UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



MINERÍA DE TEXTOS PARA CONSTRUIR UN MODELO DE MINERÍA DE DATOS PARA RECOMENDACIÓN DE OFERTAS LABORALES

TESIS:

PRESENTADO POR:

HONORIO APAZA ALANOCA

Para optar el Título Profesional de:

INGENIERO DE SISTEMAS E INFORMÁTICA

MOQUEGUA – PERÚ

2019

02

UNIVERSIDAD NACIONAL DE MOQUEGUA
CARRERA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ACTA DE SUSTENTACIÓN

El Jurado suscribiente ha calificado el trabajo de Tesis:

TITULADO: Minería de Textos para construir un modelo de Minería de Datos para recomendación de ofertas laborales

PRESIDENTE: Hugo Ester Tito Chura

MIEMBROS: Carlos Alberto Silva Delgado

José Antenio Guzmán Valdivia

ASESOR: Vaneza Flores Gutierrez

PRESENTADO POR EL BACHILLER: Honorio Apaza Alanoca

DE LA PROMOCIÓN: 2016

CUYOS RESULTADOS HAN SIDO LOS SIGUIENTES:

CALIFICATIVO DEL TRABAJO: Quince

CALIFICATIVO DE LA SUSTENTACIÓN: Quince

CALIFICATIVO FINAL: Quince (APROBADO)

POR LO EXPUESTO, EL BACHILLER: Honorio Apaza Alanoca

HA SIDO DECLARADO EXPEDITO PARA QUE SE LE CONFIERA EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMÁTICA

EN FE DE ELLO QUEDA ASENTADA LA PRESENTE ACTA

ILO, A LOS 29 DÍAS DE Mayo DE 2019

Presidente

Miembro

Miembro

El presente trabajo dedico:

A Dios, por haberme permitido lograr este anhelo muy deseado, por su infinita bondad y amor.

A mi madre Asunción Alanoca, por haberme apoyado en todo momento, por la motivación constante que me ha permitido ser una persona de bien y perseverado. Te lo dedico este peldaño de mi vida profesional con mucho amor y de todo corazón a ti madre, que siempre me guiaste desde el cielo como tal águila que vigila a su polluelo en tierra.

A mi padre Gervacio Apaza, por su paciencia, sacrificio y constancia y por haberme confiado el valor mostrado de salir adelante.

Agradecimiento:

En primer lugar, agradezco a Dios por haberme concedido este anhelo deseado, por haberme dado la sabiduría suficiente para superar todos los obstáculos y sus bendiciones.

A mi familia, porque fue la motivación constante en todo el tiempo de ventarme con el ánimo de progresar cada vez más, a mi mamá Asunción Alanoca, por su ejemplar lucha ante la adversidad de la vida y mantener la esperanza hasta el final de la batalla, a mi papá Gervacio Apaza, por su lucha constante en la vida y por haberme confiado este reto. A mis hermanos Rolando Apaza y Licerio Apaza por su apoyo incondicional antes y durante mi vida universitaria.

Al ex presidente Ollanta Humala, por haber apostado por jóvenes talentos de recursos económicos bajos y de extrema pobreza, al programa nacional de beca y crédito educativo – beca 18, gracias por confiar y apostar.

A la Universidad Nacional de Moquegua, mi alma mater, por haberme dado la oportunidad de formar parte de esta casa de estudio y ser parte del proyecto de "proyectos de investigación de tesis de estudiantes y egresados para obtención del título profesional, con financiamiento de canon minero, sobre canon y regalías mineras, UNAM -2016", y por la formación profesional a través de la Escuela Profesional de Ingeniería de Sistemas e Informática y sus profesores.

Resumen

En pleno siglo XXI la tecnología digital y el internet ha generado satisfacción en muchos aspectos a la vida cotidiana de seres humanos, la presente investigación se enfoca al problema de cómo puede ser aprovechado adicionalmente de su condición el crecimiento de información no estructurada sobre las ofertas laborales y currículum vitae en diferentes sitios web para recomendación de ofertas laborales. Ya que esta información se encuentra escrita en lenguaje humano y por ende las computadoras no pueden interpretar la información en lenguaje natural.

Los sistemas de recomendación necesitan un valor numérico sobre los gustos y disgustos de los usuarios para hacer listas de recomendación basada en colaboración o similitud de contenido. Sin embargo, en la presente investigación la información a considerar son los términos o palabras relevantes que describen al candidato, experiencias y habilidades.

Esta investigación se centra en el proceso de minería de textos, la cual consiste en el proceso de estructuración del modelo de datos partir de textos que describen a candidatos y ofertas laborales, el cual se considera como datos de entrada para los algoritmos de sistema de recomendación. Por tratarse de información textual y en lenguaje humano, se aplican las teorías de técnicas de minería de textos y procesamiento lenguaje natural para el desarrollo del presente trabajo, comenzando desde la recuperación de información hasta la estructuración de minería de datos a partir de datos de currículums vitae y ofertas laborales de sitios web.

Como técnica relevante para el presente estudio se destaca la técnica frecuencia de término – frecuencia inversa de documentos -tf-idf, el cual permite a identificar términos relevantes de currículums vitae con relación a las ofertas laborales de sitios web. La misma que ayuda a determinar la relevancia de los currículums vitae con respecto a las ofertas laborales con un valor numérico a través del promedio de valores tf-idf. De tal manera que el valor ponderado pueda ser usado como valor rating de relevancia de currículums vitae para recomendación de ofertas laborales.

Abtract

In the XXI century digital technology and the internet has generated satisfaction in many aspects to the daily life of human beings, this research focuses on the problem of how can be taken advantage of additionally the growth of unstructured information on job offers and curriculum vitae in different web sites to recommend job offers. Since this information is written in human language and the computers can not interpret the information in natural language.

Recommendation systems need a numerical value on the likes and dislikes of users to make recommendation lists based on collaboration or similarity of content. However, in the present investigation the information to be considered are the relevant terms or words that describe the candidate, experiences and abilities.

This research focuses on the process of text mining, which consists of the process of structuring the data model based on texts that describe candidates and job offers, which is considered as input data for the recommendation system algorithms. Because it is textual information and in human language, the theories of text mining techniques and natural language processing are applied for the development of this work, starting from the information retrieval to the structuring of data mining based on CV and job offers data of websites.

As a relevant technique for the present study we emphasize the technique, frequency term - inverse frequency of documents -tf-idf, which allows to identify relevant terms of CVs in relation to job offers of websites. The same one that helps to determine the relevance of the CVs with respect to job offers with a numerical value through the average of tf-idf values. In such a way that the weighted value tf-

idf can be used as a rating value of curriculums vitae relevance for the recommendation of job offers.

Índice

I. EL PROBLEMA DE INVESTIGACIÓN 15
1.1 Descripción de la realidad problemática 15
1.2 Formulación del problema 20
1.2.1 Problema general 20
1.2.2 Problema especifico 20
1.3 Justificación y la importancia del proyecto de investigación 20
1.4 Objetivos 21
1.4.1 Objetivo general 21
1.4.2 Objetivos específicos 21
1.5 Hipótesis 22
1.5.1 Hipótesis general 22
1.5.2 Hipótesis específicas 22
II. MARCO TEÓRICO 23
2.1 Antecedentes de estudio 23
2.2 Bases teóricas 24
2.2.1 Minería de textos 24
2.2.2 La minería textual y su relación con otras disciplinas 28
2.2.3 Procesamiento Lenguaje Natural 30
2.2.4 Minería de datos 30
2.2.5 Sistemas de recomendación 31
2.3 Definición de términos 33
III. MARCO METODOLÓGICO 35
3.1 Lugar de ejecución 35
3.2 Tipo y diseño 35
3.2.1 Tipo de investigación. 35
3.3 Línea de investigación 35

3.4 Operacionalización de variables 35
3.5 Población y muestra 36
3.5.1 Población 36
3.5.2 Muestra 37
3.6 Técnicas e instrumentos para recolección de datos 37
3.6.1 Técnica 37
3.6.2 Instrumentos 37
3.7 Diseño experimental y análisis de datos. 37
IV. PRESENTACION DE RESULTADOS 39
4.1 Recuperación de información 39
4.2 Procesamiento del lenguaje natural 42
4.2.1 Limpieza y transformación de datos 42
4.2.2 Técnica TF-IDF. 46
4.2.3 Implementación de (TF-IDF) 53
4.2.4 Interpretación del resultado. 65
4.3 Extracción de información 70
4.4 Minería de datos 73
V. CONCLUSIONES 79
5.1 Conclusiones generales 79
5.2 Conclusiones específicas 79
VI. RECOMENDACIONES 80
VII. Referencias 81
VIII. Anexos 89

Índice de tablas

Tabla 1: Diferencias entre Minería de textos, datos y recuperación textual según: (Hearst). 29

Tabla 2: Operacionalización de las variables. 36

Tabla 3: Diseño experimental basada en la metodología de (Hears). 38

Tabla 4: Información recuperada de ofertas laborales desde sitios web. 41

Tabla 5: Información de currículums vitae recuperada desde archivos HTML.... 42

Tabla 6: Datos de las ofertas laborales recuperada. 51

Tabla 7: Información distribuida en documentos de ofertas laborales en orden alfabético en la programación del algoritmo de TFIDF 54

Tabla 8: Información distribuida en documentos de currículums vitae en orden alfabético en la programación del algoritmo de TFIDF 55

Tabla 9: Ejemplo de estructuración relevancia de términos del primer CV respecto a la primera oferta laboral. 69

Tabla 10: Promedios de valores relevantes de TF-IDF para respectivos documentos. 71

Tabla 11: Promedio general de TF-IDF sobre los valores TFIDF de respectivos documentos como se muestra en la tabla anterior. 72

Tabla 12: Resumen de la matriz de la Tabla 11. 73

Tabla 13: Matriz de rating de los CV respecto a las ofertas laborales. 74

Tabla 14: Modelo de datos detallado con rating de currículums vitae respecto a las ofertas laborales 78

Índice de figuras

Figura 1: Reporte de (Internet World Stats, 2018) sobre el uso de internet en distintos continentes del mundo. 15

Figura 2: Ejemplo de categorización según (Brun & Senso, 2004) p.6 26

Figura 3: Una representación gráfica sobre procesos de minería de textos en libro de (Kwarter, Text Mining in practice with R, 2010) 28

Figura 4: Ejemplo de calcificaciones de intervalo de 5 puntos en el libro de: (Aggarwal, 2016). 33

Figura 5: Código de programación con el marco de trabajo Scrapy para recuperación de información de ofertas laborales de sitios web. 40

Figura 6: Fragmento del código de programación para recuperación de información de currículums vitae 40

Figura 7: Código de programación de NLTK para limpieza de datos de currículums vitae y ofertas laborales. 43

Figura 8: Resultado de limpieza de datos con NLTK, listas de términos sin ruido y caracteres impertinentes. 44

Figura 9: Términos con sus respectivas frecuencias en un CV. 45

Figura 10: Nube de palabras según su valor de frecuencia. 45

Figura 11: Información de la descripción del candidato en su currículum vitae en formato HTML. 49

Figura 12: Información recuperada sobre la experiencia laboral del candidato en los documentos HTML. 50

Figura 13: Información recuperada sobre habilidades del candidato en la los documentos HTML recolectadas. 50

Figura 14: Datos de entrada para if-idf, vectores que almacenan términos de currículums vitae y ofertas laborales. 56
Figura 15: Primera parte de diagrama de flujo de tf-idf configurado para la presente investigación. 57
Figura 16: Segunda parte del diagrama de flujo de tf-idf configurado para la presente investigación. 60
Figura 17: Diagrama flujo completa de tf-idf. 61
Figura 18: Resultado de procesamiento de datos de currículums vitae y ofertas laborales con TF-IDF. 64
Figura 19: Muestra de información de ofertas laborales en bruto. 66
Figura 20: Muestra de información en bruto de los currículums vitae. 66
Figura 21: Muestra del resultado de procesamiento con NLTK las ofertas laborales. 66
Figura 22: Muestra del resultado de procesamiento con NLTK los currículums vitae. 67
Figura 23: Muestra de términos relevantes del primer CV en la primera oferta laboral. 67
Figura 24: Ilustración de la relevancia del primer CV respecto a varias ofertas laborales. 75
Figura 25: Ilustración grafica de relevancia de dos CV (1 y 2) con respecto a varias ofertas laborales. 76

Prólogo:

CV = Currículum Vitae.

