Recommender System Using Web Scraping for Enrollment in MOOCs of Students in Engineering Careers at the Public University of Arequipa

Jerson Erick Herrera Rivera, Bachiller[1]
[1]Universidad Nacional de San Agustín de Arequipa, Perú, jherrerar@unsa.edu.pe

Abstract*– From the large amount of educational material that is accessible on the Internet, and especially with MOOCs, university students (specifically the case of the Professional School of Systems Engineering of the National University of San Agustín of Arequipa) see complicated the task of choosing some of the online courses that the e-learning websites offer. Students must access each platform (edX, Coursera, Udemy, Lernanta, etc.), search for available courses according to filters that the student sees fit, read the course contents in detail, verify the instructor's experience, duration of the course, its methodology, and other relevant characteristics for the students. Mainly, it is tedious to navigate between hundreds of courses from different platforms and find a course that fits their interests (usually not very well defined by their lack of experience or knowledge in the field of computer science). Under these circumstances, this research work seeks to develop a Recommender System based on the content of all available courses that are being offered on the edX and Udemy platforms, using Web Crawling and Web Scraping techniques to obtain the information. Thus, the system recommends to the student what courses they could study from the edX and Udemy platforms, which fit their interests in accordance with subjects they have studied at the university as part of the curriculum plan. The recommendation given will be based on the similarity between the contents of each e-learning course and contents of university subjects that the student has identified as of greater interest. To reach the objective, data has been analyzed on the courses of the two mentioned e-learning platforms and the content of the 71 subjects that make up the syllabus of the Systems Engineering career. The proposed system achieved its objective of providing objective recommendations to students during the decision making process in which e-learning courses should be enrolled according to their interests.

Keywords*— Recommender system, content based model, mooc, term frequency, inverse document frequency, scraping, crawling.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2019.1.1.42
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

17th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Industry, Innovation, And Infrastructure for Sustainable Cities and Communities", 24-26 July 2019, Jamaica.

Sistema de Recomendación Usando Web Scraping Para Matrícula en MOOCs de Estudiantes en Carrera de Ingeniería en Universidad Pública de Arequipa

Jerson Erick Herrera Rivera, Bachiller[1]
[1]Universidad Nacional de San Agustín de Arequipa, Perú, jherrerar@unsa.edu.pe

Abstract*– A partir de la gran cantidad de material educativo que es accesible en Internet, y en especial con los MOOCs, los estudiantes universitarios (en específico el caso de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín de Arequipa) ven complicada la tarea de elegir algunos de los cursos en línea que los sitios web de e-learning ofrecen. Los estudiantes deben acceder a cada plataforma (edX, Coursera, Udemy, Lernanta, etc.), buscar los cursos disponibles de acuerdo a filtros que el estudiante vea por conveniente, leer a detalle los contenidos del curso, verificar la experiencia del instructor, duración del curso, metodología de la misma, y demás características relevantes para los estudiantes. Principalmente, les es tedioso navegar entre cientos de cursos de diferentes plataformas y encontrar algún curso que se ajuste a sus intereses (generalmente no muy bien definidos por su falta de experiencia o conocimiento en el campo de la informática). Bajo estas circunstancias, el presente trabajo de investigación busca desarrollar un Sistema de Recomendación basado en el contenido de todos los cursos disponibles que estén siendo ofrecidos en las plataformas edX y Udemy, haciendo uso de técnicas de Web Crawling y Web Scraping para obtener la información. Así, el sistema recomienda al estudiante qué cursos podrían estudiar de las plataformas edX y Udemy, que se ajusten a sus intereses en concordancia a asignaturas que hayan estudiado en la universidad como parte del plan curricular. La recomendación dada estará basada en la similitud entre contenidos de cada curso e-learning y contenidos de asignaturas universitarias que el estudiante haya identificado como de mayor interés. Para alcanzar el objetivo se ha analizado data de los cursos de las dos plataformas e-learning mencionadas y el contenido de las 71 asignaturas que conforman el plan de estudios de la carrera de Ingeniería de Sistemas. El sistema propuesto alcanzó su objetivo de brindar recomendaciones objetivas a estudiantes durante la toma de decisión sobre en qué cursos e-learning deberían matricularse de acuerdo a sus intereses.

Keywords*—Recommender system, content-based model, mooc, term frequency, inverse document frequency, scraping, crawling.

