar inactivo durante un periodo extenso y luego volver a conectarse [43, 48]. Esto implica que el primer paso es encontrar el periodo máximo de tiempo en el que un estudiante está inactivo pero sin dejar el curso. A este intervalo le llamaremos *Máximo periodo inactivo*.

Para encontrar este intervalo hay que tener en cuenta que el máximo tiempo que puede llevar un estudiante inscrito en este curso son 790 días (pues la extracción de datos es a día 1 de junio de 2021). Por lo que se han seleccionado aquellos que llevan por lo menos un 70 % del tiempo inscritos, lo que haría un total de 629 estudiantes que llevan en el curso un mínimo de 553 días. Además se han descartado 13 estudiantes por no haberse conectado nunca y por lo tanto no tener un máximo periodo inactivo.

Se ha escogido esta cifra teniendo en cuenta un término medio entre elegir un número suficiente de estudiantes y un número de días no demasiado elevado.

En la tabla 3.1 se puede observar que, por ejemplo, 683 estudiantes llevan inscritos en el curso el 60 % del que este lleva existiendo, y tan solo 565 llevan desde el principio.

Porcentaje de curso	0	50	60	70	80	90	100
Número estudiantes	1190	850	683	629	570	568	565

Tabla 3.1: Número de estudiantes aue llevan cada porcentaje de tiempo en el curso

Tras introducir estos datos en el algoritmo creado para encontrar el máximo periodo inactivo, se obtienen los resultados mostrados en la tabla 3.2

Percentil	10	25	50	70	80	90	95	98	100
Máximo periodo inactivo	62	197	198	201	205	223	269	339	553

Tabla 3.2: Cada percentil corresponde con el máximo periodo inactivo

Con lo que se puede observar que, por ejemplo, si observamos el percentil 25, indica que el 75 % de los estudiantes han estado más de 197 días desconectados pero luego se han vuelto a conectar.

No obstante, también hay que tener en cuenta que a pesar de que un estudiante se vuelva a conectar tras un largo periodo de inactividad, puede haber decidido abandonar ya el curso, por lo que hay que elegir un tramo en el que los estudiantes ya se conecten con muy poca frecuencia. Después

de esto, hay que tener en cuenta que aunque un usuario haya estado más días desconectado, si este ha obtenido ya un certificado o su nota es igual o superior a 0.5 (aprobado), concluiremos que ha superado el curso y no lo ha abandonado.

A modo de consecuencia, podríamos elegir 223 días inactivos como el límite de un usuario para volver a conectarse, puesto que el 90 % de los usuarios se vuelven a conectar aunque estén inactivos este periodo de tiempo. Por esto, se van a seguir estas dos reglas:

Obtención de certificado: Hay que tener en cuenta que aunque un usuario haya estado más días desconectado, si este ha obtenido ya un certificado o su nota es igual o superior a 0.5 (aprobado), concluiremos que ha superado el curso y no lo ha abandonado.

Usuarios matriculados recientemente: Por otro lado tiene sentido descartar a aquellos usuarios que lleven poco tiempo matriculados o avancen despacio, por lo que se descarta a los usuarios cuyo tiempo inactivo sea inferior a 223 días.

El resultado de estas normas implica el diagrama de decisiones mostrado en la figura 3.1

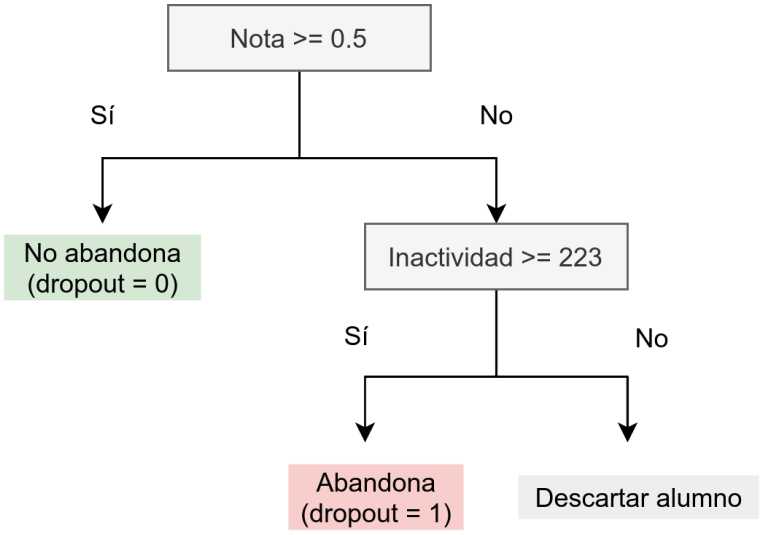

Figura 3.1: Diagrama de decisión para el abandono en WebApp

Con este algoritmo, a cada estudiante del curso se le ha asignado un valor de abandono dependiendo de si se considera que ha dejado el curso (abandono = 1) o no (abandono = 0), o se descarta. Teniendo en cuenta las apreciaciones de la figura 3.1, se contemplan finalmente 901 estudiantes sin descartar.

3.2. Abandono en tiempo real

Una vez implementado el algoritmo, se aplica a todos los estudiantes del curso para su último día en el curso a fecha de 1 de junio de 2021 (puesto que muchos estudiantes después continuarán inscritos

en el curso).

Para cada estudiante se guarda su valor de abandono en la colección *final_dropout*, donde el valor 0 implica que continúa el curso o que ya ha obtenido su certificado (aprobado), 1 implicaría abandono y -1 descarte.

Estos datos corresponderían al abandono para el último día que se tienen datos, aunque el verdadero reto sería poder saber si el usuario abandona a tiempo real. De esta manera se podría realizar una intervención al estudiante y animarle a continuar con su nivel de implicación [45].

Para ello se ha decidido trabajar de la misma forma que se hace con la obtención del certificado de aprobado, tal y como se muestra en la sección 4.3. Con los algoritmos seleccionados se podrá hacer una predicción del abandono para cada día del curso con los indicadores del estudiante y su nota actual. Los resultados de la eficacia de este método se explican en la sección 5.2.

4

Desarrollo

En el siguiente capítulo se detalla la implementación, explicando las fases que se han seguido.

4.1. Descripción

El objetivo del desarrollo ha sido crear un sistema informático de predicción de certificado y abandono, para poder saber si un estudiante va a aprobar el curso o abandonarlo.

La implementación se ha desarrollado en tres fases:

1.– **Organización y tratamiento de datos:** Se realiza el estudio de las colecciones recibidas en la base de datos (no SQL) y se crean nuevas que cumplan lo necesario para poder realizar el paso siguiente.

2.– **Entrenamiento de los modelos y predicción:** Se utilizan los datos tratados para entrenar los cinco algoritmos descritos en la sección 4.3 y almacenar los resultados obtenidos.

3.– **Visualización de los resultados e interacción con la herramenta en Dash:** Se permite al usuario entrenar los modelos eligiendo las opciones necesarias y ver las respuestas generadas.

Para ello se establecen dos variables a predecir:

Certificado: Un estudiante obtiene el certificado del curso si su nota es mayor o igual a 0.5 (siendo 1 el máximo y 0 el mínimo).

Abandono: Se considera que un estudiante ha abandonado el MOOC cuando cumple con los requerimientos establecidos en el capítulo 3

Este módulo se ha agregado a la plataforma web *edX-LIMS*. Para ello se ha procesado el contenido de la base de datos de *WebApp* [32], procediendo a la creación de nuevas tablas y una nueva vista para la interfaz gráfica.

El código se ha estructurado en un único archivo: *predictions.py*

También se ha creado otro archivo en la carpeta de vistas con las gráficas y todo lo necesario para poder ejecutar las funciones desde la interfaz.

4.2. Organización y tratamiento de datos

La primera parte para la creación del módulo ha sido crear las colecciones necesarias. Para ello se han obtenido las colecciones de *edX-LIMS* [4, 16], mostradas en la sección 2.3.2 necesarias para ello:

course_users: Contiene los datos de los usuarios, incluyendo su nota final y si obtiene o no certificado.

final_indicators Colección con los indicadores de cada usuario para cada día.

user_grade_avgs: Tabla con las notas semanales de cada estudiante.

A partir de estas colecciones recibidas, el objetivo era poder agrupar los datos de tal manera que para cada usuario (de manera anonimizada siempre), y para cada día del estudiante en el curso, se tengan los indicadores para esa fecha, la nota para ese día y el valor asignado de abandono.

Es importante señalar que como cada usuario puede haber empezado el curso en un día diferente, para una fecha concreta cada usuario lleva un progreso diferente. Por eso, a la hora de hacer las predicciones tiene sentido agrupar los indicadores empezando a contar desde el primer día para cada estudiante. Para ello se añade una variable en la nueva colección de *indicators_and_data_acc* de cuántos días lleva un usuario inscrito, *number_day*.

Hay que tener en cuenta que se tiene la nota semanal de cada estudiante, por lo que al resto de días de la semana se le asigna la nota de la semana anterior o cero en el caso de la primera semana.

Para organizar los datos de manera adecuada se han creado las siguientes colecciones:

Nombre	**Campos**
final_dropout	Id usuario, Abandono
user_first_day	Id usuario, Primer día
indicators_and_data_acc	Id del curso, Id usuario, Primer día, Día actual, Número de día, Estado de certificado, Nota actual, Aprobado, Abandono, Número de eventos, Número de sesiones, Tiempo de vídeo, Tiempo de problemas, Tiempo de navegación Tiempo de foro, Tiempo total, Eventos del foro, Eventos de navegación, Eventos de problemas, Eventos de vídeo, Dias consecutivos de inactividad, Días conectado, Vídeos distintos vistos, Problemas distintos vistos

Tabla 4.1: Colecciones añadidas para el tratamiento de datos

Donde cada tabla tiene su función:

final_dropout: Para cada usuario se le asigna un valor de abandono teniendo en cuenta sus días consecutivos de inactividad y su nota para el día de la recolección de datos. (Véase cápitulo 3)

user_first_day: Se obtiene para cada usuario el día en el que empieza el curso, puesto que no tiene por qué ser el mismo día en el que se unió a edX (fecha registrada en la colección de *course_users*.

indicators_and_data_acc: Contiene los indicadores extraídos de *edX-LIMS* y mostrados en la sección 2.3.2. Es decir hay una entrada por cada día de cada usuario (un total de 639944). Presentando en cada una el estado de los indicadores del estudiante de manera acumulativa.