TM = Text mining (Minería de Textos).

NLP = Natural language processing (Procesamiento Lenguaje Natural).

SROL = Sistema de recomendación de Ofertas Laborales.

LSI = Latent Semantic Indexing (Indización Semántica Latente).

SR = Sistema de Recomendación.

NLTK = Natural Language Toolkit (Kit de Herramientas de Lenguaje Natural).

Feedback = Retro alimentación.

TF-IDF = Term Frequency–Inverse Document Frequency (Frecuencia de Términos- Frecuencia Inversa de Documentos).

ASCII = Abbreviated from American Standard Code for Information Interchange. (Abreviado del Código Estándar Americano para el Intercambio de Información).

R = Lenguaje de programación R.

Términos = Palabras.

O * NET. = El Programa Principal Fuente de Información Ocupacional de EEUU.

Rating = Una clasificación según grado.

API = Application Programming Interface (Interfaz de Programación de Aplicaciones).

HTML= Hypertext Markup Language (Lenguaje de Marcado de Hipertexto).

Introducción

El principal problema que pretende resolver la presente investigación es como procesar la abundante información de currículums vitae y ofertas laborales que se encuentran en sitios web, y que esta pueda ser usada adicionalmente de su condición en otros casos, como es el presente caso que se enfoca a sistema de recomendación de ofertas laborales. Porque los sistemas de recomendación necesitan de una información estructurada de los gustos y disgustos de los usuarios sobre algunos elementos como: los productos, películas, libros, etc. en el presente caso se trata de usuarios que disponen su información sobre sus habilidades, conocimientos y experiencias en sus currículums vitae y las cuales escritas en forma textual y en lenguaje humano. De la disposición de esta información masiva se busca valorizar con un valor numérico los CV con respecto a las ofertas laborales, puesto que esta información numérica serviría como valor rating para los algoritmos de sistema de recomendación.

El objetivo de la presente investigación es abordar una forma de tratamiento de información, según el estado de arte de la investigación se encuentra una variedad de teorías de minería de textos y procesamiento lenguaje natural para tratar información textual, como es el presente caso. El principal objetivo es lograr un modelo minería de datos para recomendación de ofertas laborales basada en términos relevantes, un valor numérico que represente la relevancia de los currículums vitae con respecto a las ofertas laborales, y estos datos puedan ser usada en la aplicación de los algoritmos de SR para identificar sus similares, ya sea basada en contenido, colaborativo u otros.

La relevancia de términos juega un rol muy importante en el presente, porque en los CV se encuentra una buena cantidad de términos o palabras escritas en lenguaje humano, esta información representa a los conocimientos, habilidades y experiencias del candidato, el peso de estos términos nos conduce a saber que tan relevante son con respecto a las ofertas laborales.

I. EL PROBLEMA DE INVESTIGACIÓN

1.1 Descripción de la realidad problemática

El internet y la tecnología web en los últimos años ha permitido a millones de usuarios interactuar en distintas plataformas, como se puede apreciar en el reporte de (Internet World Stats, 2018) sobre la intensidad de uso del internet en el mundo, en la figura N° 1 se puede apreciar la intensidad de uso clasificado por regiones.

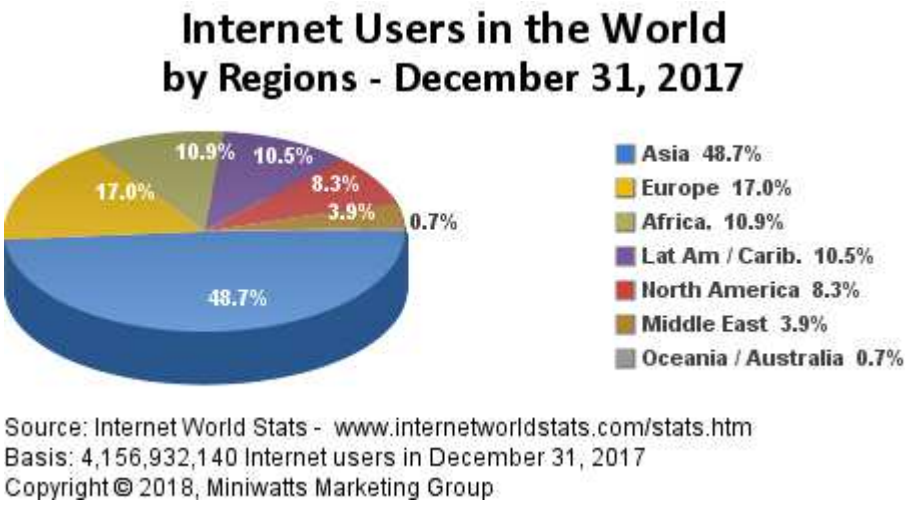


Figura 1: Reporte de (Internet World Stats, 2018) sobre el uso de internet en distintos continentes del mundo.

En el ámbito de manda y oferta de empleo ha conglomerado importante información masiva no estructurada en diversos sitios web y bases de datos, el presente trabajo se enfoca a datos exclusivas de sitios web de bolsas de trabajo, por ejemplo: Linkedin, Computrabajo, Aptitus, Bumeran, etc. En donde que se publican ofertas laborales para los profesionales en distintas áreas con contenido de información importante

como: especificaciones técnicas, conocimientos, habilidades, experiencias, etc. Además, como su servicio ofrece registro gratuito de CV a los profesionales o candidatos.

Para un sistema de recomendación es elemental la información sobre gustos y disgustos de usuarios para hacer listas de recomendaciones.

> Necesario recopilar datos sobre los gustos y aversiones del usuario para el proceso de recomendación. La recopilación de datos se realiza durante la fase fuera de línea, mientras que las recomendaciones se determinan durante la fase en línea cuando un usuario específico interactúa con el sistema. (…) Durante la fase en línea, las preferencias propias del usuario se combinan con el contenido para crear las predicciones.
>
> (Aggarwal, 2016, pág. 146)

Lo que nos lleva a plantear el presente proyecto con el objetivo de buscar la forma de transformar el contenido de términos de los currículums vitae y ofertas laborales en un valor representativo.

En un aplicativo de sistema de recomendación existen dos tipos de datos de entrada, datos de los usuarios y de los elementos o artículos a recomendar.

> Los usuarios tienen referencias para ciertos artículos, y estas preferencias deben ser extraídas de los datos. Los datos en sí se representan como una matriz de utilidad, que proporciona a cada uno de los elementos de usuario, un valor que representa lo que se conoce

> sobre el grado de preferencia de ese usuario para ese elemento. Los valores provienen de un conjunto ordenado, por ejemplo, enteros del 1 al 5 que representan la cantidad de estrellas que el usuario dio como calificación para ese elemento. Suponemos que la matriz es escasa, lo que significa que la mayoría de las entradas son "desconocidas". Una calificación desconocida implica que no tenemos información explícita sobre la preferencia del usuario por el artículo.
>
> (Rajaraman & Ullman, 2014, pág. 2)

Lo que nos conduce a preguntarnos en el ámbito de demanda y oferta laboral, ¿Cómo obtener el valor rating o preferencias de los currículums vitae con respecto a las ofertas laborales? para poder hacer recomendación de ofertas laborales en base a los términos escritos que referencian a sus habilidades, conocimientos y experiencias del candidato.

Sin embargo, en otras investigaciones en línea de minería de textos y procesamiento lenguaje natural, se abordan técnicas de procesamiento de información textual y escrito en lenguaje humano. En la investigación de (Allahyari, y otros, 2017) resume:

> Inmensa cantidad de datos generados por diversos sitios web, sistemas de información u otros medios de comunicación, llama la atención al uso de métodos de minería de textos y otros relacionados para descubrir conocimientos. La información textual en lenguaje humano es difícil de entender por las computadoras, por lo que hay

> necesidad de diseñar algoritmos para procesar la inmensidad de información. (pág.1)

El presente estudio precisamente trata de los documentos como currículums vitae y ofertas laborales que están escritos de forma textual y en lenguaje humano, el cual es difícil de entender para las computadoras. Sin embargo, en su investigación de (Torre, 2017) define:

> La minería de textos surge ante la necesidad de dar valor adicional a la gran cantidad de documentos provenientes de diferentes fuentes como: email, intranets, sistema de trámite documentario, etc. Los cuales pueden ser almacenados en forma de textual en diferentes bases de datos en las empresas. (pág. 2)

El escenario de la presente investigación se asemeja al caso anterior, por tratarse de información textual acumulada en las bases de datos de las empresas que prestan servicios de búsqueda de perfiles profesionales y ofertas laborales.

Por otra parte, en la investigación sobre el enfoque de minería de datos para monitorear los requisitos del mercado laboral de (Karakatsanis, y otros, 2016) sugiere:

> Un enfoque basado en la minería de datos para identificar las ocupaciones más demandadas en el mercado laboral moderno. Para lograr esto, desarrolló un modelo de indexación semántica latente que es capaz de hacer coincidir el anuncio de trabajo extraído de la

web con los datos de la descripción de la ocupación en la base de datos O * NET. (pág.1)

En el presente trabajo estudia el problema de abundancia de data sobre currículum vitae y ofertas de puestos laborales en bolsas de trabajo, la cuales escritas en lenguaje humano que las computadoras por si solas no pueden interpretar. Hasta esta era del avance de la tecnología tenemos una variedad de teorías que pretenden resolver el problema de tratamiento de la información no estructurada y convertir en una data estructurada para que sea entendible para computadoras y seres humanos. Pero, principalmente en la presente se enfoca a recomendación de ofertas laborales, se busca obtener un valor rating de los currículums vitae y ofertas laborales en base al contenido de términos en cada uno de estos documentos aplicando técnicas de minería de textos. Además, se puede dar otro uso adicional de su condición, en otras palabras, aprovechar la abundante data para construir modelos para su uso posterior con algoritmos inteligentes como: sistema de recomendación, reconocimiento de patrones, clasificar la empleabilidad de profesionales, etc.

