Spark para Learning Analytics: análisis del abandono en cursos online

---

David Torres Pascual

Tutora: Dra. Estrella Pulido Cañabate

Resumen

Los MOOC son cursos online masivos y abiertos que han revolucionado la educación durante los últimos años. Sin embargo, desde el primer momento han tenido tasas de abandono muy elevadas. Numerosos estudios han tratado de abordar este tema utilizando técnicas de aprendizaje automático, buscando predecir el abandono en base a diferentes atributos.

Este trabajo es un caso de estudio que, utilizando diferentes algoritmos de clasificación y herramientas del ecosistema Big Data Apache Spark, predice el abandono o la continuidad de los usuarios de la primera edición del curso online *Jugando con Android - Aprende a Programar tu Primera App* ofertado a través de la plataforma edX.

Para ello utiliza los datos procedentes del log para generar los atributos y la clase abandono. Tras realizar una limpieza y transformación de los datos, se aplican tres algoritmos, Random Forest, SVC y Gradient Boosting y se calculan varias métricas para evaluar las clasificaciones.

Palabras clave: *MOOC, Learning Analytics, Big Data, aprendizaje automático, abandono.*

Índice

Índice de figuras

Índice de tablas

Agradecimientos

A la Cátedra UAM-IBM por el apoyo y la formación durante estos meses, y en especial a Estrella por darme la oportunidad de formar parte de ese proyecto. A Gonzalo por todos los comentarios técnicos y sus explicaciones. A Paulo por la documentación de Spark.

A los compañeros del Experto en Big Data, y particularmente a Pablo, por todo lo que me ha enseñado, a Borja, a Alex y a Arucas.

A *Gente de Cátedra*: Alex, Alexandre, Borja, Roy, Miguel, Ángel y Lydia.

A hermano.

"No matter. Try again, fail again, fail better."(Samuel Beckett)

1. Introducción

1.1. Motivación

El desarrollo de nuevas tecnologías aplicadas a la educación ha permitido la aparición de plataformas colaborativas como las *wikis*, para compartir conocimiento sobre un tema determinado, o los repositorios online de cursos (*Open Courses*), donde se almacenan vídeos, apuntes y otros recursos de cursos que se ofrecen de forma presencial en las universidades. Otra de las grandes novedades en el campo de la tecnología aplicada al conocimiento ha sido Khan Academy, que ofrece formación online sobre una gran variedad de temas. Pero fue en el año 2012 cuando tuvo lugar el nacimiento de uno de los fonómenos más relevantes para la difusión del conocimiento a través de la tecnología. Ese año fue el punto de partida de los MOOC (Massive Open Online Courses), que poco a poco se han convertido en un referente en la educación online, como complemento a la educación existente y como punta de lanza de numerosas novedades aplicadas a la transmisión del conocimiento. Podemos destacar tres plataformas relevantes, aunque hay muchas más: Coursera, fundada por profesores de Stanford; Udacity, también con profesores de esa universidad y la que primero buscó obtener beneficios económicos de esta idea; y edX, impulsada por Harvard y el MIT. Estas plataformas educativas han evolucionado y cambiado con los años, pero conservan unas características comunes: los MOOC se ofertan a un público **masivo** y apriori ilimitado; en formato **abierto**, permitiendo el acceso al contenido por parte de cualquier persona siempre que tenga acceso a Internet; y de forma **online** a través de plataformas web como las antes mencionadas.

Los MOOC comenzaron funcionando como **cursos** de estructura semanal, con vídeos, diferentes tipos de actividades y foros para comunicarse con el equipo de profesores y entre usuarios. Además, ofrecían certificados gratuitos a los usuarios que los completaban, así como diplomas oficiales a aquellos que pagaban determinadas tasas. Actualmente este modelo está cambiando [21] hacia formatos nuevos: series de cursos temáticos, cursos menos estructurados y más cortos, con menos carga de actividades y más proyectos prácticos. Sin embargo, a pesar de los cambios, del prestigio de las universidades participantes y, en general, de la calidad del material ofrecido, los bajos porcentajes de personas que superan los cursos y las enormes cifras de abandono han generado una gran preocupación en los defensores de este modelo, pero también han dado lugar a un interés

por conocer las causas [1], predecir el abandono [27, 11] y analizar si es posible extrapolar estos análisis a otros campos del *Learning Analytics*, esto es, el uso de datos producidos por los estudiantes y modelos de análisis para descubrir información y conexiones que ayuden a predecir y mejorar el aprendizaje [7].

1.2. Objetivos

Este trabajo se enmarca dentro de un proyecto conjunto de análisis de MOOC para tratar de extraer información de los mismos. Una primera parte correspondió al aprovisionamiento de los datos procedentes del log de la plataforma, su transformación en un fichero *json* y algunos análisis basados en técnicas de aprendizaje automático [9]. La segunda fase corresponde a este trabajo, y se detallará en los siguientes párrafos.

En esta fase del proyecto, la investigación busca tres objetivos centrales, diferentes pero relacionados entre sí. En primer lugar, aplicar herramientas de Big Data y algoritmos de clasificación al campo del *Learning Analytics*, desarrollando modelos para mejorar la oferta y la calidad de las instituciones educativas.

En segundo lugar, conseguir información agregada del comportamiento de los usuarios de los cursos MOOC, de tal forma que se puedan entender algunos patrones generales en los mismos aplicables a futuros diseños de estos cursos. Con este objetivo se busca que los cursos del futuro estén mejor adaptados a lo que los usuarios hacen, para que la transmisión de conocimiento sea mejor tanto para educadores como para alumnos.

Por último, identificar usuarios que abandonan el MOOC de forma temprana habiéndose implicado en cierta manera en el mismo y conocer su comportamiento y sus características para poder buscar soluciones individualizadas a sus problemas. Una detección temprana de los estudiantes de bajo rendimiento o de escasa participación que podrían acabar abandonando el curso permitiría que se llevara a cabo un plan de acción para aumentar la motivación en el aprendizaje y para satisfacer las necesidades especiales de los estudiantes individuales. Estas medidas se pueden implementar en las plataformas MOOC en el futuro para superar algunas de sus limitaciones actuales.