I. Introducción

Los estudiantes de la Escuela Profesional de Ingeniería de Sistemas (EPIS) de la Universidad Nacional de San Agustín de Arequipa (UNSA) como parte de su motivación por aprender y ser cada vez profesionales con mayor competitividad, ven en los MOOCs (*Massive Open Online Courses*) una maravillosa oportunidad para desarrollarse personal, académica y profesionalmente.

Los MOOCs cuentan con tres elementos bien definidos: son **open**, lo que significa que cualquier persona puede usarlos para aprender; son **free**, lo que implica que no existe una barrera financiera para que puedan ser usadas por los estudiantes; y son **online**, las personas pueden accederlas desde Internet [1].

Cada vez más la tecnología y la educación se han desarrollado e integrado, plataformas MOOC como Coursera, edX, Udemy, Learnanta, etc. han impactado poderosamente el ambiente de la educación tradicional y se ha tenido que modificar los estilos de aprendizaje de los estudiantes [2].

A pesar de que cada plataforma cuenta con sus propios sistemas de recomendación, principalmente de enfoque colaborativo, resulta muy difícil para ellos dar recomendaciones a usuarios de los cuáles no se sabe absolutamente nada. Para ello es necesario que al menos exista alguna reacción, algún *like* o comentario, alguna búsqueda, que pueda resolver un problema muy común como es el de *cold start* [3], y a partir de ello se pueda generar algún perfil del nuevo usuario, y con ellos una recomendación mínimamente válida.

Los estudiantes para escoger un curso que se adapte a sus intereses deben de navegar por cada plataforma, buscar entre las cientos de páginas de resultados e ingresar a cada enlace, y leer sobre los contenidos, duración, instructores, competencias de cada uno de esos cursos. Realizar todas estas actividades para identificar los cursos idóneos para cada uno, implica mucho tiempo y esfuerzo. Por ejemplo, la plataforma Udemy cuenta con 83 páginas de resultados con 12 cursos cada uno, y sólo en la categoría "Desarrollo". Ello significaría que el estudiante en búsqueda de un curso para matricularse, tendría que visitar cerca de 1000 páginas web y leer toda su información, para finalmente, tomar la decisión y escoger un curso y estudiarlo.

Los estudiantes generalmente escogen un curso si consideran que se trata de uno conveniente para ellos de acuerdo a diferentes criterios (cursos interesantes acorde a sus preferencias, tiempo de duración, idioma, conocimientos previos, competencias que se adquieren al finalizar el curso, acceso a certificación, etc.). El principal acercamiento que los estudiantes tienen a conceptos manejados en estas plataformas son los aprendidos durante las clases de universidad. Así, los estudiantes conocen la terminología, contenidos, tecnologías, herramientas, competencias, que hayan visto durante el dictado de clases, por lo que búsquedas de cursos online están orientadas a los contenidos de las asignaturas que haya estudiado en la universidad.

Es necesario por tanto una herramienta que sugiera adecuadamente a los estudiantes en qué cursos *e-learning* pueden matricularse basados en sus preferencias de acuerdo a

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2019.1.1.42
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

las asignaturas que hayan estudiado previamente en la universidad como parte del plan de estudios, obteniéndose una recomendación objetiva y exacta, todo ello haciendo uso de *web crawling* y *web scraping* [4] (para la obtención y selección de información de cursos), y de las herramientas y técnicas de los Sistemas de Recomendación, específicamente con modelos basados en contenido.

El presente trabajo está organizado como sigue: la sección 2 da un *overview* de trabajos relacionados aplicando modelos de sistemas de recomendación de MOOCs en el ámbito educativo. La sección 3 describe la propuesta de solución, los objetivos que se buscan alcanzar con el desarrollo de la investigación y la contribución a los estudiantes de la escuela profesional. La sección 4 detalla el procedimiento para desarrollar el sistema de recomendación basado en contenido usando *web crawling* y *web scraping*. La sección 5 explica los resultados del modelo basado en contenido del sistema de recomendación desarrollado. Finalmente, la sección 6 describe las conclusiones alcanzadas.

II. Trabajos Relacionados

En [2] se reconoce que el contenido redundante y data de usuarios no adecuada, conllevan a recomendaciones ineficientes. Así, se propone un modelo de recomendación de recursos altamente preciso haciendo uso de DBN (*deep belief networks)* en ambientes MOOCs. Este modelo mina atributos de estudiantes y de contenidos de cursos, e incorpora el comportamiento del estudiante para generar vectores del perfil del estudiante como entrada al modelo profundo. El modelo es puesto en producción con data de la plataforma MOOC de la Universidad Normal de China Central.