En resumen, el presente trabajo aborda sobre cómo se puede construir un modelo de minería de datos aplicando técnicas minería de textos en los términos escritos en los documentos de currículum vitae y oferta de puestos laborales, y que esta pueda ser usado en recomendación de ofertas laborales.

1.2 Formulación del problema

1.2.1 Problema general

¿Cómo estructurar un modelo de minería de datos usando técnicas de minería de textos para recomendación de ofertas laborales a partir de currículum vitae y ofertas laborales de sitios web?

1.2.2 Problema especifico

¿Cómo aplicar técnicas de minería de textos en el procesamiento de los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales?

1.3 Justificación y la importancia del proyecto de investigación

En los últimos años ha incrementado considerablemente las ofertas laborales en diferentes páginas web de bolsas de trabajo, la búsqueda de empleo se efectuó en internet a través de redes sociales, páginas de web, enviando currículum vitae digitales a diferentes empresas, esto ha generado una data no estructurada muy importante de la que se puede aprovechar para hacer recomendación de ofertas laborales basada en términos relevantes de currículums vitae para beneficiar a las compañías reclutadores de talentos y buscadores de empleo.

Por otro lado, la mayoría de las páginas web de bolsas de trabajo carecen de una recomendación de ofertas laborales de manera más inteligente y la sugerencia de candidatos a los empleadores. También se busca contribuir a la comunidad científica a confirmar o negar la aplicabilidad de las teorías de técnicas de minería de textos para este tipo de datos en el ámbito de inserción laboral.

Por último, el trabajo está basado en las recientes investigaciones en línea de sistema de recomendación, minería de textos, minería de datos y procesamiento lenguaje natural, que busca construir y proponer un modelo de minería de datos para recomendación de ofertas laborales a partir de texto no estructurado de currículum vitae y ofertas laborales en portales web de bolsa de trabajo.

1.4 Objetivos

1.4.1 Objetivo general

Proponer un modelo de minería de datos a partir de currículum vitae y ofertas laborales de sitios web para recomendación de ofertas laborales.

1.4.2 Objetivos específicos

Aplicar las técnicas de minería de textos para procesar los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales.

1.5 Hipótesis

1.5.1 Hipótesis general

Sí, es posible construir un modelo minería de datos con técnicas minería de datos para recomendación de ofertas laborales a partir de currículum vitae y ofertas laborales de sitios web.

1.5.2 Hipótesis específicas

Es eficiente la aplicabilidad de técnicas de minería de textos para procesamiento de los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales.

II. MARCO TEÓRICO

2.1 Antecedentes de estudio

A continuación, se precisa algunas investigaciones que abordan procesamiento de información de oferta y demanda laboral con técnicas de minería de textos y procesamiento lenguaje natural. En estudio realizado por (Gao & Eldin, 2014) se procesó "Más de 20,000 anuncios de trabajo de diversos sitios web se procesó con método de minería de textos para identificar habilidades y experiencias de la información derivada de las páginas web del sector industria de construcción" (pág.1).

Es importante considerar las habilidades y conocimientos de los candidatos para identificar la oferta laboral coherente en base a sus características profesionales, según (Kinoa, Kurokia, Machidab, Furuyab, & Takanob, 2017):

> Cada usuario tiene sus propias personalidades favorables y desfavorables, la correspondencia laboral por parte de empresas de reclutamiento y candidatos es importante. En un contexto de análisis de datos que incluyen el tiempo de viaje, la ubicación del trabajo, tipo de trabajo, tarifas, conjunto de habilidades de candidato, etc. Y al aplicar palabras clave en un proceso de aprendizaje automático utilizando herramientas de minería de textos, como resultado se descubre palabras claves efectivas para un sistema de coincidencia de trabajos. (pág.1)

La identificación de conocimientos de la data no estructura fue demostrada en la investigación de (Almada, Elias, Gómez, Mendoza, & López, 2017) en donde "se logra mediante el PLN y TM analizar el texto no estructurado de las hojas de vida y ofertas laborales, proponen una forma de utilizar NLP y TM para identificar los perfiles de conocimiento para las puestos de ingeniería de software" (pág.10)

La aplicación de una de las técnicas de minería de textos para procesamiento de información de las ofertas y demandas laborales, se destaca en la investigación de (Malherbe, Diaby, Cataldi, Viennet, & Aufaure, 2014) presenta:

> Un método novedoso para formalizar el contenido textual de las ofertas de trabajo que tiene como objetivo identificar la información y los campos más relevantes expresados por ellos y aprovechar esta formalización compacta para la recomendación de trabajo y la coincidencia de perfiles en entornos de redes sociales (pág. 1).

2.2 Bases teóricas

2.2.1 Minería de textos

Existen diversas definiciones sobre minería de textos, entre de los cuales a continuación se citan algunos: Según (Kwarter, 2010) "existen varias definiciones como: La minería de texto es el proceso de destilación de información procesable del texto" (pág. 12). Mientras para (Kao & Poteet, 2007) la minería de textos "es el descubrimiento y la extracción de

conocimiento interesante y no trivial del texto libre o no estructurado" (pág. 12). La minería textual es una aplicación de la lingüística computacional y del procesamiento de textos que pretende facilitar la identificación y extracción de nuevo conocimiento a partir de colecciones de documentos o corpus textuales. (Brun & Senso, 2004, pág. 1). "La minería de textos es un campo nuevo y creciente que intenta recopilar información significativa de texto en lenguaje natural. Puede caracterizarse libremente como el proceso de análisis de texto para extraer información que es útil para propósitos particulares" (Witten, pág. 1).

Entonces, minería de textos se puede definir como un campo reciente en análisis de texto y extracción de conocimiento y/o información significativa desde un conjunto de data no estructurada generada en forma textual y que es difícil interpretación para los computadores.

La utilidad de minería de textos se da en diversos aspectos en la vida cotidiana, por ejemplo: Según (Kwarter, 2010) "la extracción de texto representa la capacidad de tomar grandes cantidades de datos no estructurados y extraer rápidamente conocimientos útiles y novedosos que pueden ser analizadas" (pág. 20).

Según (Zanini & Dhawan, 2015), se distingue cuatro etapas de minería de textos:

- **Recuperación de información**

En esta etapa de TM se define por identificación de documentos relevantes de un corpus. Es Identificar "hechos" y datos puntuales de los

documentos, a lo que nos referiremos con el término inglés feature extraction (Brun & Senso, 2004, pág. 4).

- **Procesamiento del lenguaje natural**

Hay varias definiciones de procesamiento lenguaje natural, entre los cuales el (Zanini & Dhawan, 2015) definen como: “tras haber recuperado un conjunto de documentos textuales, hay necesidad de establecer un formato único de entrada para que el computador pueda interpretar al momento de procesar y analizar el texto escrito en lenguaje humano” (pág. 2).

Mientras para (Brun & Senso, 2004), se trata de agrupación de documentos similares o clustering. Uno de los casos que ejemplifica es la categorización de automática de documentos de un conjunto de clases predefinidas por humanos, como se puede apreciar en la figura N° 2.

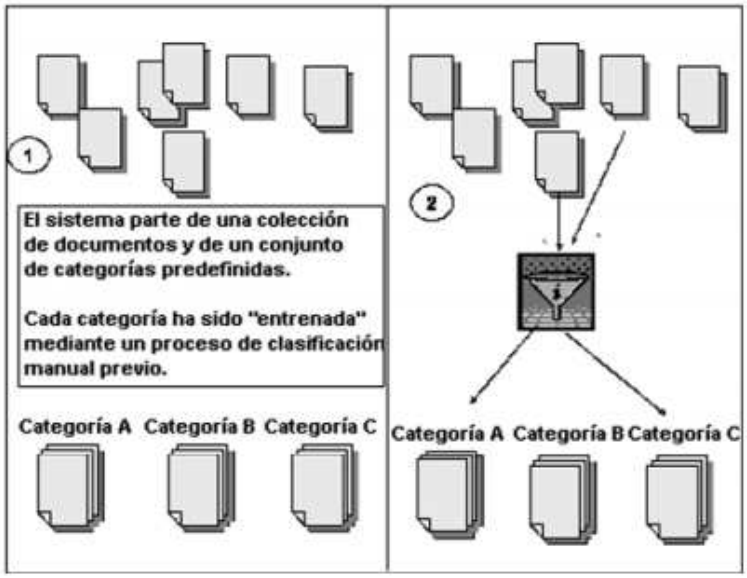


Figura 2: Ejemplo de categorización de documentos según (Brun & Senso, 2004, pág. 6)

PLN se enfoca en la comprensión el lenguaje humano mediante ordenador, interpretan y hasta dar un significado para que estos puedan ser usados de manera más práctica.

- **Extracción de información**

Según (Zanini & Dhawan, 2015) “La tarea más común realizada durante esta etapa es la identificación de términos lingüísticos para convertir en documentos estructurados, que pueden consistir en una o más palabras” (pág. 3). La extracción de información también es “basado en la frecuencia estadística de los términos y en la ponderación de la importancia de las frases y la posición que estas ocupan en el documento” (Brun & Senso, 2004, pág. 7).

- **Minería de datos**

Según (Zanini & Dhawan, 2015) "Minería" es un sinónimo de "análisis", ya que el objetivo es extraer información útil de los datos de texto para crear nuevos conocimientos”. La minería de datos también consiste en “datos en un formato que haga posible su interpretación y permita al usuario moverse con facilidad entre los distintos textos analizados” (Brun & Senso, 2004, pág. 7).

Es una etapa en donde se busca de la información encontrar una estructura de datos y se pueda aplicar métodos estadísticos, matemáticos, etc. Técnicas de estándar para construir y/o extraer conocimientos para la toma de decisiones y análisis avanzada sobre los datos.