La creación de las dos primeras tablas, aunque incluidas en la tercera, es útil para ahorrar tiempo de acceso, pues la colección *indicators_and_data_acc* es muy pesada y cargarla desde la base de datos es muy costoso.

4.3. Predicciones

En esta sección se van a explicar los algoritmos que se han decidido utilizar para las predicciones, las métricas para valorar su eficacia y el procedimiento seguido para la mejora de resultados.

4.3.1. Algoritmos

Se han elegido los siguientes algoritmos por ser los mejores para la predicción de certificado y abandono [3,14], y más usados [2]. Todos los algoritmos empleados han sido obtenidos de la biblioteca Scikit-learn [39], y los procedimientos seguidos se detallan en [49,50]:

Generalized Linear Model: Modelo lineal para aproximar una variable dependiente con variables independientes y un término aleatorio. Sirve para predecir la nota exacta, por lo que para comprobar su eficacia se tiene en cuenta si la nota predicha es un aprobado o un suspenso. Se obtiene del paquete *linear_model.LinearRegression* sacado de [50] .

Neural Network: Las redes neuronales usan algoritmos de aprendizaje que siguen modelos matemáticos inspirados en el cerebro humano [51], en el entorno de los MOOCs han recibido especial peso desde 2017 [52], en Scikit-learn se obtiene en el paquete *neural_network.MLPClassifier*.

Support Vector Machine: Son un conglomerado de algoritmos matemáticos de aprendizaje supervisado para maximizar una función con respecto a una colección de datos. Dadas las variables de entrada, separa las clases buscando el hiperplano que maximice la distancia

entre el hiperplano y los vectores de soporte [53]. El algoritmo empleado se encuentra en el paquete de Scikit-learn *svm.SVC*.

Random Forest: Conjunto de técnicas de aprendizaje de clasificación y regresión en el que se combinan un grupo de árboles de decisión utilizando *bagging*, esta técnica hace que se reduzca el error, consiguiendo mejores resultados que con un solo árbol de decisión [54]. Se utiliza el paquete *ensemble.RandomForestClassifier* de Scikit-learn.

Stochastic Gradient Boosting: Algoritmo que combina de forma escalonada varios modelos de predicción débiles utilizando *boosting* [28] en Scikit-Learn *ensemble.GradientBoostingClassifier* [55]

4.3.2. Procedimiento

A la hora de entrenar los modelos se han utilizado los datos de la colección *indicators_and_data_acc*, descartando las columnas que contienen IDs, nota, abandono y certificado. Salvo para el abandono, en cuyo caso sí que se conserva el certificado.

Después hay que normalizar los indicadores que se van a utilizar.

Cada modelo se puede entrenar con una cantidad diferente de datos, por ello, hasta encontrar la manera definitiva se probaron distintas opciones:

La primera idea fue entrenar cada modelo con los indicadores de cada día del curso (los mostrados en la subsección 2.3.2) para todos los usuarios, es decir se agrupan los indicadores del primer día para cada usuario y se entrena el modelo, y así sucesivamente con cada día. El problema de hacer esto es que al ser este curso un MOOC asíncrono puede haber muchos días de inactividad seguidos, lo que ”ensuciaría” los datos por estar vacíos, dando lugar a resultados sin sentido.

Lo mismo pasa si además se usan solo los indicadores del día actual, un día en el que un usuario no se ha conectado, todos sus indicadores están a cero, pero sin embargo su nota puede no ser cero, lo que también lleva a incongrunencias en los resultados.

La solución final arregla los dos problemas anteriores, primero se hace acumulativa la tabla de indicadores, por lo que los progresos de los usuarios se suman a los que ya llevan hechos anteriormente. Después también se ha optado por un modelo acumulativo en el entrenamiento, en el que se entrena cada modelo con los indicadores para ese día y todos los anteriores. Esto último hace que cada día que se entrena, tarde más que el anterior, haciendo que tanto el tiempo de entrenamiento como el de predicción sigan una función cuadrática.

Por ello, se siguieron tres fases de creación y mejora:

- Indicadores no acumulativos y entrenamiento no acumulativo por día
- Indicadores acumulativos y entrenamiento no acumulativo por día

- Indicadores acumulativos y entrenamiento acumulativo con los días anteriores

La última opción es la mejor, puesto que al añadir más datos, las predicciones son más precisas [48].

Con los datos que recibe cada modelo se ha optado por usar el 75 % para el entrenamiento y el 25 % para test. De esta manera se dividen en cuatro conjuntos:

X_train: 75 % de los indicadores para el entrenamiento.

X_test: 25 % de los indicadores para el test.

y_train: Los datos correspondientes a la columna a predecir para el entrenamiento.

y_test: Los datos correspondientes a la columna a predecir para el test.

Una vez obtenidos estos conjuntos, se entrena el modelo con *X_train* e *y_train*, y con el modelo ya entrenado se hacen las predicciones con *X_test*. Además se miden los tiempos de entrenamiento y predicción.

También se ha obtenido la importancia de las variables para los algoritmos *Random Forest* y *Stochastic Gradient Boosting*, utilizando la función *feature_importances_* (solo disponible para estos dos algoritmos), que devuelve el porcentaje de importancia de variable para cada indicador.

4.3.3. Métricas

Para ver la eficacia de los algortimos se han usado las métricas AUC, F1 y la matriz de confusión. Hay que tener en cuenta que para comprobar la operatividad de los algoritmos usaremos los conjuntos de datos *y_train*, *y_test*, *prediction_train* y *prediction_test*, donde estos dos últimos son los resultados de la predicción para el entrenamiento y para el test respectivamente, mientras que en *y_train* e *y_test* se encuentran los resultados reales.

F1

F1-score es una métrica de evaluación que consiste en hacer la media armónica entre la precisión y exhaustividad. Se ha utilizado el paquete *metrics.f1_score* de Scikit-learn [39]. Esta medida se va a utilizar para ser visualizada en la interfaz gráfica para la predicción de certificado y abandono.

AUC

La métrica AUC muestra el área bajo la curva Receiver Operating Characteristic (ROC) , que relaciona las tasas de verdaderos y falsos positivos. Indica la probabilidad de que el modelo clasifique un ejemplo positivo aleatorio más alto que un ejemplo negativo aleatorio [56]. Esta métrica se va

a utilizar para la visualización de resultados en la interfaz gráfica, y se va a utilizar el paquete *metrics.roc_auc_curve* de Scikit-learn [39]

Matriz de confusión e informe de clasificación

Para estudiar la eficacia de los algoritmos, en un primer momento se han usado las funciones de Scikit-learn [39] *confusion_matrix* y *classification_report*

Los valores obtenidos son la matriz de confusión (explicada en la sección 2.2), precisión (ecuación 2.1), exhaustividad (ecuación 2.2), exactitud (ecuación 2.3) y F1-score.

```
Stochastic Gradient Boosting
[[57692   220]
 [  193  3096]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     57912
           1       0.93      0.94      0.94      3289

    accuracy                           0.99     61201
   macro avg       0.97      0.97      0.97     61201
weighted avg       0.99      0.99      0.99     61201

[[19209    91]
 [   71  1030]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     19300
           1       0.92      0.94      0.93      1101

    accuracy                           0.99     20401
   macro avg       0.96      0.97      0.96     20401
weighted avg       0.99      0.99      0.99     20401
```

Figura 4.1: Matriz de confusión e informe de clasificación para el día 70 del curso con el algoritmo *Stochastic Gradient Boosting*

Como se puede ver en la imagen 4.1, se ha impreso la matriz de confusión y el informe de clasificación para las predicciones tanto con el propio entrenamiento como con el test. Para ello se han usado los indicadores de los estudiantes hasta el día 70.

Esto se ha utilizado a modo de prueba con todos los algoritmos, para perfeccionar el entrenamiento, aunque no se muestra en la interfaz gráfica.

4.4. Almacenamiento

Los resultados obtenidos se han guardado en la base de datos de MongoDB, donde más adelante van a ser utilizados para la interfaz gráfica.

La primera idea de diseño para el almacenamiento fue hacer una tabla en la base de datos por

cada gráfica que se quería mostrar (como por ejemplo la AUC score). Sin embargo esta maqueta se apartó porque obligaría al usuario a entrenar todos los algoritmos a la vez el mismo número de días.

La solución fue crear una tabla por cada algoritmo y objetivo de la columna a predecir. Este diseño, aunque implique la creacion de más colecciones, dota al sistema de disociación, lo que permitirá entrenar algoritmos de forma conjunta o individual.

Para las predicciones se han creado las colecciones resultantes:

- Linear Model Certificate
- Linear Model Dropout
- Random Forest Certificate
- Random Forest Dropout
- Stochastic Gradient Boosting Certificate
- Stochastic Gradient Boosting Dropout
- Support Vector Machine Certificate
- Support Vector Machine Dropout
- Neural Network Certificate
- Neural Network Dropout

Y cada una de ellas cuenta con los mismos tipos de datos:

- Número de día
- Valor AUC-score
- Valor F1-score
- Tiempo de entrenaimento
- Tiempo de predicción

Además, para los algoritmos de *Random Forest* y *Stochastic Gradient Boosting* se han creado dos tablas para cada uno con la importancia de las variables para el certificado y abandono. Estas colecciones guardan para cada día y cada indicador utilizado en el entrenamiento, la importancia de esa variable para entrenar el modelo.

Para utilizar los datos de estas tablas se han añadido nuevas funciones al fichero *edxmongostore*. Esta clase sirve como acceso a MongoDB gracias a la librería PyMongo [41].