1.3. Estructura

Dividiremos el trabajo en cuatro secciones. En la primera de ellas repasaremos brevemente algunos estudios recientes sobre la aplicación de *Learning Analytics* en los MOOC, principalmente técnicas de aprendizaje automático para clasificación, de modo que sirvan de base para nuestra investigación y para conocer los temas que más interés están suscitando entre la comunidad académica. En este apartado también veremos en detalle las definiciones más aceptadas de los elementos que vamos a utilizar en nuestro diseño. En segundo lugar, explicaremos el diseño del estudio presentando los datos analizados, las herramientas elegidas y el modelo utilizado. Continuaremos con una tercera sección donde se presentan y discuten los resultados del trabajo. Por último, plantearemos las conclusiones y el trabajo futuro que se puede desarrollar a partir de los resultados obtenidos en este primer acercamiento a los datos.

2. Estado del arte

En los últimos años han aparecido numerosas investigaciones que utilizan los datos generados por las plataformas de cursos online para obtener información relativa al comportamiento, rendimiento y resultado de los estudiantes, así como sobre la efectividad de los cursos a la hora de mantener a los usuarios activos en la plataforma o transmitir el conocimiento. En este apartado revisaremos algunos de estos estudios, partiendo de los trabajos más generales para, posteriormente, centrarnos en aquellos que tratan sobre el abandono de un curso, analizando la forma en que tratan los diferentes conceptos y los modelos que utilizan.

A diferencia de las clases tradicionales, los estudiantes matriculados en los MOOC a menudo muestran una gran variedad de motivaciones y niveles de compromiso con las actividades propuestas y el contenido ofrecido en el curso. En consecuencia, los MOOC presentan unas tasas de abandono muy altas, lo que ha provocado tanto la preocupación de sus promotores como la curiosidad de los investigadores [8]. En general, se puede decir que hay dos grandes grupos de causas que explican este problema: por un lado, tal y como hemos mencionado, la diferente motivación de los estudiantes y, por otro, el diseño, la presentación y la calidad de los cursos. Pero, ¿cómo podemos atender a estos factores para tratar de minimizar su efecto?

Desde la literatura enfocada a *Learning Analytics* y al aprendizaje automático [16, 18, 27] se ha intentado buscar soluciones para enfrentar los desafíos que surgían desde las plataformas de educación online. La mayoría de los estudios coinciden en destacar el abandono como uno de los grandes retos a superar utilizando este tipo de técnicas analíticas. Incluso el problema del abandono ha pasado desde el mundo académico a una de las competiciones de aprendizaje automático más prestigiosas del mundo: la KDD Cup en su edición de 2015.

Debido a esta relevancia del abandono como fenómeno a estudiar, el primer aspecto que hay que aclarar a la hora de realizar el análisis es la propia **definición de abandono**. No existe una definición universalmente aceptada de abandono en los MOOC, y los diferentes investigadores y analistas han usado sus propias definiciones de acuerdo a sus estudios, por lo que las comparaciones son difíciles de realizar. Consideramos dos definiciones que agrupan diferentes formulaciones del abandono de un estudiante en un curso. Cada una de ellas aporta matices diferentes dependiendo del objetivo de los investigadores que las utilizan.

- Definición 1: la no participación en las actividades de la semana final del curso. Implica que el estudiante no realice actividades durante la última semana del curso [12, 18, 27]. La principal debilidad de esta aproximación es que no capta bien el tránsito del estudiante por el curso y puede ser sensible a usuarios que, habiendo logrado alcanzar el objetivo en la calificación, no hagan las actividades de esta última semana.

- Definición 2: dentro del análisis del abandono en perspectiva temporal, entendemos por abandono la no realización de actividades en la semana actual. Supone que no aparezca ningún evento durante ningún día de la semana que se está analizando [3, 15, 22, 24, 23].

Otro punto relevante es el tipo de evento que valoraremos a la hora de considerar que el usuario ha abandonado el curso, para tratar de no descartar a aquellos usuarios que, por ejemplo, vuelven al curso de forma puntual a realizar una acción diferente a la que nosotros hemos considerado.

Dentro de un proyecto a largo plazo como este, entenderemos que un modelo de predicción de abandono de calidad debe ser verificado sobre la base de diferentes

definiciones de abandono y hay que ser relativamente flexible en ello. Sin embargo, dentro de esta fase del proyecto nos decantaremos por ver si el usuario ha realizado algún tipo de actividad evaluable durante las siguientes dos semanas del curso.

El otro aspecto importante de la investigación es el conjunto de atributos o elementos que se analizarán para clasificar a los usuarios. Dentro de este tema hay mucha literatura que ha puesto el foco en diferentes aspectos que veremos en las siguientes líneas.

Algunos autores han abordado el tema del abandono teniendo en cuenta no solo la información (agregada y temporal) de las semanas previas del curso analizado (lo que llaman aprendizaje *in-situ*), sino también otros aspectos, como patrones transferidos procedentes de otros cursos, para intentar mejorar el algoritmo [5, 14].

En otros casos, tenemos autores que analizan el comportamiento y rendimiento en los MOOC estudiando los patrones de interacción con los videos de los cursos: por ejemplo uno basado en la secuencia de eventos creados, y otro sobre la secuencia de las posiciones visitadas en un vídeo. Con el análisis basado en eventos se extraen características fundamentales recurrentes (como cadenas de play y pausa) y con las secuencias de posiciones (desplazamiento por el vídeo) podemos ver las dificultades del usuario a la hora de comprender el contenido de los vídeos. Estos modelos de predicción mediante estos patrones pueden mejorar sustancialmente la calidad de predicción en términos de precisión. Según los autores, estos modelos son útiles en situaciones donde hay datos de entrenamiento limitados, como por ejemplo, para la detección temprana en las primeras semanas o en cursos cortos [6].