En [5], Hou et al. proponen una metodología para recomendar cursos personalizados considerando la secuencia del currículo con soporte para *big data*. Trabaja con retroalimentación, que es usada para mejorar el desempeño de la recomendación en futuros estudiantes. La secuencialidad de cursos es analizada utilizando clúster para recomendar cursos apropiados a cada usuario. Para brindar recomendaciones de alta precisión, consideran tres puntos esenciales: heterogeneidad de los estudiantes, relaciones entre cursos y modelos de predicción en *big data*.

En [6], se propone una metodología para recomendar MOOCs aplicando técnicas de *data mining*. En primer lugar separa a estudiantes activos (han estudiado dos o más cursos) y estudiantes pasivos (han estudiado un solo curso). Ambos grupos pasan por etapas de pre procesamiento y selección de atributos. A los estudiantes activos, se les aplica técnicas como: KNN, *Decision Tree*, *CN2 Rule Induction*, Regresión Logística. Por otro lado, a los estudiantes pasivos se les aplican las técnicas de *K-means Clustering* y *Random Forest*. En el primer grupo, se obtuvieron resultados con mayor precisión utilizando la técnica de Regresión Logística. Por otro lado, en el segundo grupo (donde la información disponible fue considerablemente menor al grupo de estudiantes activos), fue necesario combinar las técnicas de *Clustering* y *Random Forest* para obtener recomendaciones eficientes y relevantes.

En [7] se presenta un *framework* basado en agentes para recomendación y descubrimiento de cursos *e-learning* de acuerdo a las necesidades y preferencias de los estudiantes. Se definieron los siguientes agentes: el asistente del estudiante, el *broker* del estudiante, el asistente proveedor de contenido, el *broker* proveedor de contenido, el facilitador de directorio, y son precisamente los *broker* quienes ofrecen la recomendación. Además, el *framework* cuenta con dos mecanismos de *matching* para recomendar los cursos más adecuados a las necesidades y preferencias de los estudiantes; el primer mecanismo es un *shell* experto basado en reglas y el segundo, es un modelo de evaluación multicriterio.

En [8] hacen uso del enfoque de filtrado colaborativo de los sistemas de recomendación, aprovechando su efectividad y eficiencia, en conjunto con el método llamado *Multi-Layer Bucketing Recommendation* (MLBR) propuesto en la investigación. Este método representa la información de estudiantes en vectores de la misma dimensión, y los organiza posteriormente en cubos que contienen estudiantes con varios cursos en común. Además, se usa la técnica de *map-reduce* para mejorar la eficiencia.

En [9] reconocen que el enfoque de filtrado colaborativo de los sistemas de recomendación no es efectivo cuando se trata con data escasa y con atributos de usuarios alta dimensión, lo que implica baja eficiencia en las recomendaciones. Se propone el uso de DBN en funciones de aproximación, extracción de atributos, predicción, clasificación. El entrenamiento del modelo DBN es alcanzado con pre-entrenamiento no supervisado y retroalimentación supervisada. Se concluye que este modelo es más eficiente que utilizar diferentes técnicas del enfoque de filtrado colaborativo.

En [10] presentan un sitio web que recomienda cursos en los cuales los usuarios pueden obtener las habilidades, que son solicitadas en las publicaciones de empleos ideales para cada uno de ellos. La recomendación está basada en modelos de Factorización de Matrices combinados con algoritmos de Filtrado Colaborativo, para predecir tendencias de cursos y posibles valoraciones de cada estudiante. La arquitectura presentada cuenta con: a) Motor de recomendación, b) sitio web (sistema de monitoreo, sistema de valoración, motor de búsqueda, alertas), c) *Web Crawler* (busca proveedores de MOOCs y redes sociales), y d) Base de datos (gestión de usuarios, cursos, habilidades, empleos).

En [11] se propone un sistema de recomendación híbrido basado en *machine learning*. Hace uso de valoraciones implícitas sobre los cursos, para determinar el comportamiento de cada estudiante y generar recomendaciones para usuarios con preferencias similares. El sistema es entrenado con gradiente descendente. El principal inconveniente encontrado es lo computacionalmente costoso que resulta realizar recomendaciones en tiempo real. Para resolver este

inconveniente, se propone el concepto de vecindario, y con ello el uso de técnicas de *clustering*.