4.5. Visualización

Para poder utilizar el nuevo sistema informático creado, se le ha añadido una nueva ruta a la aplicación *edX-LIMS*. Se puede acceder desde la ventana principal, *course_dashboard* mediante un botón.

En la nueva vista, el usuario puede escoger los algoritmos que desee entrenar entre los cinco posibles, seleccionar la variable a predecir (o las dos), cuántos días clasificar, o actualizar los datos

previos. Se pueden actualizar varios algoritmos a la vez aunque cada uno de ellos lleve diferentes días entrenados.

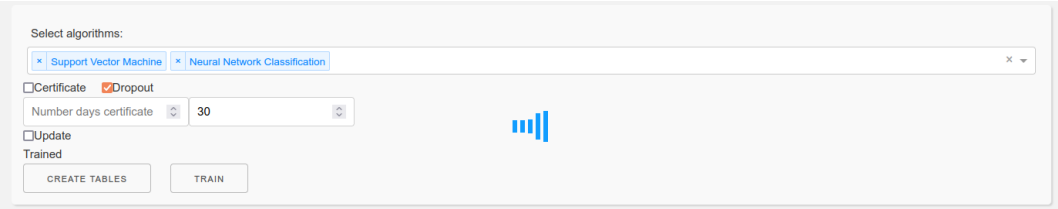

Figura 4.2: Selección de algoritmos por parte del usuario para entrenar los modelos

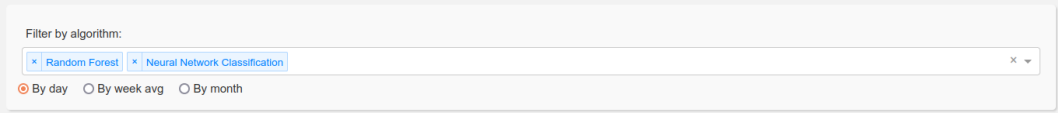

Figura 4.3: Selección de algoritmos por parte del usuario para visualizar los resultados

Se han incluído un total de doce gráficos que muestran los diferentes resultados obtenidos y una tabla que muestra los datos contenidos en *indicators_and_data_acc* (explicada en la sección 4.2).

La página cuenta con una cabezera similar a la del resto de vistas, mostrando el nombre del curso y del módulo (*Predicciones* en este caso). La vista se puede dividir en distintas secciones:

Creación de tablas y entrenamiento: En esta parte el usuario puede entrenar los modelos como quiera. Se muestra en la figura 4.2.

Selección de algoritmos: Figura 4.3. El usuario puede seleccionar los algoritmos previamente entrenados y elegir el periodo de tiempo del que quiere la muestra (diario, semanal o mensual). Esta selección afecta a las dos variables a predecir. Para cada una de estas dos variables (certificado y abandono) se muestran seis gráficas (Ejemplo figura 4.4):

Métrica AUC: Valor de la métrica AUC para cada día del algoritmo seleccionado.

Métrica F1: Valor de la métrica F1 para cada día del algoritmo seleccionado.

Tiempo de entrenamiento: Tiempo de entrenamiento para cada día del algoritmo.

Tiempo de prediccción: Tiempo de predicción para cada día del algortimo.

Importancia de las variables: Se puede seleccionar los indicadores que quiere ver el usuario. Estos afectarán a las dos gráficas:

- Importancia Variables Stochastic Gradient Boosting
- Importancia Variables Random Forest

Tabla de datos: Muestra la información guardada en *indicators_and_data_acc*. Se puede filtrar y guardar los filtros. Figura 4.5

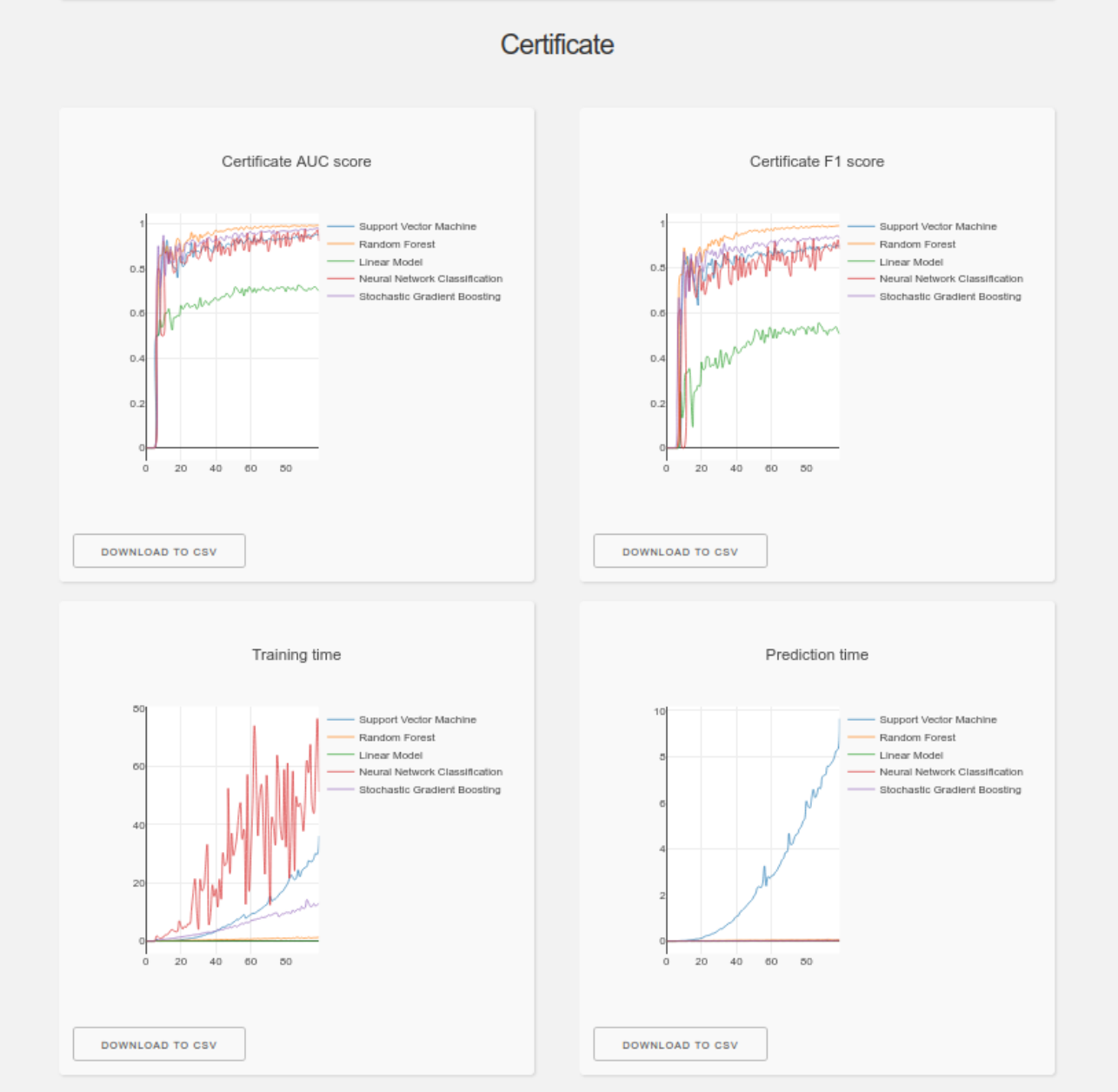

Figura 4.4: Ejemplo de vista del módulo *Predicciones*, parte de Certificado

También se ha puesto la opción de poder descargar los datos de las gráficas y las tablas nuevas de la base de datos mediante botones. El archivo se descargará en *csv*.

Se ha añadido la opción de poder cambiar de idioma de la aplicación, a Español o Inglés. Además se ha hecho fácilmente escalable para que se puedan incorporar otros idiomas de cara al futuro.

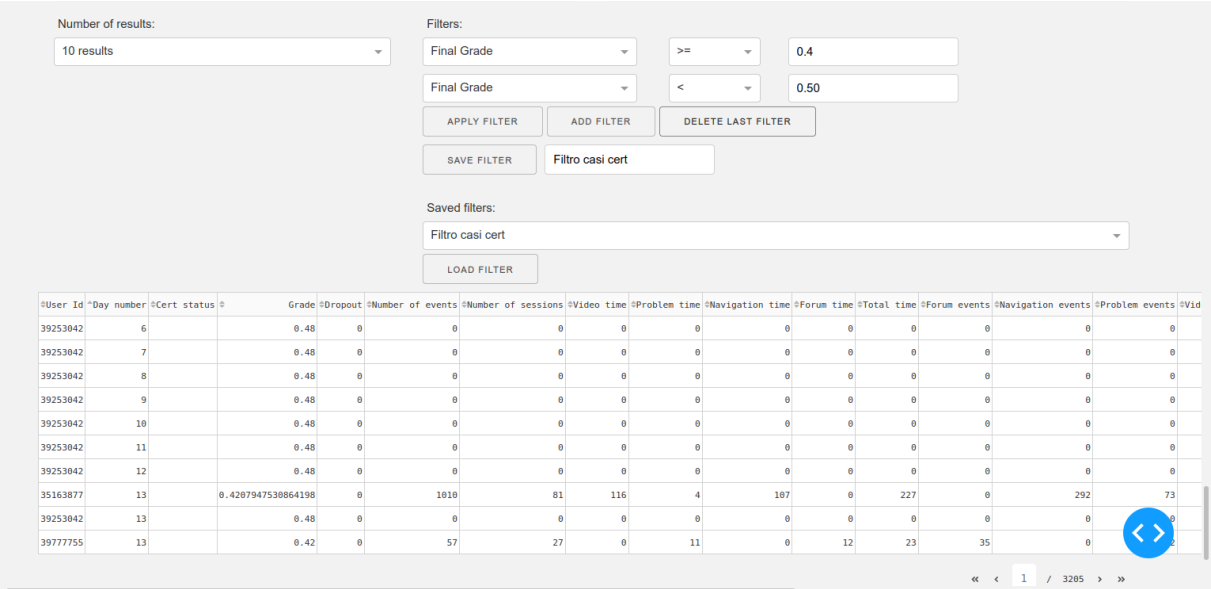

Figura 4.5: Muestra de la tabla de indicadores y notas con filtro

4.6. Pruebas

Para la realización de pruebas del sistema informático se han utilizado los datos reales de la base de datos, del curso *WebApp* [32].