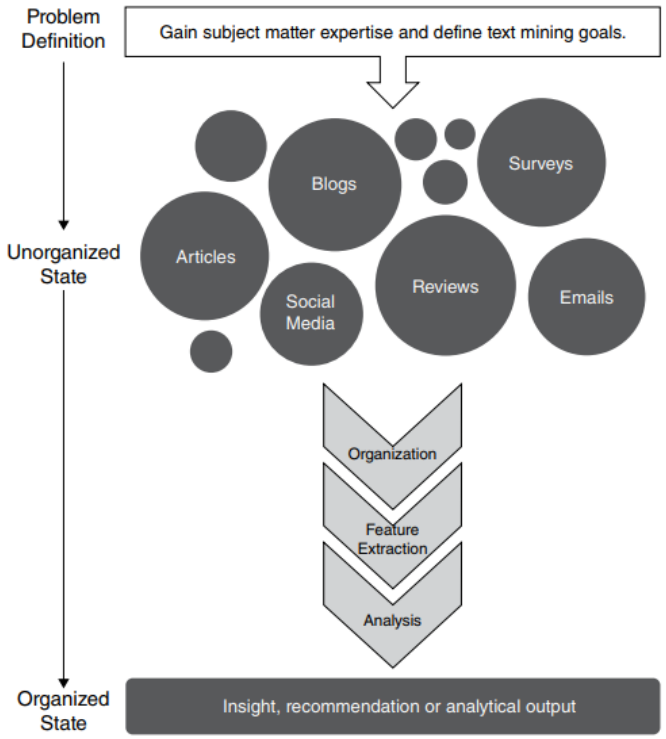


Figura 3: Una representación gráfica sobre procesos de minería de textos en libro de (Kwarter, 2010)

2.2.2 La minería textual y su relación con otras disciplinas

Entre diversas disciplinas similares "existe una clara relación entre minería textual, minería de datos, recuperación de información y lingüística computacional" (Brun & Senso, 2004, pág. 3).

Minería de datos busca el análisis automático o semiautomático de grandes cantidades de datos para identificar patrones desconocidos hasta ahora. En cambio, minería de textos busca extraer conocimiento de un conjunto de datos no estructurados.

Lingüística computacional y recuperación de información comparten técnicas de recuperación textual. Según (Brun & Senso, 2004) existe:

> La diferencia entre minería textual y recuperación de información se encuentra en que el objetivo, de ésta última es identificar los documentos relevantes para un usuario dentro de una colección. La recuperación textual parte de una representación formal de los documentos sobre los que se realizará la búsqueda, y de la formulación de las necesidades de información del usuario mediante un sistema de representación equivalente. Sin embargo, la recuperación textual no pretende facilitar el proceso de análisis ni la extracción de nuevos conocimientos, como sí pretende la minería textual (pág. 4).

Tabla 1: Diferencias entre Minería de textos, datos y recuperación textual según: (Hearst).

	Búsqueda de patrones	Búsqueda de ítems y datos	
		Nuevos	Conocidos
Datos no textuales	Minerías de datos	?	Búsqueda en BB.DD.
Datos textuales	Lingüística computacional	Minería de textos	Recuperación textual

2.2.3 Procesamiento Lenguaje Natural

Es un campo de ciencia de la computación, inteligencia artificial y lingüística computacional que estudia la interacción entre humano y computador, su función es que las computadoras entiendan, analicen y generen el lenguaje natural. Esto es posible gracias a diferentes técnicas que permiten la automatización de diferentes formas de comunicación natural, así como el habla, escrita, lectura, traducción, etc. Según (Cortez, Vega, & Pariona, 2009), sus principales aplicaciones son muy variadas porque tienen un alcance muy grande y entre ellos se menciona algunas a continuación:

- Traducción automática
- Recuperación de la información
- Extracción de información y resúmenes
- Resolución cooperativa de problemas
- Tutores inteligentes
- Reconocimiento de voz

2.2.4 Minería de datos

Es uno de los métodos de la inteligencia artificial basada en estadísticas y ciencias de la computación para explorar grandes cantidades de datos para descubrir patrones de comportamiento.

Es un campo de Ciencia de la Computación encargada de descubrir patrones en grandes volúmenes de conjuntos de datos, gracias a las

técnicas de inteligencia artificial, procesamiento lenguaje natural, machine learning, etc. su principal objetivo consiste en extraer información relevante de un conjunto de datos y transformarla en la estructura comprensible para el uso posterior.

- Selección de conjunto datos.
- Análisis de las propiedades de los datos.
- Transformación del conjunto de datos de entrada.
- Seleccionar y aplicar la técnica de minería de datos.
- Extracción de conocimiento.
- Interpretación y evaluación de datos.

(Cedano, 2015, pág. 24).

2.2.5 Sistemas de recomendación

Sistema de recomendación se clasifica como un sistema inteligente que hace sugerencias a sus usuarios sobre un determinado producto o elemento, la recomendación se basa en un estudio previo sobre los datos históricos de los clientes, sobre sus características, necesidades, actividades, etc. A través de un procesamiento de datos determina ítems de interés para el cliente y posteriormente genera una lista de recomendaciones posibles de interés del usuario.

La importancia de sistemas de recomendación radica en medio de negocios online, donde comparten información entre los clientes y la compañía de negocios, por ejemplo, Netflix tiene interacción con sus clientes por medio de sus diversas aplicaciones online, espacio propicio

para realizar feedback sobre los gustos y disgustos de sus clientes con respecto a las películas que oferta la compañía, la forma de recopilar información es por medio de valoraciones de satisfacción que los usuario manifiestan a través de ratings, se da cuando el usuario ratifica sus gustos dando una calificación a la película con un número de un rango de 1 – 5 de estrellas, como se aprecia en la figura N° 4. Los ratings son almacenados en sus bases de datos y procesado para hacer recomendaciones de películas similares a las que el usuario haya puntuado.

Existen en la actualidad diferentes formas de mantener feedback con los clientes, por ejemplo, Amazon, AlieExpres, etc. Entre las compañías de venta online mantienen feedback de una forma distinta que Netflix, un cliente completa su satisfacción de algún producto con concretar la adquisición, de tal manera que la puntuación no es numérica o rating, entonces acuden a analizar características de los productos que el cliente adquiere, la frecuencia con que adquiere, etc. A partir de esto se puede identificar distintas formas de obtener feedback y tipos de información que se recopila en forma explícita e implícita.

- Información implícita

Es cuando la información de las características, necesidades, etc. del usuario no es clara, por ejemplo, cuando se recopila el número de producciones de una canción, consultas sobre características de productos, las noticias que le interesa diariamente, el tipo de películas que ha visto,

etc. Es información que no expresa algo concreto sobre algún elemento el usuario.

- Información explicita

Es la información que de alguna manera u otra que proporciona el usuario es concreta, donde la información podría ser numérica, por ejemplo, cuando el usuario califica con número su grado de agrado de una determinada película, producto, como se puede ver en la figura N° 4. Es la información proporcionada directamente por el usuario y hay plena credibilidad de la información.



Figura 4: Ejemplo de calcificaciones de intervalo de 5 puntos en el libro de: (Aggarwal, 2016).

2.3 Definición de términos

CV = Currículum Vitae.

TM = Text mining (Minería de texto).

NLP = Natural language processing (Procesamiento lenguaje Natural).

SROL = Sistema de recomendación de Ofertas Laborales.

LSI = Latent Semantic Indexing (Indización semántica latente).

SR = Sistema de Recomendación.

NLTK = Natural Language Toolkit (Kit de herramientas de lenguaje natural).

Feedback = Retro alimentación.

TF-IDF = term frequency–inverse document frequency (Frecuencia de términos- frecuencia inversa de documentos).

ASCII = abbreviated from American Standard Code for Information Interchange (abreviado del Código Estándar Americano para el Intercambio de Información).

Términos = Palabras.

O * NET. = El Programa principal fuente de información Ocupacional de EEUU.

Rating = Una clasificación según grado.

API = Application programming interface (Interfaz de programación de aplicaciones).

HTML= Hypertext Markup Language (Lenguaje de marcado de hipertexto).

III. MARCO METODOLÓGICO

3.1 Lugar de ejecución

- País: Perú
- Región: Moquegua
- Provincia: Ilo
- Distrito: Pacocha
- Lugar: Universidad Nacional de Moquegua filial Ilo.

3.2 Tipo y diseño

3.2.1 Tipo de investigación.

La presente investigación es de tipo aplicada.

3.3 Línea de investigación

Inteligencia Artificial, línea de investigación definida en la Escuela Profesional de Ingeniería de Sistemas e Informática de la Universidad Nacional de Moquegua.

3.4 Operacionalización de variables

La presente investigación de tipo aplicada tiene orientación, tal como lo refirió el científico Keit Satanovich, “el principal objetivo de la investigación aplicada es predecir el comportamiento específico de una determinada configuración” (Satanovich, 2007, pág. 106), enfocando a resolver problemas de vida cotidiana, proyectando e idealizando la mejor

forma de aplicar los conocimiento teóricos, con el objetivo de mejorar la vida cotidiana. Por lo tanto, como variable independiente se considera minería de textos por lo que tiene variedad teorías y principalmente técnicas de procesamiento de información. Y por otro lado se considera como variable dependiente modelo de datos para recomendación de ofertas laborales.

Tabla 2: Operacionalización de las variables.

Variable	Tipo	Indicador	Escala de medición
Minería de textos.	Independiente.		
Modelo de minería de datos para recomendación de ofertas laborales.	Dependiente.	Rating (valor de relevancia)	Relevancia de términos.

3.5 Población y muestra

3.5.1 Población

En la presente investigación la población consta de:

- 10,000 ofertas laborales.
- 10,000 currículums vitae.

3.5.2 Muestra

En la presente investigación se considera como muestra lo siguiente:

- 10,000 ofertas laborales a nivel nacional.
- 5,000 currículums vitae a nivel nacional.

3.6 Técnicas e instrumentos para recolección de datos

3.6.1 Técnica

Scrapy.

3.6.2 Instrumentos

Técnicas de recuperación de información.

3.7 Diseño experimental, métodos para la presentación y análisis de datos.

El presente trabajo está basado en articulo (Zanini & Dhawan, 2015) y propiamente presentado en la parte marco conceptual.

- Recuperación de información
- Procesamiento del lenguaje natural (NLP)
- Extracción de información
- Minería de datos

Son etapas que conducirá a cumplir los objetivos del presente trabajo, que es construir de un modelo minería de textos para recomendación de ofertas laborales.

Tabla 3: Diseño experimental basada en la metodología de (Hears).

Cuadro de diseño experimental			
Recuperación de información.	Procesamiento del lenguaje natural (NLP).	Extracción de información.	Minería de datos.
Recolección de información de currículum vitae y ofertas laborales de web.	Procesamiento de datos con técnicas de minería de datos y procesamiento lenguaje natural.	Extracción de información relevante para la propuesta del modelo de minera de datos.	Data estructurada para recomendación de ofertas laborales.

IV. PRESENTACION DE RESULTADOS

A Continuación, se sigue la metodología planteado por (Zanini & Dhawan, 2015) para desarrollar el presente proyecto, se sigue los cuatro etapas de minería de textos planteadas por el autor.

4.1 Recuperación de información

Se implementó técnicas de recuperación de información en lenguaje de programación Python, la herramienta de trabajo scrapy es un marco de python que permitió recolectar información de ofertas laborales y currículums vitae de candidatos para fines de investigación.