En [12] presenta una aplicación que recomienda curso usando CBR (*Case-Based Reasoning*). CBR es el proceso de resolver nuevos problemas basados en soluciones de problemas similares del pasado. La arquitectura del sistema cuenta con tres capas: a) capa interfaz de usuario, b) capa de funciones del sistema (creación y procesamiento de casos) y c) capa de data del sistema (conformado por el caso base, caso del usuario, caso de usuario similar). La última capa es apoyada por un *crawler* que aprovecha el contenido de archivos XML o HTML, para enriquecer la base de datos y generar recomendaciones más precisas.

III. Propuesta de Solución

Ante la problemática descrita previamente, es que se propone el desarrollo de un Sistema de Recomendación de MOOCs. Los Sistemas de Recomendación (*Recommender Systems* - RS) [13] son técnicas y herramientas de software que brindan sugerencias de ítem para ser usados por un usuario. Las sugerencias dadas por un RS están dirigidas a apoyar a los usuarios procesos de toma de decisiones, tales como qué productos comprar, qué música escuchar, o qué noticias leer [14]. Los RS son principalmente dirigidos hacia individuos que carecen de competencias o experiencia personal para evaluar la cantidad abrumadora de elementos que un sitio web puede ofrecer [15].

En los RS se pueden distinguir básicamente dos clases de enfoques de recomendación: basado en contenido y filtrado colaborativo. En los RS basado en contenido, los atributos descriptivos de los ítems son usados para hacer las recomendaciones. El término "contenido" hace referencia a esas descripciones. Así, los ratings y el comportamiento de los usuarios, son combinados con la información del contenido disponible en los ítems [16].

Haciendo el paralelo con el propósito de la investigación, los ítems equivalen a los diferentes cursos que son ofrecidos en plataformas de MOOCs. El contenido equivale al texto del resumen del curso, contenido, y las competencias que se lograrían. Incluso el contenido también involucra la sumilla de las asignaturas que forman parte del Plan de Estudios, que los estudiantes identifiquen que son de mayor interés. Los atributos son el resultado del análisis de textos que se van a aplicar a la información de cada curso y de cada asignatura. Los usuarios hacen referencia a los estudiantes de la EPIS que tienen pensado seguir un MOOC en alguna plataforma (Udemy, edX). Finalmente, los ratings equivalen a la preferencia de cada estudiante sobre las asignaturas que haya cursado, y por tanto, el comportamiento del usuario hace referencia a la predisposición que tiene cada estudiante hacia un campo dentro de la computación, donde mostraría mayor rendimiento y terminaría el curso satisfactoriamente.

A. *Objetivo General*

Diseñar e implementar una arquitectura de Sistema de Recomendación basado en contenido que sugiera eficiente y objetivamente a los estudiantes en qué MOOCs ofrecidos en Udemy y edX deberían matricularse, basados en sus preferencias.

Para ello es necesario analizar los preferencias de cada estudiante (el estudiante indica qué asignaturas han significado mayor interés durante su vida universitaria) y analizar el contenido de todos los cursos que son ofrecidos en esas plataformas (para ello se va a analizar el contenido, descripción y competencias que se obtendrán de cada curso haciendo uso de herramientas de análisis de textos).

B. *Objetivos Específicos*

1) Recolectar y estructurar la data de las sumillas que forman parte de la malla curricular.
2) Recolectar y estructurar la información de cursos de Udemy y edX, a través de las herramientas de *web crawling* y *web scraping*.
3) Analizar la información de cursos para generar modelos de predicción basado en contenido.
4) Generar una recomendación a cada estudiante del curso que debería elegir de acuerdo a sus intereses.

C. *Resume General*

La presente investigación busca resolver un problema muy común entre los estudiantes universitarios, y aún con mayor preponderancia en aquellos que están inmersos en temas de ciencia y tecnología, mediante el análisis de contenidos de cursos *online*. Todas las herramientas y técnicas utilizadas para el desarrollo de la propuesta, forman parte de *Educational Data Mining* (EDM) [17][18][19] que está enfocado en el descubrimiento de conocimiento invisible desde la base de datos educativa. EDM puede ser aplicado para descubrir patrones en conjunto de datos para automatizar el proceso de toma de decisiones de instructores, estudiantes y administradores [20].

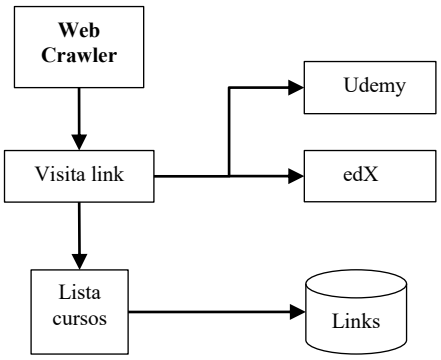

Fig. 1 Proceso de funcionamiento del web crawler.