4.6.1. Pruebas unitarias

Para probar el correcto funcionamiento de las funciones creadas de manera individual, se han realizado las siguientes pruebas:

Organización y creación de las nuevas estructuras: Se ha comprobado que se obtienen correctamente los datos de los usuarios de la base de datos, que se puede operar con ellos y que los resultados obtenidos tienen sentido lógico.

Predicciones: Comprobar que los datos con los que se va a trabajar son completos y no causará ningún error que alguno de ellos esté corrupto.

Visualizaciones: Ver cómo se muestran las gráficas, comportamiento de la herramienta ante la introducción de datos sin sentido por parte del usuario. Muestra correcta de los datos

introducidos en las gráficas tanto el eje X como el Y.

Las funciones usadas para estas pruebas se encuentran en el archivo *predictions.py*, en el apartado de pruebas. Se han ejecutado desde el main y comprobado que todas las salidas son correctas.

Estas pruebas se han realizado en un primer momento con pequeñas muestras de datos y más adelante de manera masiva.

4.6.2. Pruebas de integración

Este tipo de pruebas se realizan para comprobar la cohesión de las diferentes partes y su correcto funcionamiento en conjunto.

- Se comprueban los datos generados por los modelos en las diferentes gráficas creadas.
- Utilización de todos los elementos de la interfaz gráfica para mostrar su correcto funcionamiento

4.6.3. Pruebas del sistema

El entorno de desarrollo en el que se ha programado el módulo ha sido un ordenador portátil con Ubuntu 18.04.5 LTS, 12GB de RAM y Core i7-6500U.

El entrorno de producción en el que se ha desplegado es un ordenador de sobremesa con Linux Mint 19.2 Cinnamon, 16GB de RAM y Core i7-4771.

4.6.4. Técnica de Thinking Aloud

La técnica de Thinking Aloud o Pensar en Voz Alta sirve para evaluar la usabilidad del sistema. Esta prueba consiste en pedirle a un usuario que siga una lista de tareas a realizar sobre el sistema mientras dice en voz alta todo lo que opina. Estas opiniones se recogen para poder evaluar los problemas de usabilidad de la aplicación.

Las tareas que se van a realizar se encuentran detalladas en el anexo A.

5

Resultados

Este capítulo presenta los resultados obtenidos con la utilización del sistema informático en el MOOC asíncrono de *WebApp* [32]. Se han creado modelos diarios con los cinco algoritmos tanto para certificado como para abandono.

5.1. Certificado

Para la predicción de certificado se han entrenado todos los algoritmos 100 días. Se ha escogido este valor porque se puede observar que a partir de aproximadamente la tercera semana de trabajo del estudiante, los valores no varían.

En la visualización de los resultados (imagen 5.1) se puede observar que:

- Los mejores algoritmos para la predicción de certificado son *Stochastic Gradient Boosting* y *Random Forest*, con un valor máximo de AUC de 0.99 y 0.97 y F1-score de 0.98 y 0.94 respectivamente.
- El peor algoritmo es *Generalized Linear Model*, con un máximo de 0.72 de valor de AUC y 0.55 de F1-score.
- A partir del día 25 aproximadamente, ya se tienen valores superiores a 0.9 en AUC.

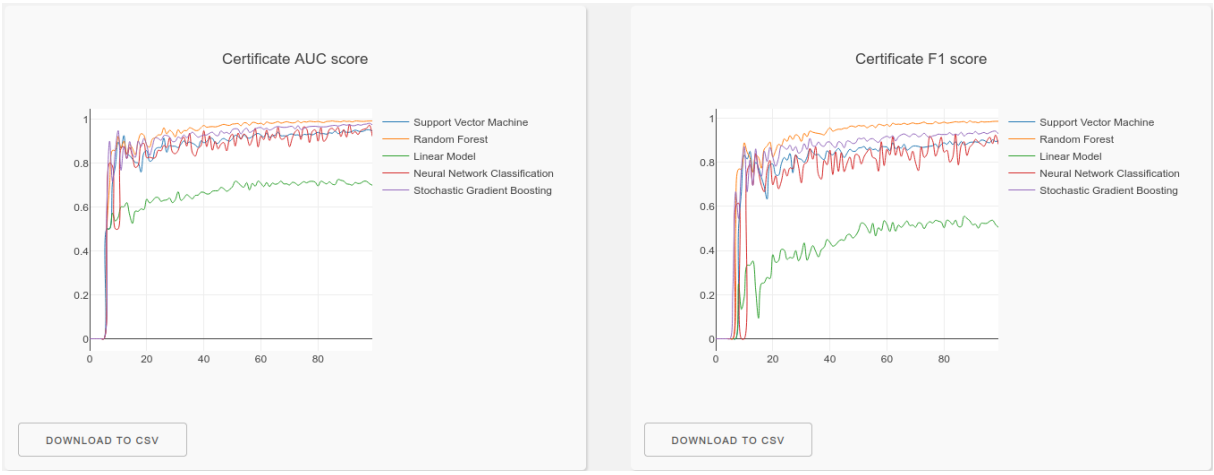

Figura 5.1: Resultados de predicción para el certificado

5.2. Abandono

En las predicciones de abandono se han entrenado 100 días los algoritmos de *Support Vector Machine* y *Neural Network*, 300 días *Stochastic Gradient Boosting* y 400 días *Random Forest* y *Linear Model*. Esta variación en los días entrenados se debe al tiempo que tarda en entrenarse cada algoritmo, entrenando menos tiempo los algoritmos más lentos (desarrollado en la sección 5.3).

Se sacan las siguientes conclusiones de la figura 5.2:

- La mética F1-score es muy inestable hasta el día 150.
- Se puede notar una subida en la gráfica a partir de aproximadamente el día 220. Esto se justifica con lo explicado en la sección 3.1, puesto que una gran parte de los estudiantes están inactivos ese periodo de tiempo y luego se vuelven a conectar.
- Los mejores algortimos vuelven a ser *Random Forest* y *Stochastic Gradient Boosting* para la métrica AUC y *Linear Model* para la métrica F1-score, aunque habría que hacer un análisis más profundo con más días de entrenamiento.

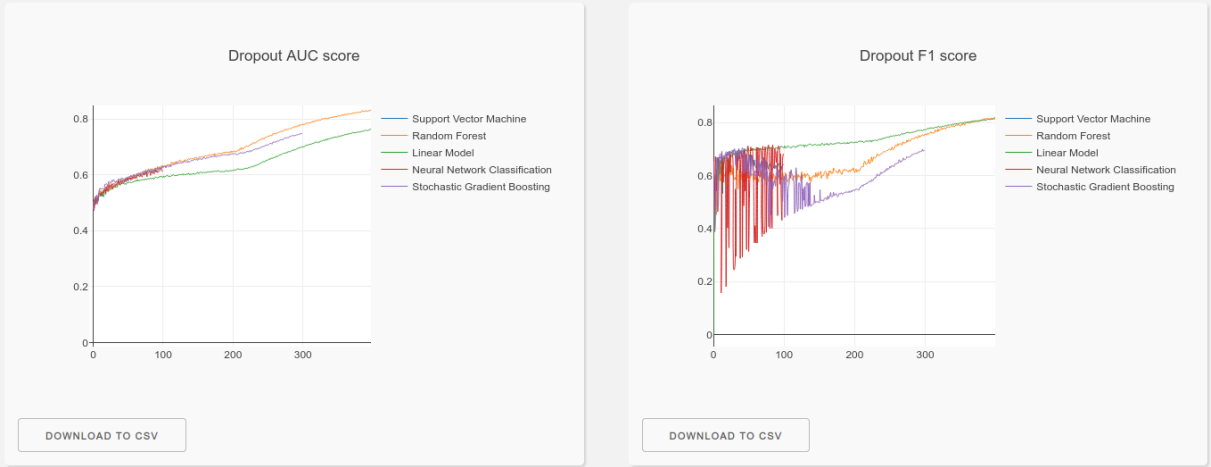

Figura 5.2: Resultados de predicción para el abandono

5.3. Tiempos de entrenamiento y predicción

Para cada algoritmo se ha medido el tiempo que tarda en entrenar el modelo y en hacer la predicción. Esto es interesante tenerlo en cuenta para saber qué modelos entrenar en el futuro.

En las figuras 5.3 y 5.4 se pueden ver los tiempos en seguntos (eje Y) para cada día (eje X). En ellas se puede observar lo siguiente:

- El tiempo de predicción es mucho menor que el de entrenamiento para todos los algoritmos tanto para el certificado como abandono.
- *Neural Network* y *Support Vector Machine* son los algoritmos más lentos.
- *Linear Model* es el más rápido, sin llegar a alcanzar 0.5 segundos.