Otros autores analizan los MOOC utilizando los principios de microeconomía. Usando el modelo de riesgos proporcionales de Cox analizan la tasa de desgaste y usan datos demográficos, encontrando que los estudiantes más jóvenes, los participantes de Estados Unidos y las mujeres tienen menos probabilidades de completar el curso [2].

También encontramos estudios más concretos de elementos de los cursos, como el análisis de la participación en los foros [17] o el análisis de sentimiento a través de los comentarios de los foros [26, 20] para entender mejor la interacción de los usuarios con la plataforma.

En definitiva, existen una gran variedad de perspectivas y modelos para analizar

los MOOC. En general no hay ninguna que generalice especialmente bien debido a la diversidad de formatos de cursos. Como veremos más adelante, el modelo por el que apostaremos nosotros también se adapta al tipo de curso que estamos analizando, pero esperamos que a lo largo del proyecto en el que se incluye este trabajo se pueda obtener un modelo más generalizable.

3. Diseño

3.1. Datos

Los datos utilizados para este estudio proceden de la primera edición del curso *Jugando con Android - Aprende a Programar tu Primera App*, impartido por profesores de la Escuela Politécnica Superior en la plataforma edX durante los meses de febrero, marzo y abril de 2015. El curso tiene una duración de siete semanas y, tras un test opcional de conocimientos de Java, cada semana consta de videos, ejercicios sobre los videos, actividades, proyectos, y un examen final en la semana 7. Además hay un foro y documentación complementaria. A partir de los logs generados en esta plataforma se creó un fichero *json* semiestructurado en la primera fase del proyecto [9].

Los datos en el fichero *json* se presentan como un conjunto de eventos temporales correspondientes a las diferentes acciones que realiza un usuario en la plataforma. Cada fila corresponde a un usuario, y entre los eventos encontramos la visualización, pausa o parada de vídeos, desplazamientos en el mismo, realización y resultados de los ejercicios, comentarios en los foros, búsquedas de palabras, etc. Todos estos eventos van acompañados de un *timestamp* que ordena cronológicamente los eventos y nos indica el instante en el que tuvieron lugar.

El archivo original contiene información de 7172 usuarios. Este número corresponde al total de inscritos en el curso que realiza alguna acción en el mismo. Sin embargo, muchos de ellos realizan un número de acciones muy pequeño y, por lo tanto, tendrán muy poca relevancia en el análisis. Durante el preprocesado de los datos son eliminados aquellos usuarios que realizan menos de 50 eventos de cualquier tipo durante el curso, por lo que el número de usuarios que será analizado posteriormente se reduce a 2906. Además, semanalmente el número de usuarios se irá reduciendo y desaparecerán del

análisis a medida que avance el curso, al ser casos en los que se produce el abandono durante alguna de las semanas intermedias del curso, tal y como aparece en la Tabla 1. En el apartado dedicado a la explicación del modelo retomaremos esta idea para concretar algunos detalles importantes.

Tabla 1: Evolución semanal de los usuarios

Semana	Total usuarios	Continúa	Abandona
1	2906	2360	546
2	2360	2007	353
3	2007	1544	463
4	1544	947	597
5	947	791	156
6	791	696	95
7	696	696	-

3.2. Herramientas

A lo largo de este proyecto se ha trabajado con herramientas de dos áreas diferenciadas entre sí, Big Data y aprendizaje automático, pero que pueden funcionar conjuntamente. Para llevar a cabo el procesamiento de los datos y su análisis se ha utilizado Apache Spark, un ecosistema para Big Data que trabaja sobre memoria y permite procesar una gran cantidad de datos con rapidez y escalabilidad [28], superando y complementándose con el ecosistema previo, Hadoop, y su aplicación del algoritmo MapReduce.

Apache Spark funciona mediante transformaciones (evaluación *lazy*) y acciones de RDD (conjuntos de datos inmutables: no se puede modificar un RDD pero se pueden crear otros nuevos a partir de los existentes) [25], especialmente en entornos distribuidos, que es donde se aprovecha todo su potencial. Además, cuenta con un conjunto de librerías orientadas a tareas específicas como MLlib y ML (para algoritmos de aprendizaje automático), SparkSQL [4] (para hacer consultas SQL sobre los datos) o GraphX (trabajo con grafos), tal y como aparece en la Figura 1:

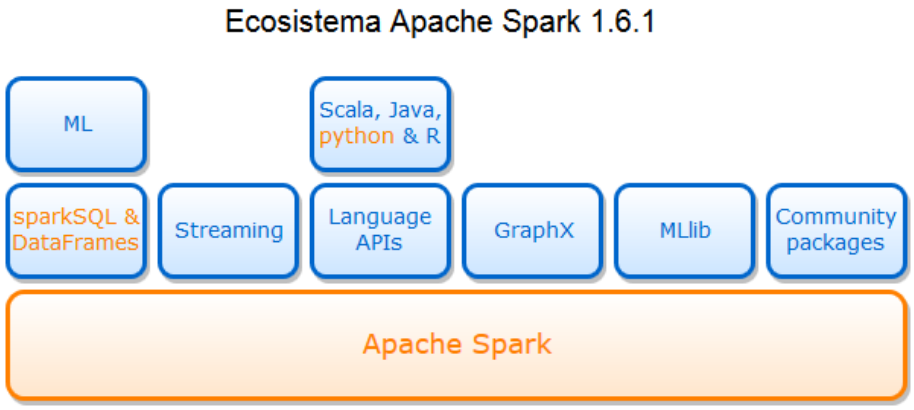


Figura 1: Ecosistema Spark con elementos utilizados

En este proyecto, la versión utilizada ha sido la 1.6.1, a través de la API para el lenguaje Python (pyspark). En cuanto al tratamiento de los datos, además del trabajo directo sobre RDD, actualmente Apache Spark soporta dos API: DataFrames (con SparkSQL) y los recientes DataSets para la versión 2.0. Estas últimas son API de alto nivel para el tratamiento y análisis con datos estructurados o semiestructurados (por ejemplo tablas de base de datos, ficheros *json, parquet...*), y optimizan la capacidad de almacenamiento y procesamiento del ecosistema. En este caso, con datos semiestructurados, se ha trabajado con los DataFrames y SparkSQL, logrando un esquema autoinferido, métodos similares a SQL para la selección, filtrado, ordenación, etc. de los datos, compatibilidad con múltiples formatos y escalabilidad de la aplicación. Este último aspecto es importante debido al objetivo final del proyecto en el que se inscribe este trabajo, ya que aunque el volumen de datos sea reducido en estos momentos, posteriormente se puede aplicar a datos procedentes de nuevos cursos o ediciones, y debe mantener tiempos de ejecución aceptables. En definitiva, Apache Spark ha permitido convertir datos semiestructurados en datos más estructurados, y por lo tanto, lograr un formato desde donde es más fácil extraer información y conocimiento.