El sistema propuesto ha involucrado el análisis, diseño e implementación de un RS que permitirá a los estudiantes saber en qué cursos ofrecidos por Udemy (https://www.udemy.com) y edX (https://www.edx.org) deberían matricularse acorde a sus preferencias (que son especificados por ellos mismos, al indicar sus asignaturas universitarias preferidas) y reducir de ese modo, el tiempo en encontrar un curso idóneo y reducir la posibilidad de deserción del curso elegido. Algunas

contribuciones que se buscan son: a) RS de MOOCs a matricularse para estudiantes de la EPIS, b) identificación de cursos más afines a cada estudiante de acuerdo a analizadores de texto, c) *clustering* de MOOCs según el contenido del mismo, su resumen, y competencias que se buscan alcanzar, d) RS web en base a modelos de recomendación basado en contenido utilizando las herramientas de *web crawling* y *web scraping*.

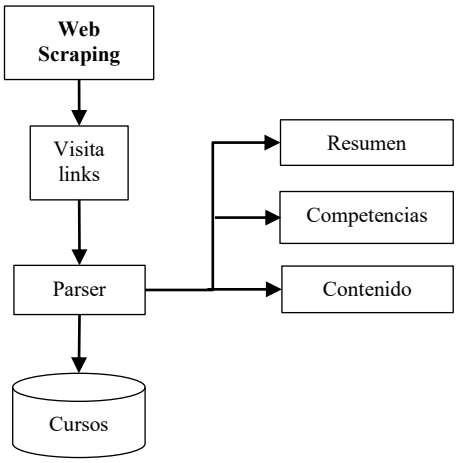

Fig. 2 Proceso de funcionamiento del web scraping.

En la Fig. 1 se muestra la secuencia de actividades realizadas por el *web crawler*. En primer lugar, este debe contar con links de inicio para cada plataforma, a partir de los cuales hace una búsqueda recursiva de todos los cursos que encuentre, y los va almacenando en una lista para que posteriormente sean trabajados por el *web scraping*.

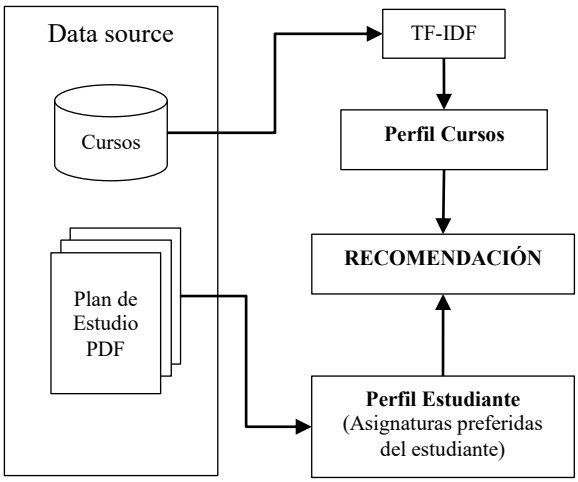

Fig. 3 Secuencia de actividades para el desarrollo del Sistema de Recomendación propuesto.

En la Fig. 2 se muestra la secuencia de actividades realizadas por el *web scraping*. Iterativamente va recorriendo cada link almacenado, ingresa a la página web y extrae el resumen, contenido y competencias esperadas de todos los cursos. Este procedimiento lo hace a través de un *parser* que ubica y guarda las etiquetas deseadas. Posteriormente, esta información capturada es almacenada para ser usada durante la generación de modelos de recomendación.

La Fig. 3 muestra el procedimiento para alcanzar el desarrollo de la propuesta y poder alcanzar los objetivos planteados previamente. Ello involucra la obtención de una base de datos con información de las asignaturas universitarias y de los cursos de las plataformas *e-learning*. Tras esto, se ha realizado dos procesos: 1) construir el perfil del estudiante haciendo uso de las asignaturas especificadas por los estudiantes de acuerdo a su preferencia y, poder hacer un análisis de texto de las sumillas de dichos cursos y 2) construir el perfil de todos los cursos *e-learning* aplicando análisis de texto de su información capturada por el *web scraping*. Finalmente se analiza el perfil del estudiante con el perfil de todos los cursos y se identifica con cuáles existe mayor afinidad y éstos son recomendados al estudiante.