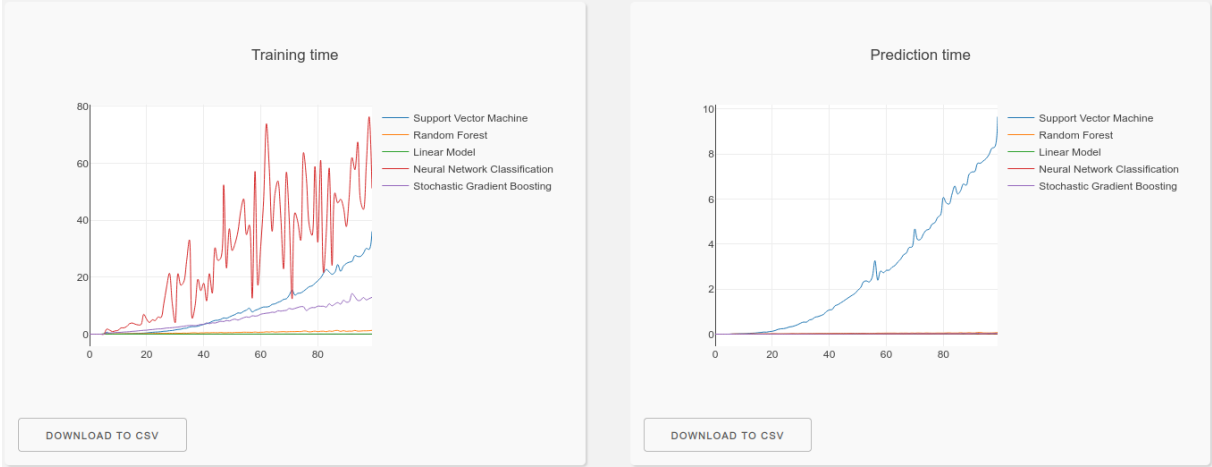

Figura 5.3: Tiempo de entrenamiento para el certificado

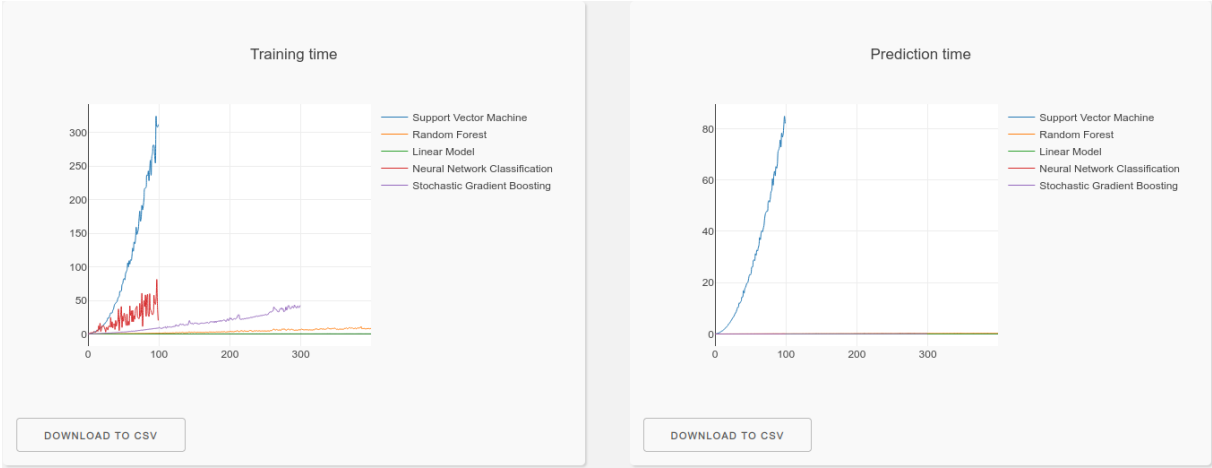

Figura 5.4: Tiempos de entrenamiento y predicción para el abandono

5.4. Importancia de las variables

En cuanto a la importancia de las variables, hay que tener en cuenta que para la predicción del abandono se tiene una variable más, el certificado.

En las imágenes 5.5 y 5.6 se muestran los resultados:

- Para el certificado el indicador más importante es *Different Problems*, destacando en *Stochastic Gradient Boosting* a partir del día 35.
- Al principio del curso es importante *Consecutive Inactivity Days*
- Las variables menos importantes para ambos objetivos son las relacionadas con el Foro, como *Forum Time*. Las relacionadas con el tiempo en general no son importantes.
- Para el abandono el indicador más importante es el certificado, aumentando a partir del día 220 como en la predicción.

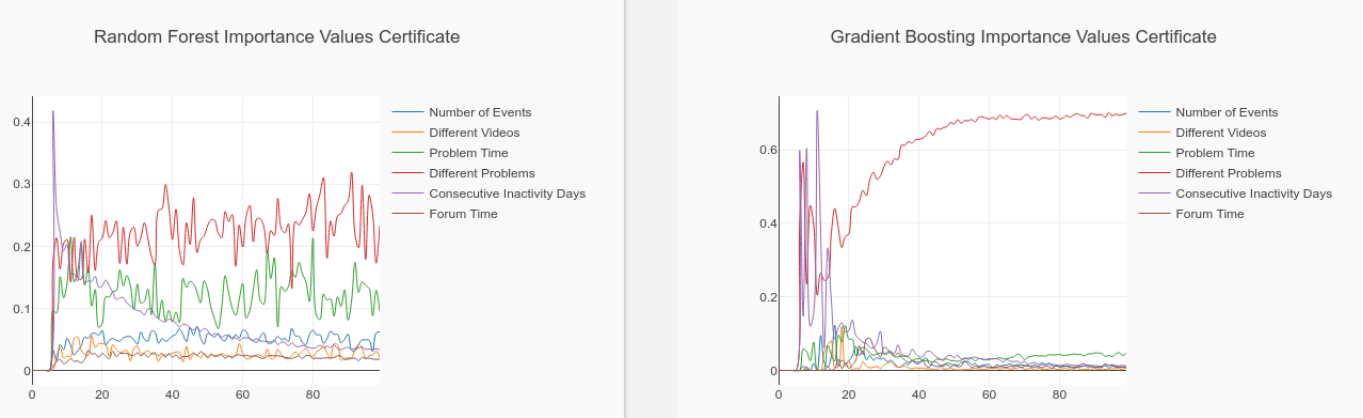

Figura 5.5: Importancia de las variables para los algoritmos de *Stochastic Gradient Boosting* y *Random Forest* (Certificado)

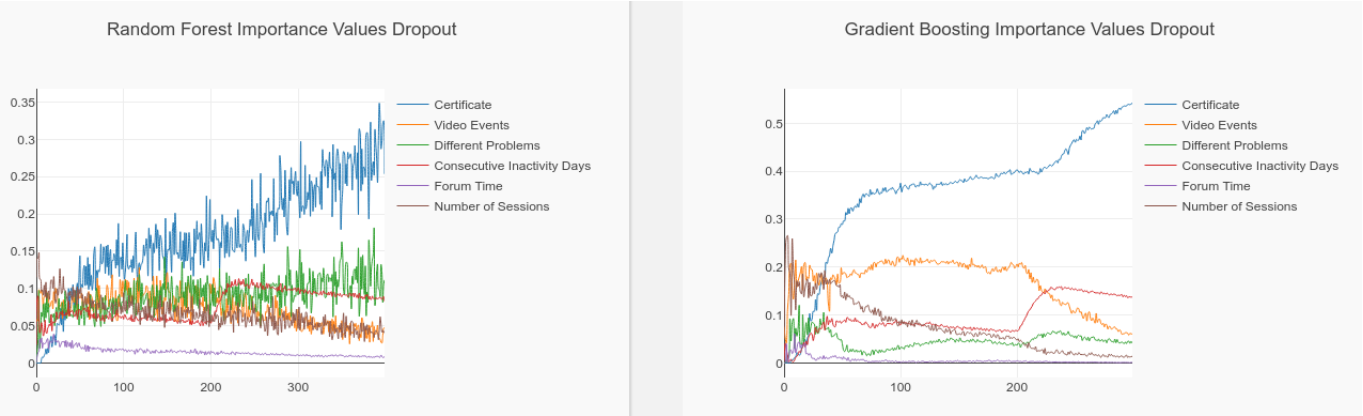

Figura 5.6: Importancia de las variables para los algoritmos de *Stochastic Gradient Boosting* y *Random Forest* (Abandono)

6

Conclusiones y trabajo futuro

En este capítulo se incluyen las conclusiones y el trabajo futuro que se podría seguir.

6.1. Conslusiones

Los MOOCs son cursos masivos online que cada vez cuentan con más estudiantes. Esta nueva forma de aprendizaje genera una gran cantidad de datos de los cuales se pueden extraer patrones y conclusiones gracias a la Analítica de Aprendizaje o Learning Analytics.

Estos cursos cuentan con un gran problema de abandono por parte de los estudiantes, por lo que se ha elegido el curso de *WebApp*, ofertado por la UAM para analizar los datos de sus estudiantes.

Se ha partido de las herramientas *edX-LIMS* (aplicación web que permite la visualización del desempeño de los estudiantes y recopila información sobre ellos) y *edX-MAS+* (herramienta que analiza los datos de los estudiantes y genera modelos de predicción), rediseñando esta última e integrándola a la primera.

Este nuevo sistema informático incluye un estudio sobre el abandono de los estudiantes en los cursos online asíncronos y cinco algoritmos de predicción para el aprobado y abandono del MOOC. Los algortimos de Machine Learning empleados han sido *Generalized Linear Model*, *Stochastic Gradient Boosting*, *Random Forest*, *Neural Network* y *Support Vector Machine*. Los resultados de las predicciones se pueden visualizar en la nueva vista de la interfaz gráfica de *edX-LIMS* y el sistema ya está siendo utilizado por la coordinación del curso de *WebApp*

6.2. Trabajo futuro

Son muchas las posibles líneas que se pueden seguir, aunque las más destacables serían las siguientes:

Adaptar el nuevo sistema y la aplicación de *edX-LIMS* a otros MOOCs. Esto sería muy interesante

para poder comparar qué comportamientos comunes tienen los estudiantes en diferentes cursos y qué actividades son las más relevantes para triunfar en el aprendizaje. Esta parte no se ha hecho porque implica el rediseño de todo *edX-LIMS*, puesto que esta aplicación está exclusivamente diseñada para el curso de *WebApp*, y otros cursos tienen una estructura de unidades y subunidades diferente.

Una aportación necesaria sería agregar la importancia de las variables para el resto de algoritmos. No se ha encontrado una forma correcta de hacer esto, puesto que las funciones existentes presentaban resultados incoherentes.

También se podrían añadir nuevos idiomas a la aplicación y traducir el resto de módulos, puesto que los MOOCs son internacionales y así prodrían llegar a un número mayor de usuarios.