Una vez realizado este primer procesamiento y obtenidas las tablas estructuradas, para las siguientes tareas, entre las que se incluyen tanto una parte del preprocesado, la limpieza de los datos y la aplicación del modelo a los mismos, se han utilizado varias librerías desarrolladas para el lenguaje de programación Python. Las principales han sido *pandas* y *numpy*, por la cantidad de métodos aplicados a estructuras de datos que tienen, y *sklearn* para el preprocesado, ajuste, clasificación y evaluación del modelo. Se ha utilizado

la versión 3.5 de Python sobre el entorno de desarrollo Anaconda.

3.3. Modelo

En este apartado explicaremos el modelo que vamos a implementar en nuestro análisis y los algoritmos y métricas de aprendizaje automático [10, 19, 13]. La idea central de la investigación consiste en realizar modelos con diferentes tablas que incluyan las interacciones de cada usuario extraídas a partir de una tabla principal en base a las franjas temporales que se vayan a estudiar. Estas tablas incluyen los eventos llevados a cabo por los usuarios que permanecen activos en el curso desde el inicio del mismo hasta ese momento para averiguar si abandonarán durante las dos semanas siguientes. En la Figura 2 podemos ver el ejemplo para tratar de estudiar si los usuarios abandonan en las semanas 3 ó 4 del curso.

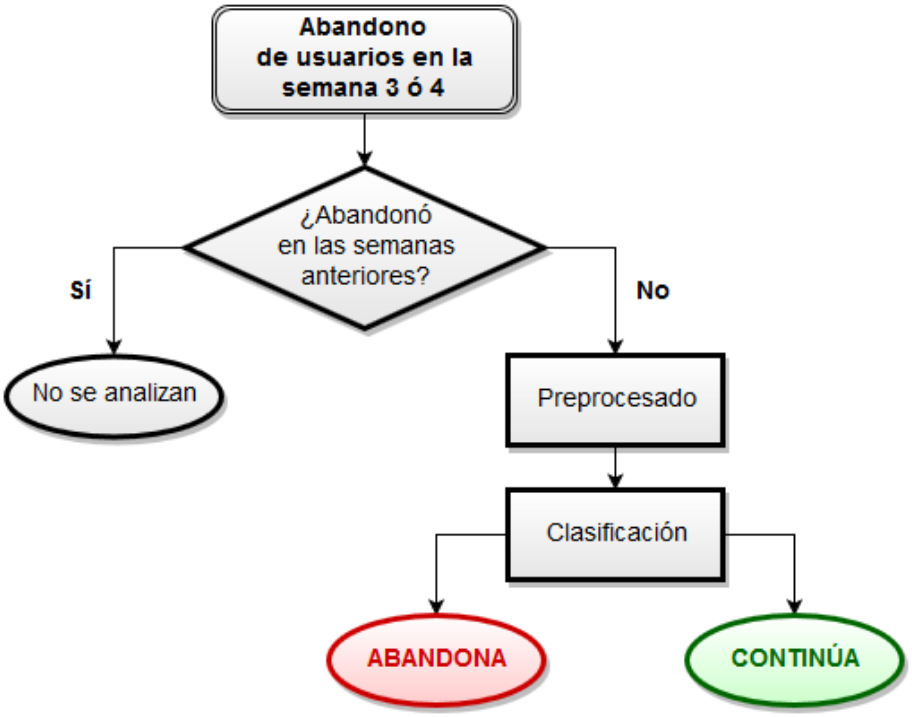


Figura 2: Diagrama de ejemplo para la clasificación de la semanas 3 y 4

Si el usuario ha abandonado en las semanas anteriores, la 1 o la 2, se elimina de la tabla, pero si ha superado la semanas previas se aplican las funciones de transformación y clasificación a sus atributos para ver si se mantiene o abandona en las dos semanas posteriores, la 3 y la 4, utilizando clasificadores de aprendizaje automático.

El aprendizaje automático se enfrenta al problema de inferir una correlación entre el conjunto de datos de los atributos, X, y una variable Y cuando se presentan como pares (X, Y). Los problemas pueden ser de clasificación, donde la Y es una variable categórica (con dos o más clases), y de regresión (Y es una variable continua). Nuestro problema se encuentra dentro de los de clasificación ya que, a partir de unos atributos X (los eventos e interacciones de los usuarios en la plataforma), clasificamos la clase Y dicotómica que denominamos **abandono** (continúa/abandona el curso).

Como hemos señalado, el modelo se basa en la utilización de un conjunto de atributos relacionados con la interacción del usuario con la plataforma para averiguar una clase *abandono* binaria, *continúa/abandona*. Estos atributos se han creado a partir de transformaciones de los eventos obtenidos del log del curso e incluyen varios tipos que se resumen en la Tabla 2:

Tabla 2: Atributos

Vídeo	Inicia un vídeo, completa un vídeo, play vídeo, pausa vídeo, se desplaza en el vídeo, total de interacciones con cada vídeo (37 vídeos diferentes)
Actividad	nº de problemas realizados, nº de intentos y calificación, nº de ejercicios del proyecto y calificación, nota del ejercicio de Java y nota del examen
Foro	nº de hilos, nº de comentarios, nº de respuestas, nº de mensajes con dudas, total de palabras y nº de búsquedas en el foro

Para definir la variable a clasificar, el **abandono**, veremos si el usuario ha realizado algún tipo de actividad evaluable durante las dos semanas siguientes del curso, es decir, si ha realizado algún problema o alguna actividad de autoevaluación. En caso negativo, entendemos que el usuario ha perdido el ritmo del curso, asumiendo que la realización de otros eventos en el curso no implica que permanezca de forma activa en el mismo.