Scrapy es una herramienta para rastreo de información de sitios web y de código abierto escrito en lenguaje python, también se puede utilizar para extraer datos mediante API o como rastreador web de propósito general. Nos permite acceder a etiquetas de HTML, donde se encuentra la información sobre las ofertas laborales y sus especificaciones técnicas de requerimiento, de tal manera que se pudo acceder a múltiples páginas web de nuestro interés y descargar información pública para el presente estudio. Mientras que los datos de curricúlums vitae se obtuvo mediante gestiones internas.



```
import scrapy
from scrapy.spider import CrawlSpider, Rule
from scrapy.linkextractors import LinkExtractor
from scrapy.exceptions import CloseSpider
from mercado.items import MercadoItem
class MercadoSpider(CrawlSpider):
    name = 'mercado'
    item_count = 0
    allowed_domain = ['www.computrabajo.com.pe']
    start_urls = ['http                    .pe/ofertas-de-trabajo/']

    rules = {
        # Para cada item
        Rule(LinkExtractor(allow = (), restrict_xpaths = ('//li[@class="siguiente"]/a'))),
        Rule(LinkExtractor(allow =(), restrict_xpaths = ('//h2[@class="tO"]/a')),
                           callback = 'parse_item', follow = False)
    }

    def parse_item(self, response):
        ml_item = MercadoItem()
        #info de producto
        ml_item['titulo'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/header/h1/text()').extract()
        ml_item['empresa'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/div[1]/div[2]/h2/text()').extract()
        ml_item['ciudad'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/header/p/span[1]/text()').extract()
        ml_item['descripcion'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[1]/text()').extract()
        ml_item['requisitos'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[1]/text()').extract()
        ml_item['funciones'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[5]/text()').extract()
        ml_item['educacion'] = response.xpath('//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[4]/text()').extract()
        ml_item['conocimiento'] = response.xpath('//*[@id="MainContainer"]/article/section[4]/ul/li[5]/p/text()').extract()

        self.item_count += 1
        if self.item_count > 10000:
            raise CloseSpider('item_exceeded')
        yield ml_item
```

Figura 5: Código de programación con el marco de trabajo Scrapy para recuperación de información de ofertas laborales de sitios web.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from bs4 import BeautifulSoup
l= "/media/honorio/Honorio LNX/AAA/Date Base/CANDIDATEVCS/CV"
w = ".html"
UserCV=[]
for i in range(20000):
    datauser=[]
    files=("{}{}{}".format(l,i,w))
    #print files
    html_file=open(files,'r')
    page=html_file.read()
    soup=BeautifulSoup(page,'html.parser')
    reviews=soup.find_all('br')
    #print reviews
    for p in reviews:
        datauser.append(p.get_text())
        #print "============================",datauser

    UserCV.append(datauser)
    #print ":::::::::::::::::::::::::::::::::::::::::::::::::::",UserCV
```

Figura 6: Fragmento del código de programación para recuperación de información de currículums vitae.

*El código de programación completa de recuperación de información se encuentra en los anexos.

Resultado de recolección de datos:

Data de ofertas laborales. – Se recolectó 10,000 ofertas laborales, en los cuales se encuentra la siguiente información: El título, ciudad, nombre de la empresa, descripción, requisitos, funciones y conocimientos de la oferta laboral, en estos campos se encuentra un conjunto de términos que describe a las necesidades y especificaciones técnicas de la oferta laboral, esta información nos interesa para saber la relevancia de currículums vitae con respecto a las ofertas laborales.

En siguiente tabla N° 4 se puede observar la información recolectada de currículums vitae semiestructurada en campos específicos.

Tabla 4: Información recuperada de ofertas laborales desde sitios web.

Id_Job	Titulo	Empresa	Ciudad	Descripcion	Requisitos	Funciones	Educacion	Conocimientos
0	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
1	Vendedores rotativos para tiendas Baby Infanti...	SGA Consultores SAC	Miraflores Lima	Por encargo de las Tiendas Baby Infanti y Lego...	Secundaria completa. Manejo fluido a nivel usu...		Educación Secundaria	
2	Asistente Contable Técnico o Universitario Egr...	Adecco Perú S.A.	San Borja Lima	Por encargo de nuestro cliente empresa líder ...		Análisis de depreciaciones. Asientos contables...	Técnico	Microsoft Excel
3	Practicante de Administración (Apoyo)	Industrias Plasticas Reunidas S.A.C	Lima Lima	Industrias Plásticas Reunidas es una empresa 1...		Atención directa a gerencia en relación a coor...	Técnico	Microsoft Excel

Data de currículum vitae de candidatos. – Se gestionó más de 360.000 currículums vitae de candidatos en formato HTML, estos archivos tienen el siguiente contenido: Nombre del candidato, descripción, habilidades y experiencia de los candidatos. La misma que se puede apreciar en la siguiente tabla:

Tabla 5: Información de currículums vitae recuperada desde archivos HTML.

Id_User	Nombre	Descripcion	Experiencia	Habilidades
0	Miguel Angel Garcia del Castillo	Soy un profesional capacitado para llevar adel...	publicaciones sobre sistema de recomendacion y...	python, pln, nltk, scrapy, tec
1		Cursante del IV ciclo de Administración de Neg...		Asociación Civil Sol y Mar / Club Sol y Mar ...
2		Soy una joven con grandes expectativas, respon...	CONCAR S.A. Grupo Graña y Montero Desde Mayo 2...	Concar s.a. Grupo Graña y Montero / Informátic...
3		Abogado con cuatro años de experiencia profesi...	PerfilSoy una persona 100% creativa actitud pr...	ESTUDIO JURÍDICO ROMERO / Legal / Asesoría / L...
4		En ciencias de la comunicaciones cursando el 4...		Delosi / Atención a clientes / Venta al consum...

**Los nombres de los candidatos se censuran por tratarse de información real y sensible por cuestiones de seguridad.*

4.2 Procesamiento del lenguaje natural

4.2.1 Limpieza y transformación de datos

Para la limpieza y transformación de datos se aplicó el marco de trabajo NLTK, configurado con librerías como: BeautifulSoup, word_tokenize, sent_tokenize, LancasterStemmer, WordNetLemmatizer, etc. El NLTK es un parco de trabajo que suprime caracteres que no corresponden al ASCII, convertir todos los caracteres a minúsculas, remover puntuación y acentos, omisión de enteros en la lista de términos y lematizacion de verbos en lista de palabras en token, para que todos los términos sean uniformes.

Natural Language Toolkit (NLTK) en general es un conjunto de herramientas que ayuda a procesar información escrito en lenguaje humano y que los computadores no pueden interpretar por sí solas.

La información de entrada está estructurada en dos listas generales, la primera corresponde a todos los currículums vitae y la segunda lista corresponde a todas ofertas laborales, es decir la cantidad de elementos de las listas generales es igual al número de CV y ofertas laborales respectivamente, como se explica en la figura N° 14, las cuales pasan sin excepción por la técnica NLTK.

```
def CleanWOrds(words,user,Clean_Words):

    def remove_non_ascii(words):
        #"""Elimine los caracteres que no son ASCII de la lista de palabras con tok
        new_words = []
        for word in words:
            new_word = unicodedata.normalize(
                'NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore')
            new_words.append(new_word)
        return new_words
    def to_lowercase(words):
    #"""Convierte todos los caracteres a minúsculas de la lista de palabras tokeniz
        new_words = []
        for word in words:
            new_word = word.lower()
            new_words.append(new_word)
        return new_words

    def remove_punctuation(words):
        #"""Eliminar puntuación de la lista de palabras tokenizadas"""
        new_words = []
        for word in words:
            new_word = re.sub(r'[^\w\s]', '', word)
            if new_word != '':
```

Figura 7: Código de programación de NLTK para limpieza de datos de currículums vitae y ofertas laborales.

*El código de programación completo se encuentra en anexos.

A continuación, en la figura Nº 8 se puede ver el resultado de limpieza de datos, en donde claramente se muestra suprimido los artículos, los caracteres que no corresponden a ACSII, la puntuación, los acentos, enteros etc. Este proceso también reduce

el tamaño de la información, pero principalmente su aplicación esencial es la limpieza de datos.

```
#Documento C
user=0
Clean_Words=[]
for i in CV_Habilidades:
    UserList=[]
    data = ' '.join(i)
    words = nltk.word_tokenize(data)
    CleanWOrds(words,user,Clean_Words)
    CV_Habilidades_list=Clean_Words
    user+=1
```

```
(0, '\n', [u'profesional', u'capacitado', u'llevar', u'adelante', u'p
royectos', u'sistema', u'recomendacion', u'aplicando', u'tecnicas',
u'mineria', u'textos', u'scrapy', u'pln', u'nltk', u'invesitgado',
u'publicado', u'revistas', u'sistema', u'recomendacion', u'mineria',
u'textos'])
(1, '\n', [u'cursante', u'iv', u'ciclo', u'administracion', u'negocio
s', u'responsable', u'dinamico', u'metas', u'definidas', u'gran', u'm
otivacion', u'logro', u'poseo', u'habilidad', u'aprender', u'rapide
z', u'involucro', u'prontamente', u'puntualidad', u'honestidad', u're
sponsabilidad', u'logro', u'objetivos', u'organizacion', u'recibido
```

Figura 8: Resultado de limpieza de datos con NLTK, listas de términos sin ruido y caracteres impertinentes.

Frecuencia de términos:

En el presente caso se experimentó el cálculo de frecuencia de términos de cada currículum vitae y ofertas laborales. El experimento consiste en procesar cada término o palabra de currículum vitae. Esto nos permite saber cuáles de los términos es más usado en un CV y oferta laboral, es un indicador inicial que tiene poco significado sobre el contenido de términos en los currículums vitae y ofertas laborales, pero esto nos es suficiente para plantear la relevancia de un CV con respecto a una oferta laboral.

Se pude apreciar en la figura N° 9, donde se muestra el resultado de cálculo de frecuencia de términos que fue desarrollado en lenguaje de programación R.