Una mejora interesante y fundamental es predecir el aprobado y abandono a tiempo real. Para ello habría que poder predecir los indicadores de los usuarios en base al ritmo de estudio, puesto que no se tendrían datos futuros de los estudiantes. Esto haría que se pudiese saber qué estudiantes van a suspender o abandonar el curso cuando aún se puede rectificar su comportamiento.

Bibliografía

[1] "By the Numbers: MOOCs During the Pandemic." https://www.classcentral.com/report/mooc-stats-pandemic/, Accedido 2021.

[2] "State of Data Science and Machine Learning 2020." https://www.kaggle.com/kaggle-survey-2020, 2020.

[3] L. Olmos, "Sistema Informático para el Análisis de datos en entornos educativos. Trabajo de Fin de Grado. Universidad Autónoma de Madrid," Febrero 2018.

[4] J. Soberón, "Sistema Informático de apoyo a las analíticas para el aprendizaje (Learning Analytics) para entornos educativos on-line. Trabajo de Fin de Grado. Universidad Autónoma de Madrid," Julio 2020.

[5] J. Škrubej, *The Cold War for Information Technology*. 2013.

[6] M. León-Urrutia, R. Cobos, and K. Dickens, "MOOCs and their Influence on Higher Education Institutions: Perspectives from the Insiders," *Journal OF New Approches in Educational Research*, vol. 7, pp. 40–45, Enero 2018. (DOI).

[7] I. Claros, A. Garmendía, L. Echeverría, and R. Cobos, "Towards a collaborative pedagogical model in MOOCs," in *2014 IEEE Global Engineering Education Conference (EDUCON)*, pp. 905–911, 2014. (DOI).

[8] C. Lang, G. Siemens, A. Wise, and D. Gasevic, "Handbook of Learning Analytics," *SOLAR, Society for Learning Analytics and Research*, 2017.

[9] T. Elias, "Learning Analytics: Definitions, processes and potential," 2011.

[10] "edX." https://www.edx.org/es, Accedido 2021.

[11] "UAMx." https://uamx.uam.es/, Accedido 2021.

[12] R. Cobos and J. C. Ruiz-Garcia, "Improving learner engagement in MOOCs using a learning intervention system: A research study in engineering education," *Computer Applications in Engineering Education*, 2020. (DOI).

[13] R. Cobos and L. Olmos, "A Learning Analytics Tool for Predictive Modeling of Dropout and Certificate Acquisition on MOOCs for Professional Learning," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 1533–1537, 2018. (DOI).

[14] R. Cobos and V. Macías, "edX-MAS: Model Analyzer System," *TEEM*, Cádiz, 2017.

[15] V. Macías, "Herramienta para el modelado predictivo en entornos educa- tivos en línea. Trabajo de Fin de Grado. Universidad Autónoma de Madrid," Junio 2017.

[16] R. Cobos and J. Soberon, "A proposal for Monitoring the Intervention Strategy on the learning of MOOC learners," 2020.

[17] G. Siemens, *Massive open online courses: Innovation in education*. McGreal, R., Kinuthia W., & Marshall S. (Eds), 2013. Open educational resources: Innovation, research and practice Vancou-

ver: Commonwealth of Learning and Athabasca University.

[18] J. Gómez, C. Lázaro, and J. Martínez, "Situación actual de los cursos MOOC y su impacto en las organizaciones universitarias: Revisión teórica," *VIII Congreso Virtual Internacional Transformación e Innovación en las Organizaciones*, diciembre 2020.

[19] J. Reich, "Seminario eMadrid sobre «Nuevas experiencias en entornos de aprendizaje masivos». «Fallo en la interrupción: por qué la tecnología por sí sola no puede transformar la educación»," mayo 2021.

[20] "By the Numbers: MOOCs in 2020." https://www.classcentral.com/report/mooc-stats-2020/, Accedido 2021.

[21] "By the Numbers: MOOCs in 2019." https://www.classcentral.com/report/mooc-stats-2019/, Accedido 2021.

[22] A. Martínez-Monés, Y. Dimitriadis, E. Acquila-Natale, A. Álvarez, M. Caeiro-Rodríguez, R. Cobos, M. A. Conde-González, F. J. García-Peñalvo, D. Hernández-Leo, I. Menchaca, P. J. Muñoz-Merino, S. Ros, and T. Sancho-Vinuesa, "Achievements and challenges in learning analytics in Spain: The view of SNOLA," *Revista Iberoamericana de Educación a Distancia*, vol. 23, pp. 187–212, Enero 2020. (DOI).

[23] C. Romero and S. Ventura, "Educational data mining and learning analytics:An updated survey," *WIREs Data Mining Knowl Discov.*, no. 1355, 2020. (DOI).

[24] "SOLAR." https://www.solaresearch.org/, Accedido 2021.

[25] "SNOLA." https://snola.es/, Accedido 2021.

[26] R. Cobos, S. Gil, A. Lareo, and F. A. Vargas, "Open-DLAs: An Open Dashboard for Learning Analytics," in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, (New York, NY, USA), p. 265–268, Association for Computing Machinery, 2016. (DOI).

[27] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forests and Decision Trees," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 272–278, 2012.

[28] J. H. Friedman, *Stochastic Gradient Boosting*. 1999.

[29] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017. (DOI).

[30] "Bayesian machine learning." https://algorithmia.com/blog/bayesian-machine-learning, 2020.

[31] "Confusion Matrix." https://devopedia.org/confusion-matrix, Accedido 2021.

[32] "edX: Introducción al desarrollo de aplicaciones web en línea.." https://www.edx.org/course/introduccion-al-desarrollo-de-aplicaciones-web-2, Accedido 2021.

[33] "Dash." https://plotly.com/dash/, Accedido 2021.

[34] J. Sánchez, "Principios sobre Bases de Datos Relacionales." 2004.

[35] "¿Qué es una base de datos relacional?." https://www.oracle.com/es/database/what-is-a-relational-database/, Accedido 2021.

[36] A. Boicea, F. Radulescu, and L. I. Agapin, "MongoDB vs Oracle – Database Comparison," in *2012*

Third International Conference on Emerging Intelligent Data and Web Technologies, pp. 330–335, 2012. (DOI).

[37] "pandas." `https://pandas.pydata.org/`, Accedido 2021.

[38] "NumPy." `https://numpy.org/`, Accedido 2021.

[39] "scikit-learn." `https://scikit-learn.org/stable/`, Accedido 2021.

[40] "dash extensions." `https://pypi.org/project/dash-extensions/`, Accedido 2021.

[41] "PyMongo." `https://pymongo.readthedocs.io/en/stable/`, Accedido 2021.

[42] "Dash bootstrap components." `https://dash-bootstrap-components.opensource.faculty.ai/`, Accedido 2021.

[43] P. M. Moreno-Marcos, P. J. Muñoz-Merino, J. Maldonado-Mahauad, M. Pérez-Sanagustín, C. Alario-Hoyos, and C. Delgado Kloos, "Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs," *Computers & Education*, vol. 145, p. 103728, 2020. (DOI).

[44] C. Delgado Kloos, C. Alario-Hoyos, C. Fernández-Panadero, I. Estévez-Ayres, P. J. Muñoz-Merino, R. Cobos, J. Moreno, E. Tovar, R. Cabedo, N. Piedra, J. Chicaiza, and J. López, "Proyecto eMadrid: MOOCs y Analítica del Aprendizaje," *XVIII Simposio Internacional de Informática Educativa SIIE. Ediciones Universidad de Salamanca*, pp. 491–496, 2016.

[45] M. L. Bote and E. Gómez, "Predicción de pérdida de implicación de los participantes de un curso en línea masivo y abierto," *XVIII Simposio Internacional de Informática Educativa SIIE. Ediciones Universidad de Salamanca*, pp. 33–38, 2016.

[46] J. Wong, M. Baars, D. Davis, T. V. D. Zee, G.-J. Houben, and F. Paas, "Supporting Self-Regulated Learning in Online learning Environments and MOOCs: A Systematic Review," *International Journal of Human–Computer Interaction*, vol. 35, no. 4-5, pp. 356–373, 2019. (DOI).

[47] Y. Lee, J. Choi, and T. Kim, "Discriminating factors between completers of and dropouts from online learning courses," *British Journal of Educational Technology*, vol. 44, pp. 328–337, 03 2013.

[48] P. M. Moreno, *Analıtica del aprendizaje para la predicción en escenarios educativos heterogéneos. Tesis Doctoral. Universidad Carlos III Madrid*. Julio 2020.

[49] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras TensorFlow. Concepts, Tools and Techniques to Build Intelligent Systems.* O'Reilly, 2019.

[50] G. Hackeling, *Mastering Machine Learning with scikit-learn. Learn to implement and evaluate machine learning solutions with scikit-learn*. Packt, 2017.

[51] N. Keijsers, "Neural networks," in *Encyclopedia of Movement Disorders* (K. Kompoliti and L. V. Metman, eds.), pp. 257–259, Oxford: Academic Press, 2010. (DOI).

[52] P. M. Moreno-Marcos, C. Alario-Hoyos, P. J. Muñoz-Merino, and C. D. Kloos, "Prediction in MOOCs: A Review and Future Research Directions," *IEEE Transactions on Learning Technologies*, vol. 12, no. 3, pp. 384–401, 2019. (DOI).

[53] W. Noble, "What is a support vector machine?," *Nat Biotechnol*, vol. 24, pp. 1546–1696, 2006.

[54] Z. Liu, G. Gilbert, J. M. Cepeda, A. O. K. Lysdahl, L. Piciullo, H. Hefre, and S. Lacasse, "Mode-

lling of shallow landslides with machine learning algorithms," *Geoscience Frontiers*, vol. 12, no. 1, pp. 385–393, 2021. (DOI).

[55] P. Prettenhofer and G. Louppe, "Gradient Boosted Regression Trees in Scikit-Learn." 2014.

[56] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006. ROC Analysis in Pattern Recognition.