IV. Desarrollo de la Solución

Para la implementación de la propuesta se utiliza el lenguaje de programación Python, en el ambiente de desarrollo Jupyter [21]. Adicionalmente, fue necesaria la incorporación de las librerías de Python: pandas [22], numpy [23], tika, pickle, sklearn [24], scipy [25], selenium, unicode.

A. Pre procesamiento de data

Para determinar las preferencias de los estudiantes, es necesario contar con información de cada asignatura que él mismo indique que son de su preferencia. Dicha información fue recolectada de las sumillas de las 71 asignaturas que forman parte del Plan de Estudios (en formato PDF). Para ello, se utiliza la librería tika y pandas, para ser almacenado en archivos de texto para su posterior análisis de textos.

Para generar los modelos de cada curso ofrecido en Udemy y edx se hizo uso de un *web crawler* y de *web scraping*.

El primero debe recorrer las páginas de ambas plataformas en búsqueda de cursos. Eso incluye navegar hacia diferentes páginas dentro de una paginación de resultados. Para encontrar el enlace de cada curso, se busca la etiqueta '<a>', y se extrae su atributo 'href' dentro del código HTML de la página visitada. Toda la información recolectada, luego es almacenada en un archivo csv (*comma-separated values*), para ser utilizada posteriormente por el *web scraping*.

El *web scraping* haciendo uso de la lista generada previamente, visita el enlace de cada curso encontrado y extrae información relevante de cada uno de ellos. En el caso de edX, para extraer el texto del campo 'Acerca del curso', se debe buscar la etiqueta con la hoja de estilos 'course-description wysiwyg-content'; y para extraer el texto del campo 'Lo que aprederás', se debe buscar la etiqueta con la hoja de estilos 'course-info-list wysiwyg-content'.

En el caso Udemy, para extraer el texto del campo 'Descripción', se debe buscar la etiqueta con la hoja de estilos

'description__title'; para extraer el texto del campo 'Lo que aprederás', se debe buscar la etiqueta con la hoja de estilos 'what-you-get__text', y en el caso del contenido, se busca la etiqueta con la hoja de estilos 'curriculum-header-title'.

Esta información es procesada, e incorporada al archivo csv de cursos, para que posteriormente se le aplique análisis de textos y se determinen los modelos de predicción.

Durante la aplicación del *web crawler* y de *web scraping* hubo problemas con los servicios ofrecidos por PerimeterX (https://www.perimeterx.com), quienes protegen a estas plataformas del envío automático de solicitudes a cada página. Esto fue solucionado en parte, utilizando un grupo de proxy que sirve como intermediario en las solicitudes enviadas a estas dos plataformas evitando el bloqueo temporal de los resultados.

B. *Definición del perfil del estudiante*

Los estudiantes al encontrarse muy relacionados a temas de su carrera a través de la experiencia que han tenido al estudiar asignaturas universitarias, se asume que los cursos *online* que pueden estudiar, son similares a las asignaturas universitarias que consideren de mayor preferencia. Así, el estudiante indica qué asignaturas son de mayor de preferencia para ellos, y partir del contenido de las sumillas de esas asignaturas se define el perfil inicial del estudiante. O podría indicar él mismo algunas palabras clave que son de interés, y del mismo modo, formarán parte de su perfil.

En Tabla I se listan las seis palabras más relevantes para un estudiante que, por ejemplo, indicó un conjunto de términos que serán usados para definirlo; y partir de ello generar un perfil. Cada palabra está ordenada según un índice con respecto al conjunto de los términos que forman parte del diccionario. Este vector propio del estudiante, es el que será utilizado para generar el perfil del estudiante.

TABLA I
PALABRAS CLAVE DEL ESTUDIANTE EJEMPLO

Índice de relevancia	Palabra clave
0.466	patrones
0.453	spring
0.387	orientada
0.387	Java
0.331	objetos
0.323	javascript

C. *Definición del modelo de recomendación de cursos*

A partir de la información de cada curso de Udemy y edX obtenido con el *web crawler* y el *web scraping*, y haciendo uso del objeto TfidfVectorizer [26] de la librería sklearn, se genera una vocabulario de características de todos los cursos y finalmente, se genera una matriz de características TF-IDF.

Los documentos son codificados por TF-IDF como vectores en un espacio euclidiano [3]. Las dimensiones del espacio corresponden a las características que aparecen en el vocabulario. TF (t*erm frequency*) describe qué tan frecuente un cierto término aparece en un documento (asumiendo que las palabras importantes aparecen más a menudo). IDF (*inverse document frequency*) es la medida que se combina con el TF; su objetivo es reducir el peso de términos que aparecen muy a menudo en todos los documentos. La idea es que esos términos muy frecuentes no son útiles para discriminar entre documentos, por lo que se debe dar más peso a las palabras que aparecen en unos pocos documentos.