```
> nov_mat[1:7000,]
                palabra frec
realizar       realizar 2939
clientes       clientes 2201
productos     productos 1983
trabajo         trabajo 1796
cliente         cliente 1289
venta             venta 1197
beneficios   beneficios 1137
control         control 1085
ventas           ventas 1047
contratacion contratacion 1033
fecha             fecha 1004
area               area  947
cumplir         cumplir  896
```

Figura 9: Términos con sus respectivas frecuencias en un CV.

Con el valor de frecuencia de términos se puede realizar algunos análisis de visualización, por ejemplo: gracias a la técnica Wordcloud se grafica una nube de palabras, en donde el tamaño de la palabra es acorde al valor de la frecuencia, como se puede observar en la siguiente figura N° 10, es una manera de interpretar el resultado de forma gráfica y entendible.



Figura 10: Nube de palabras según su valor de frecuencia.

4.2.2 Técnica frecuencia de término – frecuencia inversa de documento TF-IDF.

Term frequency – Inverse document frequency, en español se traduce como frecuencia de termino – frecuencia inversa de documento, es una técnica desarrollado por OnPage.org, uno de los primeros factores de clasificación usado por Google, es una técnica que expresa numéricamente la relevancia de un término en un documento y conjunto de documentos o corpus, es una técnica muy usada para minería de textos, recuperación de información.

El valor TF-IDF es proporcional a la frecuencia con el que se repite el término en un documento, pero es equilibrada por el número de documentos en que se repite el mismo término, esto hace que algunos términos sean más populares que otros en un determinado corpus de datos. Es muy usado en los motores de búsqueda como herramienta para medir la relevancia de términos o búsqueda de palabras clave.

A continuación, vemos la ecuación de TF-IDF:

$$W_{i,j} = tf_{i,j} X log\left(\frac{N}{df_{i,j}}\right)$$

En dónde:

$tf_{i,j} = Numero\ de\ ocurrencias\ de\ i\ en\ j.$

$df_{i,j} = Numero\ de\ documentos\ que\ contiene\ en\ i.$

$N = Total\ numero\ de\ documentos$

La tecnica se aplica en un conjunto de datos de corpus o documentos, para la presente investigación nuestro corpus de datos es el conjunto de currículum vitae y ofertas laborales con sus respectivos términos que describen a cada cual.

Por otra parte, la usabilidad de técnica tf-idf para un sistema de recomendación es sugerida en el libro de Recommender Systems:

> Los métodos basados en contenido se pueden usar para hacer recomendaciones de elementos y etiquetas a los usuarios. Con el fin de recomendar elementos a los usuarios, se puede crear un conjunto de datos de entrenamiento específicos del usuario en el que cada elemento se describe por sus frecuencias de etiqueta sobre los m usuarios. Estas frecuencias se pueden representar en formato tf-idf.
>
> (Aggarwal, 2016, pág. 375)

Se considera para el presente estudio por tratarse de información textual de currículums vitae y ofertas laborales, donde los términos son un conjunto cadenas de etiquetas

combinadas o vectores. Por lo tanto, “clasificamos las ofertas laborales para un candidato determinado que representan la información textual por vectores y con TF-IDF hacemos las ponderaciones de término” (Malherbe, Diaby, Cataldi, Viennet, & Aufaure, 2014, pág. 2).

A continuación, se explica la distribución de la información recuperada en documentos, porque para aplicar la técnica TF-IDF se tiene la necesidad de tener organizado en varios grupos de términos o documentos.

a) Distribución de la información de los candidatos

Considerando lo anterior, para el presente caso se analizó y se determinó considerar tres aspectos principales como documentos de corpus de los currículums vitae, los cuales son: descripción del candidato como primer documento, experiencia profesional acumulado como segundo documento y habilidades del candidato como tercer documento, los tres documentos principales tienen la información sobre los candidatos, en los cuales se buscará el valor frecuencia de termino – frecuencia inversa de documento (TF-IDF) con relación a los términos de las ofertas laborales.

En seguida se detalla el contenido de los documentos:

1) Primer documento:

Descripción de CV: Se considera como primer documento, consiste en información que describe de forma general sobre su visión, misión, objetivos, etc. Una información que da a conocer sobre sí mismo del candidato, como se puede apreciar en la figura N° 11.



Figura 11: Información de la descripción del candidato en su currículum vitae en formato HTML.

2) Segundo documento:

Experiencia profesional CV: Se considera como el segundo documento que consta de información sobre experiencia laboral acumulado del candidato, donde describe cada una de las experiencias adquiridas en diversas empresas en las que ha laborado y ha ejercido su profesión aplicando sus conocimientos, habilidades asumiendo funciones, las mismas se encuentran en la data recolectada como se puede apreciar en la figura N° 12.



Figura 12: Información recuperada sobre la experiencia laboral del candidato en los documentos HTML.

3) Tercer documento:

Habilidades de CV: Se considera como tercer documento, consta de la información sobre habilidades del candidato, en donde menciona específicamente las habilidades con la que cuenta el candidato y esta información es homogénea con el requerimiento de conocimientos por parte de ofertas laborales, en donde también se detalla sobre los conocimientos con el que debe contar el candidato postulante. La información recuperada se puede observar en la figura N° 13.

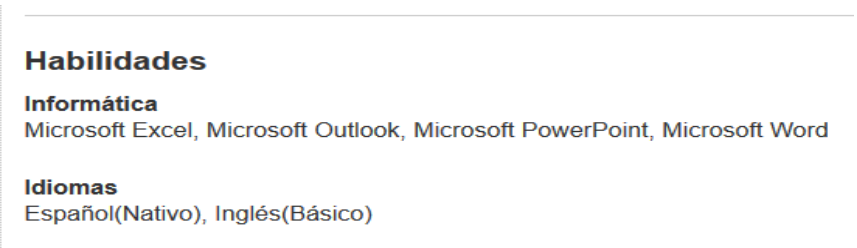


Figura 13: Información recuperada sobre habilidades del candidato en los documentos HTML recolectadas.

b) Distribución de la información de ofertas laborales

La información sobre ofertas laborales es distribuida en cinco documentos, las cuales se consideran como: Titulo, como primer documento, descripción, como segundo documento, requisitos, como tercer documento, funciones, como cuarto documento y conocimientos, como quinto documento de las ofertas laborales, como se puede apreciar en la tabla N° 6.

Tabla 6: Datos de las ofertas laborales recuperada.

Id_Job	Titulo	Empresa	Ciudad	Descripcion	Requisitos	Funciones	Educacion	Conocimientos
0	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
1	Vendedores rotativos para tiendas Baby Infanti...	SGA Consultores SAC	Miraflores Lima	Por encargo de las Tiendas Baby Infanti y Lego...	Secundaria completa. Manejo fluido a nivel usu...		Educación Secundaria	
2	Asistente Contable Técnico o Universitario Egr...	Adecco Perú S.A.	San Borja Lima	Por encargo de nuestro cliente empresa líder ...		Análisis de depreciaciones. Asientos contables...	Técnico	Microsoft Excel
3	Practicante de Administración (Apoyo)	Industrias Plasticas Reunidas S.A.C	Lima Lima	Industrias Plásticas Reunidas es una empresa 1...		Atención directa a gerencia en relación a coor...	Técnico	Microsoft Excel

1) Primer documento:

Título de oferta laboral: En donde contiene información clave que resume el contexto de la oferta laboral, la misma que se homogeniza con la información de la descripción del candidato en los currículums vitae.

2) **Segundo documento:**

Descripción de oferta laboral: Donde hay un contenido de términos que describe el contexto de la oferta laboral, dicha información es homogénea con la información de **descripción** de currículums vitae, donde también describe de manera general el contexto del candidato.

3) **Tercer documento:**

Requisitos de oferta laboral: En donde hay contenido de términos que describen a las especificaciones y requisitos mínimos que los candidatos deben cumplir, la cual es homogénea con la información se encuentra en la parte **experiencia y habilidades** de los candidatos en su currículum vitae.

4) **Cuarto documento:**

Funciones de oferta laboral: En donde hay términos que describen las funciones y/o tareas que asumiría el nuevo empleado. Para este punto se considera importante la experiencia de candidato, adquiridas en diversas empresas en las que ha aplicado sus conocimientos y habilidades en determinadas funciones, dicha información es homogénea con la información que se encuentra en la parte **experiencia** de los CV.

5) **Quinto documento:**

Conocimientos de oferta laboral: En donde se menciona específicamente las habilidades y conocimientos que debe tener el candidato, esta información es homogénea con la información que se encuentra en la parte de **habilidades** en los CV, en donde también detalla sobre los conocimientos con que cuenta el candidato.

Entonces, considerando la homogeneidad de información de documentos de currículum vitae de los candidatos y ofertas laborales se procede a calcular el valor tf-idf de cada uno de los términos.

4.2.3 Implementación de frecuencia de término-frecuencia inversa de documento (TF-IDF)

Dado la distribución de la información recuperada en documentos en el anterior punto, a continuación, se explica el desarrollo de la técnica frecuencia de términos – frecuencia inversa de documentos, para hallar el valor de relevancia de cada uno de los términos de currículum vitae con relación a todo el corpus de datos de cada oferta laboral.