[57] "GeoGebra." `https://www.geogebra.org/?lang=es-ES`, Accedido 2021.

Definiciones

bagging Método donde algoritmos simples son utilizados en paralelo. El resultado se obtiene haciendo la media de todas las salidas de los algoritmos.

boosting Técnica de combinación de algoritmos débiles con el fin de obtener un único algoritmo más fuerte. Cada algoritmo se añade con un peso diferente.

COVID-19 Enfermedad respiratoria que causó una pandemia mundial en marzo de 2020.

dashboard Tablero de usuario que permite la interacción con elementos diámicos.

DataFrame Estructura de datos bidimensional donde se pueden guardar datos organizados en filas y columnas.

UAMx Oficina para cursos MOOC y formación online de la UAM [11].

Acrónimos

AUC Area Under the Curve.

edX-LIMS Learning Intervention Monitoring Service for edX MOOCs.

edX-MAS Model Analyzer System para edX MOOC.

FN Falso Negativo.

FP Falso Positivo.

LA Learning Analytics.

MOOCs Masive Open Online Courses.

ROC Receiver Operating Characteristic.

SNOLA Spanish Network Of Learning Analytics.

SOLAR Society for Learning Analytics Research.

SQL Structured Query Language.

SRL Self-Regulated Learning.

UAM Universidad Autónoma de Madrid.

VN Veradero Negativo.

VP Veradero Positivo.

Apéndices

A

Thinking Aloud

La técnica de Thinking Aloud o Pensar en Voz Alta se ha realizado con un usuario experto en Data Science que forma parte del equipo docente de *WebApp*.

Para ello se han establecido las siguientes tareas:

A.1. Descripción Tarea 1

Esta tarea consiste en acceder al sistema de predicciones. El el usuario debe encontrar el botón que lo permite desde la pantalla principal.

A.2. Descripción Tarea 2

El usuario tiene que mostrar que entiende bien el funcionamiento de la tabla de datos e indicadores de los usuarios. Para ello tiene que filtrar aquellos estudiantes que lleven 35 días consecutivos de inactividad y ser capaz de obtener cuál es el número máximo de días que puede llevar un usuario inscrito.

A.3. Descripción Tarea 3

El usuario tiene que mostrar en las gráficas los resultados de los algoritmos *Random Forest* y *Support Vector Machine*, obtener el tiempo de entrenamiento de *Support Vector Machine* para el día 40 de la tabla de Abandono.

A.4. Descripción Tarea 4

Descargar las tablas de *Certificado AUC Score* para el algoritmo de *Support Vector Machine* y *Importancia de las Variables para Abandono con Random Forest* para los indicadores *Certificado* y *Connected Days*

A.5. Descripción Tarea 5

El usuario tiene que entrenar 100 días de actualización del algoritmo *Linear Model* para la predicción de *Certificado*.

A.6. Resolución Tarea 1

El usuario se encuentra en la pantalla principal y busca algo que le indique cómo moverse a otra pantalla. Encuentra el botón *Predicciones* en la esquina superior derecha y lo pulsa.

No ha encontrado ningún problema y le ha parecido sencillo.

A.7. Resolución Tarea 2

El usuario se encuentra en el sistema, busca alguna tabla moviéndose con el ratón hacia abajo. Cuando la encuentra se queda un rato observándola y después pulsa las flechas de la columna *Número de días* hasta ordenarla de mayor a menor. Después busca en el desplegable de filtros el indicador deseado y aplica el filtro pulsando el botón correspondiente.

El usuario encuentra como problema que las flechas de ordenar la tabla son muy pequeñas y además no le es intuitivo el número de veces que hay que pulsarlas para ordenarlo.

Aplicar los filtros le resulta sencillo.

A.8. Resolución Tarea 3

El usuario busca en la página cómo añadir los algoritmos a las gráficas. Encuentra el desplegable y los añade. Busca la tabla indicada y se da cuenta que al pasar el ratón por encima aparecen los valores del eje Y. Busca el valor indicado y lo dice en voz alta.

Esta tarea le ha parecido sencilla y no ha encontrado ningún problema.

A.9. Resolución Tarea 4

En este momento el usuario tiene dos algoritmos mostrándose en las gráficas. Busca la gráfica señalada y pulsa el botón de descargar. Cuando abre el archivo se da cuenta de que se han añadido los dos algoritmos. Vuelve a la selección de elección de algoritmos, retira *Random Forest* y vuelve a descargar el archivo. Para la gráfica de indicadores selecciona los indicadores necesarios y hace el mismo procedimiento.

Esta tarea le parece sencilla y no encuentra ningún problema.

A.10. Resolución Tarea 5

El usuario navega al inicio de la página y comenta las diferentes opciones que ve. Selecciona *Linear Model* en la lista de algoritmos, desmarca la opción *Abandono*, marca la opción *Actualizar*, introduce el número 100 en el cuadro de texto correspondiente y pulsa el botón de entrenar.

Esta tarea le ha parecido más difícil que las anteriores, ha sentido alguna duda en el procedimiento y ha tardado más de lo habitual.

B

Primer uso

En este apéndice se muestran los pasos que tiene que seguir el usuario la primera vez que abre la aplicación. Aparecerá un error la primera vez que se abre este módulo, indicando que no existen datos.

Primero: Figura B.1. Acceder al módulo de predicciones desde la pantalla principal pulsando el botón *Predictions*

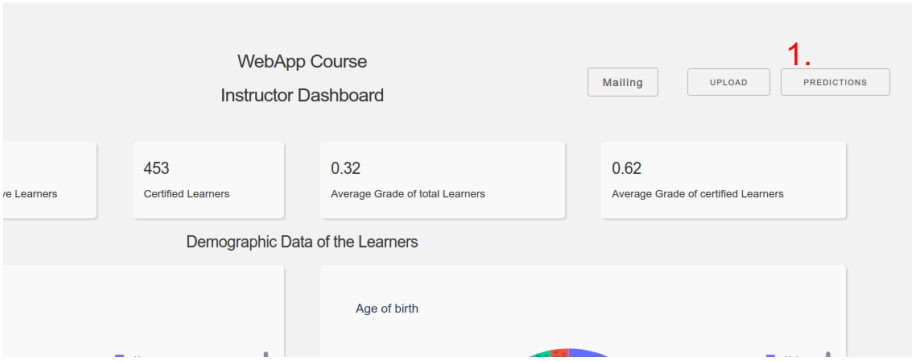

Figura B.1: Acceder al módulo de *Predicciones*

Segundo: Figura B.2. Pulsar el botón de *Crear tablas* y esperar a que termine (cuando desaparezca el símbolo de cargando).

Tercero: Reiniciar el servidor

Cuarto: Figura B.3. Seleccionar el/los algoritmos (a), elegir la/las columnas a predecir (b), introducir el número de días para las variables a predecir (c). Seleccionar actualizar (d) si ya se ha entrenado previamente el modelo seleccionado y se quiere aumentar el número de días a mostrar sin volverlos a entrenar. Finalmente pulsar el botón *Entrenar* (4).

Quinto: Volver a reiniciar el servidor

Sexto: Figura B.4. Seleccionar los algoritmos a visualizar y el tiempo.

Séptimo: Figura B.5. Pulsar el botón inferior izquierdo de la gráfica que se quiera descargar en *csv*. Se guardará en el equipo siguiendo las opciones del navegador. Figura B.6. En la parte inferior de la