4. Aplicación del modelo y discusión

En un primer momento, una vez obtenidos los diferentes conjuntos de datos para cada grupo de semanas, se ha realizado un análisis descriptivo (véase la Figura 3) para explorar

los datos. En dicha figura podemos observar las correlaciones (positiva en color morado y negativa en color verde), indicando con distinta intensidad de color el grado de correlación entre atributos. En la última fila vemos las correlaciones con la clase abandono (target), destacando una fuerte correlación negativa con el número de problemas y autoevaluaciones realizadas, el número de intentos y las notas obtenidas en dichas tareas.

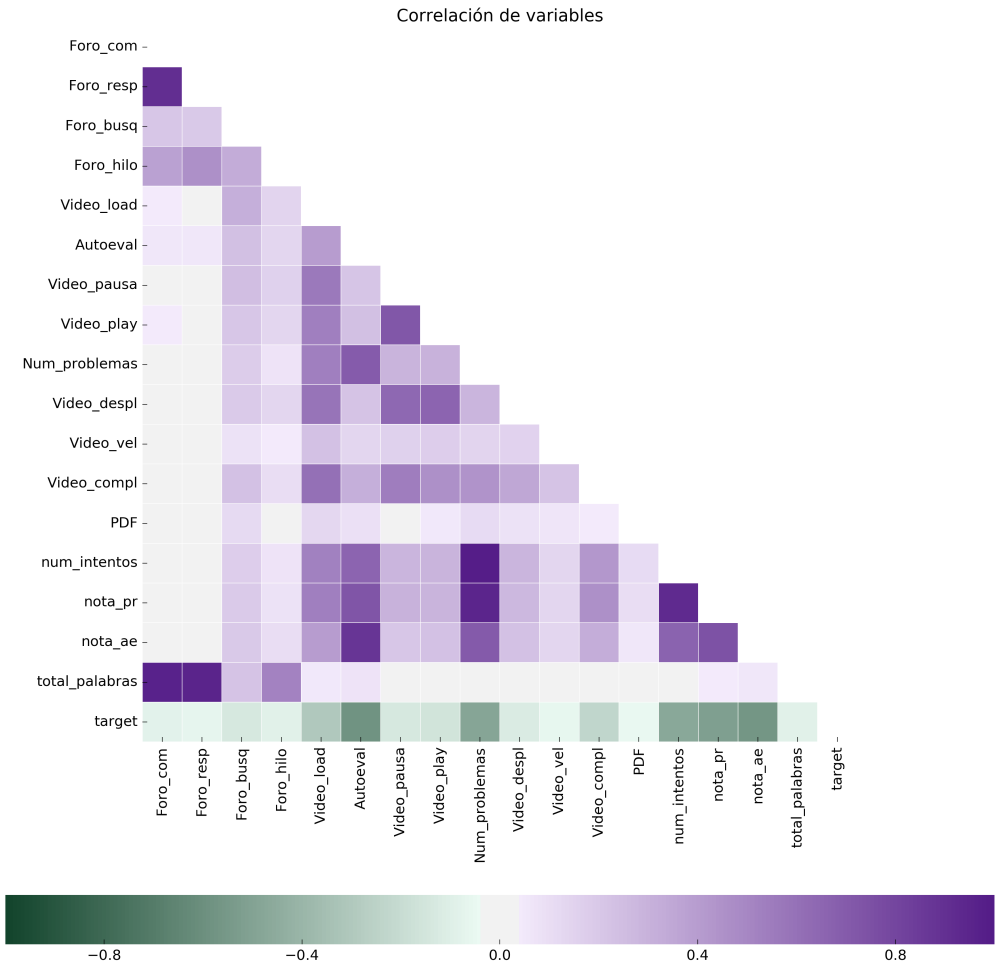


Figura 3: Correlación entre distintas variables

Para filtrar y escoger los que mejor se adaptan a nuestro conjunto de datos se ha realizado un ajuste sencillo con validación cruzada K-fold, donde K=15. A partir de este primer análisis se han observado dos aspectos. En primer lugar, que en general los resultados son pobres para el primer grupo de datos (abandono en las semanas 2 ó 3), pero mejoran en los siguientes. También se ha podido observar que existen algunos clasificadores que funcionan bastante bien con los datos de todas las semanas: SVC, Random Forest y

Gradient Boosting. Debido a ello, serán los que escogeremos para tratar de mejorar los resultados trabajando con los parámetros de los mismos.

Support Vector Classifier (SVC) es un método de clasificación incluido dentro de las Support Vector Machines (SVM) que intenta encontrar un hiperplano que separa las clases con el mayor margen posible. Este margen se define como la distancia mínima para el hiperplano de unos puntos de muestreo que se conocen como vectores de soporte.

Los métodos de conjuntos de clasificadores combinan diferentes clasificadores que pueden diferir en algoritmos, atributos y clase de entrada. Uno de ellos es el Gradient Boosting. Utiliza clasificadores muy simples como clasificadores básicos para luego centrarse en las muestras donde la clasificación ha sido incorrecta. En la ronda siguiente, se entrena otro árbol de decisión que intenta obtener estas muestras de forma correcta poniendo un peso mayor en estas muestras de entrenamiento previamente falladas. De nuevo, este clasificador probablemente tendrá algunas muestras erróneas, de modo que se reajustarán los pesos en varias rondas más.

El último algoritmo es también un conjunto de clasificadores: Random Forest. Funciona como un conjunto de clasificadores donde se entrenan y combinan varios árboles de decisión simple a través de la técnica de bagging. Se eligen los mejores por la regla de la mayoría de votos a partir de los árboles de decisión individuales.