El algoritmo consta como datos de entrada a una serie de vectores con listas de términos de los currículums vitae y ofertas laborales, los cuales son distinguidas por una clave primaria; es decir, los cinco documentos de las ofertas laborales distribuidos en

el punto anterior, se unen en una sola lista general. Por otra parte, las listas te términos de currículums vitae son desplegadas en tres listas diferentes (tres documentos distribuidos en el punto anterior).

En la implementación del algoritmo, los documentos son reconocidos en orden alfabético, los mismos que fueron distribuidas en orden numérica en al anterior punto, como se puede apreciar de manera detallada en la tabla N° 7 y 8.

Tabla 7: Información distribuida en documentos de ofertas laborales en orden alfabético para la programación del algoritmo de TFIDF

Ítem	Campos	Se considera como:
1	Titulo	Documento A
2	Empresa	No se considera
3	Ciudad	No se considera
4	Descripción	Documento B
5	Requisitos	Documento C
6	Funciones	Documentos D
7	Educación	No se considera
8	Conocimientos	Documento E

Tabla 8: Información distribuida en documentos de currículums vitae en orden alfabético para la programación del algoritmo de TFIDF

Ítem	Campos	Se considera como:
1	Nombre	No se considera
2	Descripción	Documento A
3	Experiencia	Documento B
4	Habilidades	Documento C

Tf-idf adaptado al presente caso inicia con la importación de datos pre procesados en la etapa de limpieza y trasformación de datos. Como datos de entrada se consideran dos principales vectores: La primera corresponden a las ofertas laborales (V1) y la segunda a currículums vitae (V2), el primer vector tiene como sub vectores a la cantidad de ofertas laborales procesadas, en donde los elementos son los términos de cada oferta laboral, por otra parte, en el segundo vector tiene como sub vectores la cantidad total de currículums vitae, y cuyo elementos del sub vector se consideran la cantidad total de documentos por cada CV, como se aprecia en la figura N°14.

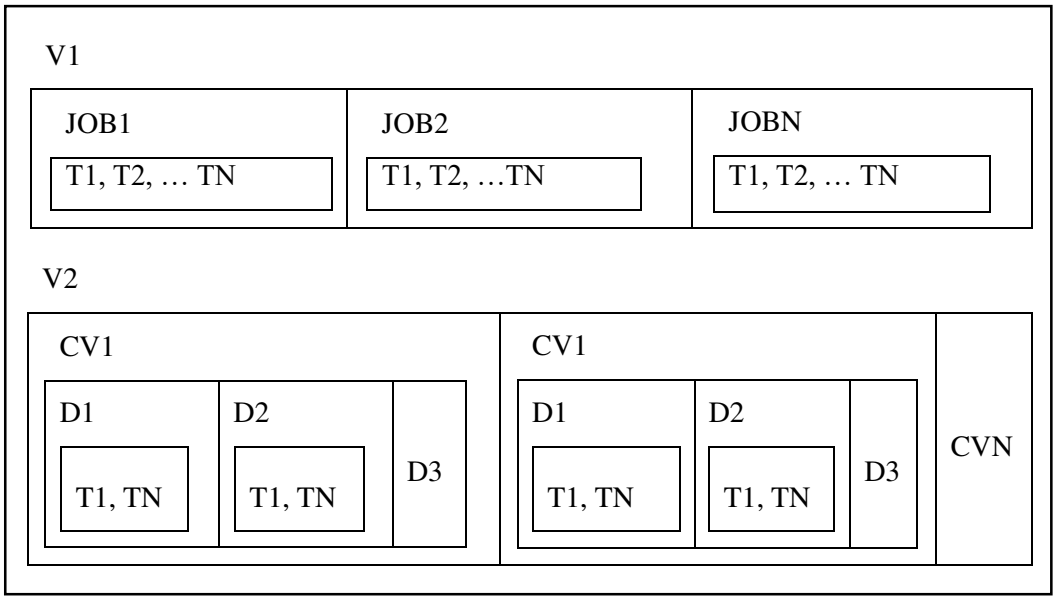


Figura 14: Datos de entrada para if-idf, vectores que almacenan términos de currículums vitae y ofertas laborales.

En donde:

V1: es el vector principal de ofertas laborales, que cuyos elementos son un conjunto de ofertas laborales (JOB1, JOB2… JOBN), y los sub vectores tienen como elementos a los términos de cada uno de los documentos de las ofertas laborales juntados en una sola lista (T1, T2… TN).

V2: es el vector principal de los currículums vitae, que cuyos elementos son un conjunto de currículums vitae (CV1, CV2… CVN), y los sub vectores de currículums vitae tienen como elementos a los documentos de cada CV (D1, D2… D3) y finalmente los sub vectores de documentos como elementos tienen a los términos de currículums vitae (T1, TN).

En la ecuación del algoritmo consta de dos partes importantes, la primera es para hallar el valor de término de frecuencia en un documento determinado ($W_{i,j} = tf_{i,j}$) y la segunda es para hallar el valor de frecuencia inversa de documento ($log\left(\frac{N}{df_{i,j}}\right)$), finalmente el producto de estos dos valores es el valor TF-IDF o relevancia de término.

En la figura N° 15 se puede apreciar la primera parte del diagrama de flujo del algoritmo tf-idf configurado para la presente investigación, donde básicamente como datos de entrada se tiene "Cv" que representa a currículums vitae y "Job" a ofertas laborales. En seguida se hacen lectura de los dos vectores principales (Cv y Job).

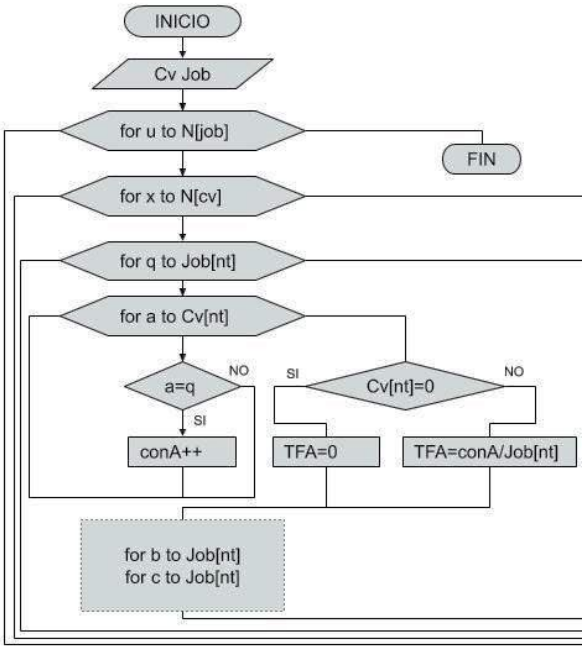


Figura 15: Primera parte de diagrama de flujo de tf-idf configurado para la presente investigación.

En donde:

for x to N[job]: Importa los datos de vector principal que corresponde a las ofertas laborales, es decir la lista total de ofertas laborales.

for u to N[cv]: Importa los datos de vector principal que corresponde a currículums vitae, es decir la lista total de CV.

for q to Job[nt]: Hace lectura de datos del sub vector cuyo elementos son términos de las ofertas laborales, es decir la lista total de términos o palabras de las ofertas laborales.

for a to Cv[nt]: Importa los datos del sub vector de documentos de los currículums vitae, es decir la lista de términos del primer documento (documento A). Y este último procedimiento se repite tres veces, ya se tiene distribuido en tres documentos los términos de los currículums vitae .

a: Variable que hace lectura a cada uno de los elementos del vector de términos de los currículums vitae, es decir por la variable "a" pasan todos los términos que contenga los documentos A. y este procedimiento se repite para documento B y C, con la finalidad de hallar el valor tf-idf en cada uno de estos documentos.

q: La variable que hace lectura a los elementos de vector de términos de ofertas laborales, es decir por la variable "q" pasan todos los términos que contenga una oferta laboral.

conA++: Es la variable del contador de términos repetidos en el primer documento (documento A), para los demás documentos solo cambia el carácter en mayúscula por el documento que corresponda (conB, conC, conD y conE).

TFA: Es la variable en donde se guarda el valor de termino de frecuencia del documento A, para los demás documentos solo cambia el ultimo carácter por el documento que corresponda (TFB, TFC, TFD y TFE).

En esta primera parte del diagrama de flujo se importa los datos necesarios y calcula la frecuencia del término (tf) de currículums vitae, de cada uno de sus documentos (documento A, documento B y documento C).

En la segunda parte del diagrama de flujo que se puede apreciar en la figura N° 16, se calcula el número de documentos en que se repite los términos de ofertas laborales, los cual es necesario para calcular al frecuencia inversa de documento (idf), para hallar el valor IDF se aplica el logaritmo de la suma total de documentos del corpus entre la cantidad de documentos en el que se repite el término ($log\left(\frac{N}{df_{i,j}}\right)$).

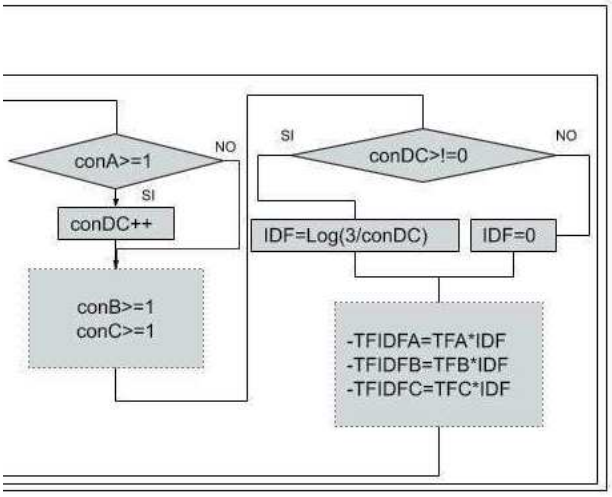


Figura 16: Segunda parte del diagrama de flujo de tf-idf configurado para la presente investigación.

Dónde:

conDC: Es el variable contador de los documentos que acumula el número de documentos en el que se repite los términos.

IDF: Es la variable que acumula la frecuencia inversa de documentos, el cual es el valor calculado de logaritmo del total de documentos del corpus entre el número de documentos en el que se repite el término.

IFIDFA: Finalmente se tiene la variable que acumula el valor de frecuencia de términos – frecuencia inversa de documentos. Este variable se repite para todos los documentos de currículums vitae, pero cambia el último carácter de la variable para los demás documentos (IFIDFA, IFIDFB e IFIDFC).

A continuación, el diagrama de flujo completo de TF-IDF adaptado para el presente estudio.

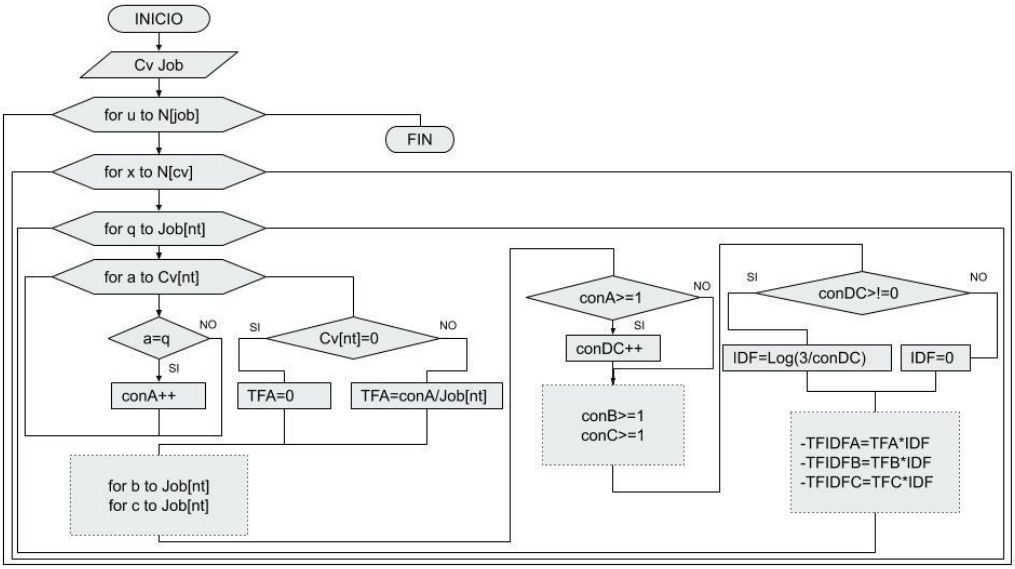


Figura 17: Diagrama flujo completa de tf-idf.

*El código de programación de implementación de técnica tf-idf está en los anexos.

Las listas de términos de los currículums vitae son las que nos permite encontrar el valor numérico real de relevancia con respecto a todas las ofertas laborales. En la figura Nº 18 se puede apreciar que se tiene una lista de términos de entrada, en donde:

- El primer grupo de términos corresponde a los términos unidas en una sola lista de los cinco documentos de las ofertas laborales.
- El segundo grupo de listas corresponde a listas de términos de los tres documentos de currículums vitae por separado.

- El tercer grupo de tres listas de términos corresponden a las listas de términos relevantes que son identificados en sus respectivos documentos de un currículum vitae, los términos son seguidos por su valor tfidf, el valor que indica el peso o relevancia. Estos valores permiten tener el valor ponderado de tfidf para cada documento del CV (se muestra en el siguiente grupo), es decir para: Descripción de CV (Documento A), Experiencia del CV (documento B) y Habilidades del CV (documento C).
- El cuarto grupo se muestran los resultados finales, donde se obtiene un valor ponderado general de if-idf, el cual es el promedio de promedios de tf-idf de los documentos del CV, el cual se considera como un indicador de relevancia de un CV con respecto a una oferta laboral, puesto que este valor corresponde a la importancia y/o peso de los términos de CV.

En el curso de Introduction to Recommender Systems: Non-Personalized and Content-Based de Coursera en la parte TFIDF and Content Filtering explica el uso de TFIDF para filtrado de términos en documentos apara sistema de recomendación basada en contenido. Por otra parte, también es usado y recomendado en la investigación de (Okaka, Mwangi, & Okeyo, 2016), en donde concluye que el TFIDF es eficiente para medir las similitudes.

Por tanto, tf-idf en el presente estudio se aplica precisamente para medir la relevancia de términos de detalles de las ofertas laborales en currículums vitae, y el valor ponderado es un indicador numérico de relevancia de un CV con respecto a las ofertas laborales y este dato se plantea como un valor rating para sistemas de recomendación, puesto que los sistemas de recomendación se basan en valoraciones o rating de los artículos por parte de los usuarios, y en este caso la valoración de un CV con respecto a las ofertas laborales basada en contenido de sus términos.