Es necesario definir algunos parámetros durante el proceso de vectorización de características. Uno de ellos es el conjunto de *stopwords* [16]; que evitan que palabras irrelevantes formen parte del vocabulario del modelo.

D. *Generación de Recomendación*

A continuación se debe contrastar el perfil del estudiante contra la matriz TF-IDF generada previamente para identificar la similitud con el vocabulario generado con toda la información de los cursos. Con la nueva matriz y haciendo uso de sklearn se calcula una matriz que indique la similitud entre las diferentes sumillas y el perfil del estudiante utilizando la similitud coseno.

Finalmente, la salida del proceso es una lista de cursos más similares a la preferencia del estudiante, acompañado del índice de similitud y el enlace del curso, ya sea en la plataforma Udemy o edX.

V. RESULTADOS

En Tabla II se muestra los cursos recomendados de las plataformas Udemy y edX, para el estudiante de ejemplo. A partir de las preferencias dadas por el estudiante, se genera la tabla que muestra información de la recomendación, como el índice de coincidencia entre el perfil del estudiante y el curso encontrado. Además, se muestra el nombre del curso, y el enlace del mismo, para que el estudiante pueda visitarlo y verificar la validez de la recomendación.

TABLA II
RECOMENDACIONES DE CURSOS PARA EL ESTUDIANTE EJEMPLO

Índice de relevancia	Nombre del curso	Enlace del curso
0.52757212	Introducción a la programación en Java: empezando a programar	https://www.edx.org/es/course/introduccion-a-la-programacion-en-java-empezando-a-programar
0.47675875	Curso de TypeScript - El lenguaje utilizado por Angular 2	https://www.udemy.com/curso-de-typescript-el-lenguaje-utilizado-por-angular-2/
0.44737383	Aprende Javascript y crea APIs con NodeJS, Angular y MongoDB	https://www.udemy.com/aprende-a-programar-con-javascript-desde-cero/
0.41027638	Club Java Master: De Novato a Experto Java. +71 horas y más!	https://www.udemy.com/club-java-master-de-novato-a-experto-java-javaee-spring-hibernate-jpa/
0.40083150	Desarrollo Web con Spring 4	https://www.udemy.com/desarrollo-web-con-spring/

Asimismo, se puede buscar cursos similares a partir de un curso que haya generado interés en el estudiante. Por ejemplo, si se necesita buscar cursos similares a 'Introducción a la programación en Java: empezando a programar' ofrecido por la plataforma edX, la solución busca entre los cursos de

ambas plataformas, y genera la lista mostrada en la Tabla III. Los campos de información son idénticos a los descritos previamente.

TABLA III
CURSOS SIMILARES AL CURSO EJEMPLO

Índice de relevancia	Nombre del curso	Enlace del curso
0.76526098	Introducción a la programación en Java: escribiendo buen código	https://www.edx.org/es/course/introduccion-a-la-programacion-en-java-escribiendo-buen-codigo
0.70426827	Aprende Programación en Java (de Básico a Avanzado)	https://www.udemy.com/aprende-programacion-en-java-desde-cero/
0.59894801	Universidad Java: De Cero a Master +67 hrs (Java 11 update)!	https://www.udemy.com/universidad-java-especialista-en-java-desde-cero-a-master/
0.54679553	Java y BlueJ | Introducción a las Bases de la Programación	https://www.udemy.com/programacion/
0.54434759	Introducción a la programación en Java: estructuras de datos y algoritmos	https://www.edx.org/es/course/introduccion-a-la-programacion-en-java-estructuras-de-datos-y-algoritmos

VI. CONCLUSIONES

A partir del desarrollo del sistema de recomendación de MOOCs, propuesto como objetivo general de la presente investigación, se puede concluir que se ha alcanzado dicho fin, puesto que la solución recomienda eficaz y eficientemente a los estudiantes los cursos ofrecidos en Udemy y edX más idóneos para ellos, de acuerdo a las preferencias que los estudiantes han indicado explícitamente, creando en ellos gran satisfacción con los resultados.

Es necesario mencionar, la gran ayuda que significó el uso de herramientas de consulta automática como son el web crawler y web scraping, dado que sirvieron para visitar y recolectar información automáticamente, reduciendo significativamente el tiempo de consulta y procesamiento de data de cada plataforma.