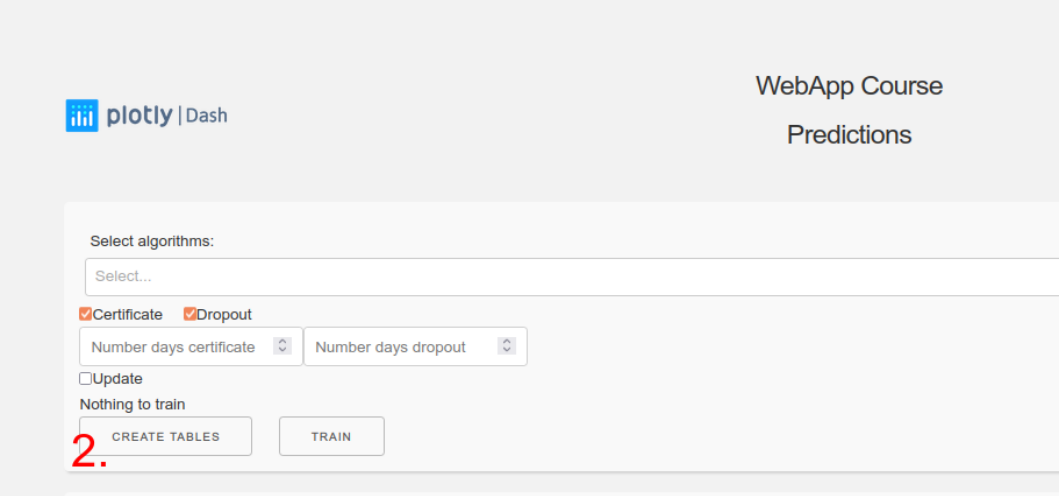

Figura B.2: Crear tablas de datos e indicadores

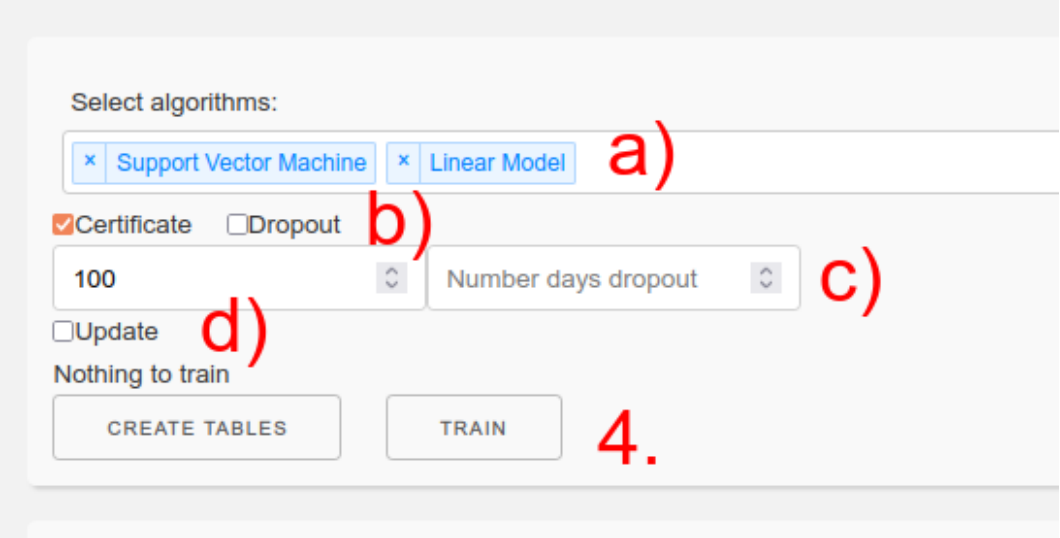

Figura B.3: Seleccionar las opciones para entrenar los modelos

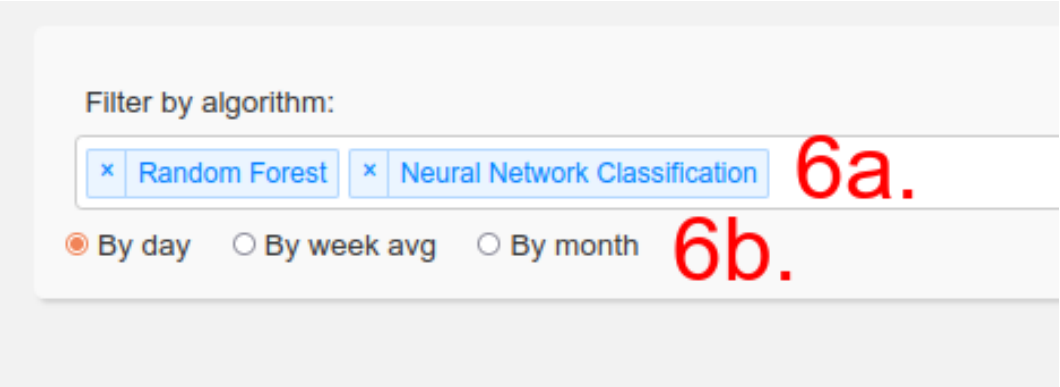

Figura B.4: Visualizar los resultados del entrenamiento

página se pueden seleccionar individualmente qué colecciones descargar de la base de datos.

Octavo: Figura B.7. Se elige el número de filas a mostrar (a). Se agregan filtros seleccionando las variables y los símbolos (b), se aplica el filtro pulsando el botón (c), y se puede guardar este escogiendo un nombre y presionando el botón *Guardar filtro* (d). También se pueden seleccionar los filtros guardados y cargarlos (e).

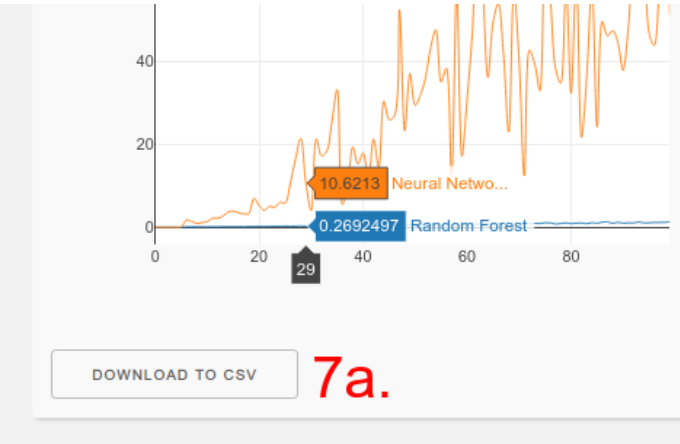

Figura B.5: Descargar gráficas en *csv*

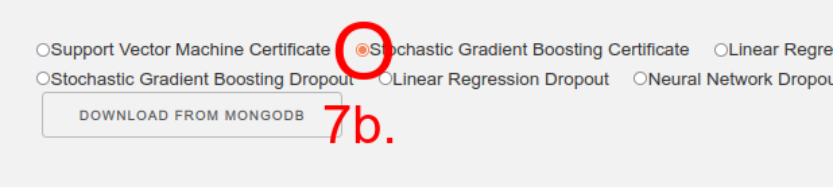

Figura B.6: Descargar colecciones en *csv*

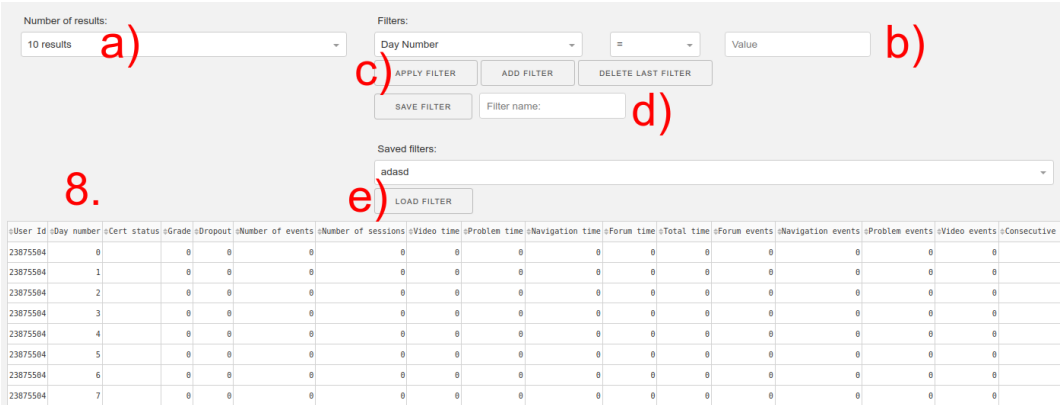

Figura B.7: Filtrado de datos a mostrar en la tabla *indicators_and_data_acc*

C

Predicción de tiempos

Se ha decidido añadir este apéndice sobre los tiempos de entrenamiento de los algoritmos.

Este anexo se ha incorporado porque puede ser útil saber cúanto tiempo va a tardar un algoritmo. Por ejemplo, para el entrenamiento y predicción de 100 días con Support Vector Machine en el entorno de desarrollo, se tardaron seis horas.

Los tiempos de entrenamiento y predicción siguen una función cuadrática de la forma C.1 debido a que cada vez se utilizan más datos, concretamente los del día anterior más los nuevos.

$$f(x) = ax^2 \tag{C.1}$$

Esto hace que se pueda intuir fácilmente en un visor de equaciones como GeoGebra [57] cuánto tiempo va a tardar un algoritmo en ser entrenado.

Si se calcula la intersección de la recta perpendicual al eje X para el día 300 con la parábola, se puede calcular que solo en entrenar el algoritmo el día 300 tardará aproximadamente 45 minutos.

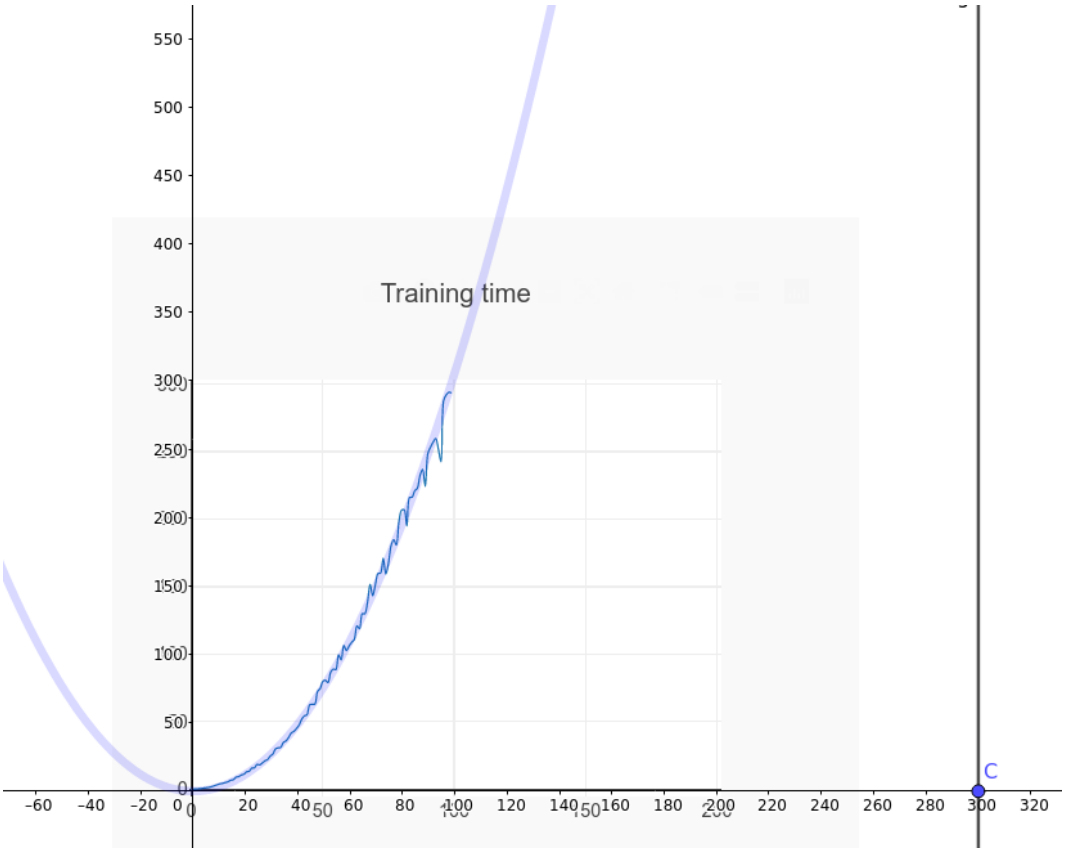

Figura C.1: Aproximación de tiempo de entrenamiento