En la Tabla 3 vemos los resultados del ajuste, clasificación y evaluación de los datos del curso siguiendo nuestro modelo y utilizando los tres clasificadores seleccionados: Random Forest, SVC y Gradient Boosting. Para intentar mejorar los resultados utilizamos algunos de los métodos implementados en la librería *sklearn* de Python. Tras dividir el conjunto de datos en uno de entrenamiento y otro de prueba, se aplica validación cruzada K-fold con K=15, pero para realizar el ajuste seleccionamos el modelo con los parámetros que mejor funcionan en cada algoritmo (usamos GridSearchCV, que selecciona los parámetros del algoritmo que dan mejores resultados atendiendo a varias medidas que explicaremos posteriormente) sobre el conjunto de validación. Después, predecimos sobre el conjunto de test y calculamos tanto las matrices de confusión, que aparecen en la Tabla 4 y siguientes, como algunas medidas de evaluación de clasificadores: *accuracy, precision, recall y la curva ROC-AUC.*

Tabla 3: *Accuracy (Acc.), Precision (Pr.) y Recall (Rec.)*

	Semanas 2-3			Semanas 3-4			Semanas 4-5			Semanas 5-6		
Alg.	Acc.	Pr.	Rec.	Acc.	Pr.	Rec.	Acc.	Pr.	Rec.	Acc.	Pr.	Rec.
SVM	0.55	0.59	0.53	0.65	0.66	0.66	0.75	0.75	0.75	0.74	0.79	0.72
RF	0.55	0.57	0.53	0.70	0.70	0.70	0.79	0.79	0.79	0.76	0.78	0.74
GB	0.33	0.65	0.40	0.56	0.67	0.61	0.66	0.74	0.68	0.62	0.75	0.57

La Tabla 3 resume la evaluación de los modelos para cada semana. La *accuracy* ($\frac{TP+TN}{P+N}$) se calcula dividiendo el total de casos clasificados correctamente entre el total de casos existentes, la *precision* ($\frac{TP}{P}$) divide los casos clasificados correctamente como positivos entre todos los que clasificamos como positivos, y el *recall* ($\frac{TP}{TP+FN}$) divide los positivos clasificados correctamente entre la suma todos los casos clasificados correctamente.

Al igual que señalamos anteriormente refiriéndonos a los primeros modelos de evaluación, podemos comprobar que los resultados de los clasificadores mejoran considerablemente con el paso de las semanas, sobre todo porque ya disponen de más datos para cada una de las observaciones que se mantienen en el curso.

Tabla 4: Matrices de confusión para el abandono en las semanas 2 ó 3

	Predicciones RF		Predicciones SVC		Predicciones GB	
	Continúa	Abandona	Continúa	Abandona	Continúa	Abandona
Continúa	108	145	135	118	36	217
Abandona	68	57	58	67	9	116

Tabla 5: Matrices de confusión para el abandono en las semanas 3 ó 4

	Predicciones RF		Predicciones SVC		Predicciones GB	
	Continúa	Abandona	Continúa	Abandona	Continúa	Abandona
Continúa	107	62	86	83	40	129
Abandona	48	148	42	154	12	184

Tabla 6: Matrices de confusión para el abandono en las semanas 4 ó 5

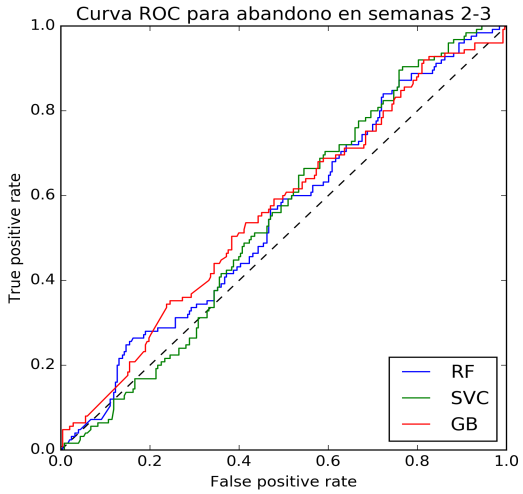
	Predicciones RF		Predicciones SVC		Predicciones GB	
	Continúa	Abandona	Continúa	Abandona	Continúa	Abandona
Continúa	113	40	117	36	66	87
Abandona	25	128	41	112	11	142

Tabla 7: Matrices de confusión para el abandono en las semanas 5 ó 6

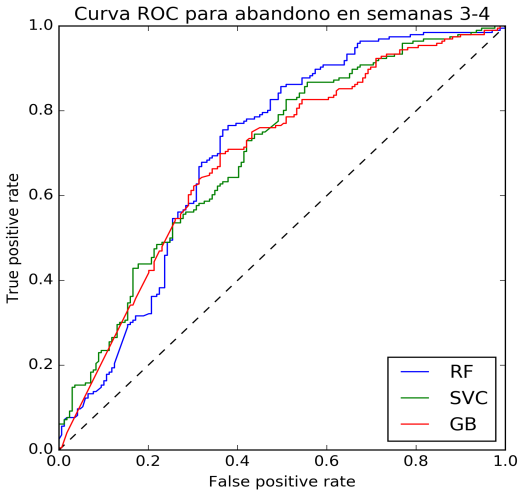
	Predicciones RF		Predicciones SVC		Predicciones GB	
	Continúa	Abandona	Continúa	Abandona	Continúa	Abandona
Continúa	130	32	121	41	92	70
Abandona	19	16	15	20	14	21

Además, las matrices de confusión (véase la Tabla 4 y siguientes) nos permiten observar que mientras que la primera semana no es particularmente buena para ningún clasficador (los resultados son muy pobres para Gradient Boosting debido a que da mucho peso a la clase abandono), en las siguientes semanas los algoritmos se comportan bastante bien a la hora de clasificar correctamente a aquellos usuarios que abandonan el curso, particularmente en las semanas 4 y 5.