ACKNOWLEDGMENT

Me gustaría agradecer a Víctor Cornejo Aparicio - Director de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional de San Agustín - por el pronto apoyo con material necesario para la investigación.

REFERENCIAS

[1] J. Kay, P. Reimann, E. Diebold, and B. Kummerfeld, "MOOCs: So many learners, so much potential.," *IEEE Intell. Syst.*, vol. 28, no. 3, pp. 70–77, 2013.

[2] H. Zhang, T. Huang, Z. Lv, S. Liu, and H. Yang, "MOOCRC: A Highly Accurate Resource Recommendation Model for Use in MOOC Environments," *Mob. Networks Appl.*, 2018.

[3] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, vol. 40. Cambridge, 2011.

[4] R. Mitchell, *Web Scraping with Python, Collecting data from the modern web*, First Edit. O'Reilly, 2015.

[5] Y. Hou, P. Zhou, J. Xu, and D. O. Wu, "Course recommendation of MOOC with big data support: A contextual online learning approach," *INFOCOM 2018 - IEEE Conf. Comput. Commun. Work.*, pp. 106–111, 2018.

[6] H. Jain, "Applying Data Mining Techniques in MOOC Recommender System for Generating Course Recommendations," Thapar University, 2017.

[7] N. Manouselis and D. Sampson, "Agent-based E-learning course recommendation: Matching learner characteristics with content attributes," *Int. J. Comput. Appl.*, vol. 25, no. 1, pp. 50–64, 2003.

[8] Y. Pang, Y. Jin, Y. Zhang, and T. Zhu, "Collaborative filtering recommendation for MOOC application," *Comput. Appl. Eng. Educ.*, vol. 25, no. 1, pp. 120–128, 2017.

[9] H. Zhang, H. Yang, T. Huang, and G. Zhan, "DBNCF: Personalized courses recommendation system based on DBN in MOOC environment," *Proc. - 2017 Int. Symp. Educ. Technol. ISET 2017*, pp. 106–108, 2017.

[10] P. Symeonidis and D. Malakoudis, "MoocRec.com : Massive open online courses recommender system," *CEUR Workshop Proc.*, vol. 1688, pp. 3–4, 2016.

[11] V. Garg and R. Tiwari, "Hybrid massive open online course (MOOC) recommendation system using machine learning," *Int. Conf. Recent Trends Eng. Sci. Technol. - (ICRTEST 2016)*, pp. 1–5, 2016.

[12] F. Bousbahi and H. Chorfi, "MOOC-Rec: A Case Based Recommender System for MOOCs," *Procedia - Soc. Behav. Sci.*, vol. 195, pp. 1813–1822, 2015.

[13] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, Second Edi., vol. 54. 2015.

[14] A. S. Lampropoulos and G. A. Tsihrintzis, *Machine Learning Paradigms Applications in Recommender Systems*. 2015.

[15] A. Klašnja-Milićević, M. Ivanović, and A. Nanopoulos, "Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions," *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 571–604, 2015.

[16] C. C. Aggarwal, *Recommender Systems The TextBook*, vol. 40, no. 3. Springer, 2016.

[17] C. Romero and S. Ventura, "Data mining in education," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 3, no. 1, pp. 12–27, 2013.

[18] R. Jindal and M. D. Borah, "A Survey on Educational Data Mining and Research Trends," *Int. J. Database Manag. Syst.*, vol. 5, no. 3, pp. 53–73, 2013.

[19] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," *Expert Syst. Appl.*, vol. 41, no. 4 PART 1, pp. 1432–1462, 2014.

[20] A. Sheshasaayee and M. Nazreen Bee, "E-learning: Mode to improve the quality of educational system," *Smart Innov. Syst. Technol.*, vol. 78, pp. 559–566, 2018.

[21] M. Ragan-Kelley *et al.*, "The Jupyter/IPython architecture: a unified view of computational research, from interactive exploration to communication and publication.," in *AGU Fall Meeting Abstracts*, 2014.

[22] W. McKinney, "pandas: a Python data analysis library," *see http//pandas. pydata. org/. Google Sch.*, 2015.

[23] T. E. Oliphant, "Guide to NumPy, 2nd," *USA Creat. Indep. Publ. Platf.*, 2015.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in {P}ython," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] E. Jones, T. Oliphant, P. Peterson, and others, "{SciPy}: Open source scientific tools for {Python}." .

[26] D. Cournapeau and M. Brucher, "TfidfVectorizer," 2007. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. [Accessed: 11-Jan-2019].