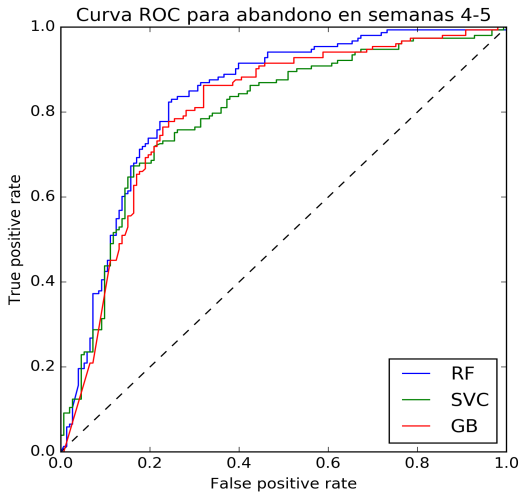
Otra medida importante es la curva ROC-AUC. Esta curva representa gráficamente una medida de calidad para algoritmos de clasificación binaria dibujando la tasa de falsos positivos frente a la tasa de verdaderos positivos (sensibilidad), que en este caso ha sido suavizada utilizando todas las probabilidades generadas por los tres modelos que hemos utilizado.



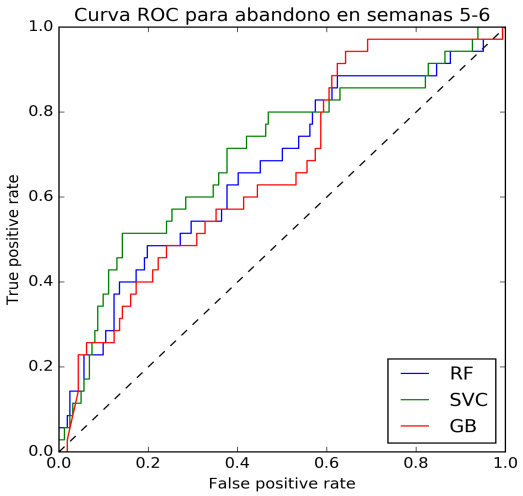
(a) Atributos de la semana 1, abandono en la 2 ó 3



(b) Atributos de las semanas 1 y 2, abandono en la 3 ó 4



(c) Atributos de las semanas 1, 2 y 3; abandono en la 4 ó 5



(d) Atributos de las semanas 1, 2, 3 y 4; abandono en la 5 ó 6

Figura 4: Curvas ROC para los cuatro conjuntos de semanas

Como podemos apreciar, en la Figura 4, la primer gráfica (a) relativa al abandono o continuidad en la segunda y tercera semana utilizando las interacciones de la primera semana, los resultados son muy pobres para los tres algoritmos. A pesar de utilizar técnicas para calcular los mejores parámetros e incorporar pesos a la clase más descompensada, no se consiguen unos resultados buenos.

Los resultados mejoran considerablemente en los dos pares de semanas posteriores, especialmente en la tercera gráfica (usando datos de las tres primeras semanas para ver el abandono en las semanas cuatro o cinco) donde se alcanzan tasas de 0.8 en el AUC. En estas dos visualizaciones se aprecia que el algoritmo Random Forest funciona ligeramente

mejor que los otros dos, además de presentar tiempos de ejecución mucho más bajos.

Por último, en la representación gráfica de los clasificadores del último par de semanas (5 y 6) vemos cómo se reducen los buenos resultados que veíamos en las dos curvas ROC anteriores. En este caso, resulta difícil clasificar el abandono por dos motivos. El primero de ellos es que las clases están descompensadas y, que nuestro supuesto está basado en la realización del último problema, pero hay muchos usuarios que completan su último problema en esta semana y no realizan el examen final de la semana siguiente. Vemos, además, que el resultado mejora ligeramente con el algoritmo de Gradient Boosting con más pesos para la clase abandono.

5. Conclusiones y trabajo futuro

Este breve estudio forma parte de la segunda fase de un proyecto más amplio basado en el análisis de MOOC con herramientas Big Data y técnicas de aprendizaje automático. En el mismo hemos visto que las metodologías de aprendizaje automático se pueden aplicar con resultados interesantes en los cursos MOOC utilizando los eventos realizados por los usuarios durante su estancia en la plataforma. En este sentido, podemos hablar de cuatro logros:

- Por un lado, los algoritmos de clasificación han funcionado bien a la hora de clasficicar a los usuarios que abandonan el curso a lo largo de las diferentes semanas utilizando el modelo que hemos presentado, especialmente desde la tercera cuando ya existen dos semanas de eventos que analizar, llegando a alcanzar métricas de acierto de 0.80.

- La introducción de pesos en los algoritmos basados en conjuntos de árboles es particularmente interesante a la hora de poner el foco en los usuarios que van a abandonar aún a costa de aumentar mucho el número de falsos positivos. Además, para medidas concretas basadas en reforzar el conocimiento y ofrecer incentivos generales, puede ser una medida efectiva.

- La realización de los problemas correctamente, así como el compromiso con proyectos a lo largo de las semanas del curso, son buenos indicadores de la continuidad en el

curso, por lo que los educadores y creadores de contenido deben trabajar en esos elementos para mejorar la continuidad de los alumnos en los cursos.

- Por último, durante el preprocesado y los análisis previos de los datos se han obtenido muchas tablas y gráficos que tienen una gran validez para presentar resultados descriptivos y para su utilización en otras fases más avanzadas de este proyecto.

A partir de este trabajo surgen dos grandes vías para el futuro. Por un lado, continuar entrenando y mejorando algoritmos de clasificación para predecir el abandono, ya sea incorporando nuevos datos procedentes de otras ediciones del curso o de otros cursos, o mejorando el preprocesado y los parámetros utilizados en el modelo.

Por otro lado, trabajar con nuevos objetivos relativos a los MOOC: realizar clústers de usuarios, identificar patrones de comportamiento y elaborar sistemas de recomendación o predicción de rendimiento.

Referencias

[1] Aleven, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R., Wang, Y., Siemens, G., Rosé, C., and Gasevic, D. The beginning of a beautiful friendship? intelligent tutoring systems and moocs. In *International Conference on Artificial Intelligence in Education* (2015), Springer International Publishing, pp. 525–528. http://www.columbia.edu/~rsb2162/paper_74.pdf.

[2] Allione, G., and Stein, R. M. Mass attrition: An analysis of drop out from principles of microeconomics mooc. *The Journal of Economic Education 47*, 2 (2016), 174–186. http://www.tandfonline.com/doi/abs/10.1080/00220485.2016.1146096?journalCode=vece20.

[3] Amnueypornsakul, B., Bhat, S., and Chinprutthiwong, P. Predicting attrition along the way: The uiuc model. http://www.aclweb.org/anthology/W14-4110.

[4] Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, M. J., Ghodsi, A., et al. Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (2015), ACM, pp. 1383–1394. https://amplab.cs.berkeley.edu/wp-content/uploads/2015/03/SparkSQLSigmod2015.pdf.

[5] Boyer, S., and Veeramachaneni, K. Transfer learning for predictive models in massive open online courses. In *Artificial Intelligence in Education* (2015), Springer, pp. 54–63. http://groups.csail.mit.edu/EVO-DesignOpt/groupWebSite/uploads/Site/Boyerveeramachaneni228.pdf.

[6] Brinton, C. G., Buccapatnam, S., Chiang, M., and Poor, H. V. Mining mooc clickstreams: On the relationship between learner video-watching behavior and performance. *arXiv* (2015). https://arxiv.org/abs/1503.06489.

[7] Buckingham Shum, S. Learning analytics policy brief. UNESCO. http://iite.unesco.org/pics/publications/en/files/3214711.pdf.

[8] Clark, D. Moocs: Course completion is the wrong measure of course success. *Class Central MOOC Report* (2016). https://www.class-central.com/report/moocs-course-completion-wrong-measure/.

[9] González-Gallego, M. Predición y análisis de interacciones de usuarios en plataformas de enseñanza online. Master's thesis, Universidad Autónoma de Madrid, Junio 2016.

[10] Grus, J. *Data Science from Scratch: First Principles with Python*. O'Reilly Media, 2015.

[11] Halawa, S., Greene, D., and Mitchell, J. Dropout prediction in moocs using learner activity features. *Proceedings of the European MOOC Stakeholder Summit 2014*. http://web.stanford.edu/~halawa/emoocs_slides_final.pdf.

[12] He, J., Bailey, J., Rubinstein, B. I., and Zhang, R. Identifying at-risk students in massive open online courses. In *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015). http://people.eng.unimelb.edu.au/zr/publications/AAAI2015-MOOC.pdf.

[13] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An Introduction to Statistical Learning*. Springer, 2013.

[14] Kennedy, G., Coffrin, C., de Barba, P., and Corrin, L. Predicting success: how learners' prior knowledge, skills and activities predict mooc performance. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge* (2015), ACM, pp. 136–140. http://dl.acm.org/citation.cfm?id=2723593.

[15] Kloft, M., Stiehler, F., Zheng, Z., and Pinkwart, N. Predicting mooc dropout over weeks using machine learning methods. https://www2.informatik.hu-berlin.de/~kloftmar/publications/emnlp_mooc.pdf.

[16] Linn, M. C., Gerard, L., Ryoo, K., McElhaney, K., Liu, L., and Rafferty, A. N. Computer-guided inquiry to improve science learning. *ScienceMag* (2014). http://science.sciencemag.org/content/344/6180/155.

[17] Onah, D. F., Sinclair, J., and Boyatt, R. Exploring the use of mooc discussion forums. In *Proceedings of London International Conference on Education* (2014), LICE, pp. 1–4. https://www2.warwick.ac.uk/fac/sci/dcs/people/research/csrmaj/daniel_onah_lice14.pdf.

[18] Ramesh, A., Goldwasser, D., Huang, B., Daume III, H., and Getoor, L. Learning latent engagement patterns of students in online courses. In *Twenty-Eighth AAAI Conference on Artificial Intelligence* (2014). http://www.dan-goldwasser.com/papers/AAAI14.pdf.

[19] Raschka, S. *Python Machine Learning*. Packt, 2015.

[20] Robinson, C., Yeomans, M., Reich, J., Hulleman, C., and Gehlbach, H. Forecasting student achievement in moocs with natural language processing. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (2016), LAK '16, ACM, pp. 383–387. http://doi.acm.org/10.1145/2883851.2883932.

[21] Shah, D. Less experimentation, more iteration: A review of mooc stats and trends in 2015. *Class Central MOOC Report* (2016). https://www.class-central.com/report/moocs-stats-and-trends-2015/.

[22] Sharkey, M., and Sanders, R. A process for predicting mooc attrition. http://www.anthology.aclweb.org/W/W14/W14-41.pdf#page=57.

[23] Sinha, T., Li, N., Jermann, P., and Dillenbourg, P. Capturing attrition intensifying structural traits from didactic interaction sequences of mooc learners. *arXiv* (2014). http://www.aclweb.org/anthology/W14-4108.

[24] Taylor, C., Veeramachaneni, K., and O'Reilly, U.-M. Likely to stop? predicting stopout in massive open online courses. *arXiv* (2014). http://arxiv.org/abs/1408.3382.

[25] Thomspon, J. Transformations and actions. In *Databricks* (2015). http://training.databricks.com/visualapi.pdf.

[26] Wen, M., Yang, D., and Rosé, C. P. Sentiment analysis in mooc discussion forums: What does it tell us. *Proceedings of educational data mining 1* (2014). http://www.cs.cmu.edu/~mwen/papers/edm2014-camera-ready.pdf.

[27] Yang, D., Sinha, T., Adamson, D., and Rose, C. P. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. *Proceedings of the 2013 NIPS Data-driven education workshop 11* (2013), 14. https://www.cs.cmu.edu/~diyiy/docs/nips13.pdf.

[28] Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., and Stoica, I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation* (2012), USENIX Association, pp. 2–2. http://www-bcf.usc.edu/~minlanyu/teach/csci599-fall12/papers/nsdi_spark.pdf.