```
OferList=[]
TFIDF_OferDescription_list=[]
campo=CorpusjobUnduplicated
TFIDFCV(campo,CV_Description_list,CV_Experiencia_list,CV_Habilidades_list,OferList)
TFIDF_CVwords_list=OferList
```

GRUPO 1

```
Oferta laboral:
set([u'publicaciones', u'conocimientos', u'necesita', u'python', u'mineria', u'laborales', u'proyecto', u'lenguaj
e', u'procesameinto', u'scrapy', u'artificial', u'tecnologica', u'inteligencia', u'ofertas', u'peru', u'investigad
or', u'minerria', u'preparado', u'procesamiento', u'aplicando', u'innovacion', u'sistema', u'liderar', u'encargad
o', u'recomendacion', u'natural', u'proyectos', u'textos', u'etc', u'tecnicas', u'profesional', u'nltk', u'linea',
u'desarrollador'])
```

GRUPO 2

```
Docuemntos de Currículum Vitae:
[u'profesional', u'capacitado', u'llevar', u'adelante', u'proyectos', u'sistema', u'recomendacion', u'aplicando',
u'tecnicas', u'mineria', u'textos', u'scrapy', u'pln', u'nltk', u'invesitgado', u'publicado', u'revistas', u'siste
ma', u'recomendacion', u'mineria', u'textos']
[u'publicaciones', u'sistema', u'recomendacion', u'mineria', u'textos']
[u'python', u'pln', u'nltk', u'scrapy', u'tec']
```

GRUPO 3

```
Resultados:
DOC A:  [u'mineria', '0.039', u'scrapy', '0.019', u'aplicando', '0.052', u'sistema', '0.039', u'recomendacion',
'0.039', u'proyectos', '0.052', u'textos', '0.039', u'tecnicas', '0.052', u'profesional', '0.052', u'nltk', '0.01
9']
DOC B:  [u'publicaciones', '0.22', u'mineria', '0.081', u'sistema', '0.081', u'recomendacion', '0.081', u'textos',
'0.081']
DOC C:  [u'python', '0.22', u'scrapy', '0.081', u'nltk', '0.081']
```

GRUPO 4

```
PROMEDIOS DE TF-IDF
Id_Job | 0 | Id_Cv | 0 | promedio general | 0.092 | promedios por documentos | 0.04 | 0.109 | 0.127
```

Figura 18: Resultado de procesamiento de términos de los currículums vitae y ofertas laborales con TF-IDF.

4.2.4 Interpretación del resultado.

A continuación, en la figura N° 19 se aprecia la información de la primera oferta laboral sin procesar, recuperada y distribuida en documentos (A, B, C, D y E), un conjunto de términos que corresponden a la primera oferta laboral y sus especificaciones técnicas divididas en documentos. En la figura N° 20 se aprecia la información de un currículum vitae en su estado natural, el cual corresponde al primer CV.

En seguida, en la figura N° 21 se aprecia el resultado del procesamiento con nltk, en donde se puede apreciar una lista de terminos libre de caracteres incoherentes, acentos, enteros, puntuaciones y todas uniformizadas en minúsculas, de lo que era cinco documentos se obtiene una sola lista compuesta por términos de los cinco documentos de la oferta laboral.

En la figura N° 22 se aprecia el currículum vitae procesado con nltk y separadas en tres documentos (A, B y C), se mantienen separados porque se hallará en eses documentos la relevancia de términos de la oferta laboral.

A. Investigador
B. Se necesita un profesional muy preparado para liderar proyectos de innovacion tecnologica en el peru, investigador en linea de inteligencia artificial, con publicaciones sobre sistema de recomendacion de ofertas laborales, mineria de textos, procesameinto lenguaje natural, etc.
C. Conocimientos en procesamiento lenguaje natural, python, scrapy, minerria de textos, sistema de recomendacion etc
D. Desarrollador encargado del proyecto de sistema de recomendacion de ofertas laborales aplicando de tecnicas de mineria de textos y procesamiento lenguaje natural
E. python, scrapy, procesamiento lenguaje natural, nltk

Figura 19: Muestra de información de ofertas laborales en bruto.

A. Soy un profesional capacitado para llevar adelante proyectos sobre sistema de recomendacion aplicando tecnicas de mineria de textos, scrapy, pln, nltk, he invesitgado y publicado revistas sobre sistema de recomendacion y mineria de textos
B. publicaciones sobre sistema de recomendacion y mineria de textos
C. python, pln, nltk, scrapy, tec

Figura 20: Muestra de información en bruto de los currículums vitae.

```
set([u'publicaciones', u'conocimientos', u'necesita', u
'python', u'mineria', u'laborales', u'proyecto', u'leng
uaje', u'procesameinto', u'scrapy', u'artificial', u'te
cnologica', u'inteligencia', u'ofertas', u'peru', u'inv
estigador', u'minerria', u'preparado', u'procesamiento'
, u'aplicando', u'innovacion', u'sistema', u'liderar',
u'encargado', u'recomendacion', u'natural', u'proyectos
', u'textos', u'etc', u'tecnicas', u'profesional', u'nl
tk', u'linea', u'desarrollador'])
```

Figura 21: Muestra del resultado de procesamiento con NLTK las ofertas laborales.

```
A. [u'profesional', u'capacitado', u'llevar', u'ade
   lante', u'proyectos', u'sistema', u'recomendacio
   n', u'aplicando', u'tecnicas', u'mineria', u'tex
   tos', u'scrapy', u'pln', u'nltk', u'invesitgado'
   , u'publicado', u'revistas', u'sistema', u'recom
   endacion', u'mineria', u'textos']
B. [u'publicaciones', u'sistema', u'recomendacion',
   u'mineria', u'textos']
C. [u'python', u'pln', u'nltk', u'scrapy', u'tec']
```

Figura 22: Muestra del resultado de procesamiento con NLTK los currículums vitae.

En el siguiente ejemplo se muestran los 5 primeros términos y sus respectivos valores tf-idf (minería, scrapy, aplicando, sistema y recomendación) del documento A, como se puede apreciar en la figura N° 23. Los cuales se detallan en la tabla N° 9, en donde se aprecia el proceso de procesamiento de los términos para obtener los valores: frecuencia de termino (tf), frecuencia inversa de documentos (idf) y finalmente el valor frecuencia de términos – frecuencia inversa de documentos (tfidf) para cada uno de los documentos de currículums vitae.

```
DOC A:  [u'mineria', '0.039', u'scrapy', '0.019', u'apl
icando', '0.052', u'sistema', '0.039', u'recomendacion'
, '0.039', u'proyectos', '0.052', u'textos', '0.039', u
'tecnicas', '0.052', u'profesional', '0.052', u'nltk',
'0.019']
DOC B:  [u'publicaciones', '0.22', u'mineria', '0.081',
u'sistema', '0.081', u'recomendacion', '0.081', u'texto
s', '0.081']
DOC C:  [u'python', '0.22', u'scrapy', '0.081', u'nltk'
, '0.081']

PROMEDIOS DE TF-IDF
Id_Job | 0 | Id_Cv | 0 | promedio general | 0.092 | pro
medios por documentos | 0.04 | 0.109 | 0.127
```

Figura 23: Muestra de términos relevantes del primer CV en la primera oferta laboral.

Para medir la relevancia de los currículums vitae se procesa cada uno de los términos de las ofertas laborales, en la tabla N° 9 se pude aprecia los 5 primeros términos de la primera oferta laboral, los cuales se miden su relevancia en los términos del primer currículum vitae, cabe mencionar en anteriores puntos que los términos de un CV son distribuidos en tres documentos (descripción, experiencia y habilidades).

Entonces, los campos: FTA, FTB y FTC, corresponden a la frecuencia de términos en sus respectivos documentos del currículum vitae.

En seguida se aprecia el campo IDF, lo cual corresponde al valor frecuencia inversa de documentos, el cual se obtiene al calcular el logaritmo de número total de documentos entre el número de documentos en el que se repite el término.

Posteriormente se obtiene el valor de frecuencia de términos – frecuencia inversa de documentos, los cuales se ubican en los campos TFIDFA, TFIDFB y TFIDFC, el cual se halla al aplicar el producto de los dos valores anteriores (TF, IDF) para respectivos documentos.

En parte inferior de la tabla N° 9 se aprecia el valor **Promedio de if-idf por documentos**, el cual corresponde al promedio de todos los valores if-idf de los términos relevantes en respectivo documento.

Finalmente, en seguida se aprecia el valor **Promedio general**, el cual es el promedio general de los promedios de tf-idf de sus respectivos documentos. Este valor es el que representa al valor de relevancia de un currículum vitae con respecto a una oferta laboral en base a los términos escrito en forma textual en los campos de un CV, como: descripción del candidato, experiencia del candidato y habilidades del candidato.

Tabla 9: Ejemplo de estructuración relevancia de términos del primer CV respecto a la primera oferta laboral.

Termino	FTA	FTB	FTC	IDF	TFIDFA	TFIDFB	TFIDFC
Minería	2/21	1/5	0/5	log(3/2)	0.039	0.081	0
Scrapy	1/21	0/5	1/5	log(3/2)	0.019	0	0.081
Aplicando	1/21	0/5	0/5	log(3/1)	0.052	0	0
Sistema	2/21	1/5	0/5	log(3/2)	0.039	0.081	0
Recomendación	2/21	1/5	0/5	log(3/2)	0.039	0.081	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
Promedio de if-idf por documentos					0.04	0.109	0.127
						Promedio general	0.092

4.3 Extracción de información

El principal problema de recomendación de ofertas laborales es tener información de valoración de los currículums vitae que está escrito en lenguaje humano, en el proceso anterior se obtuvo promedios de tf-idf para cada documento de los currículums vitae. Como datos de salida de la técnica frecuencia de términos – frecuencia inversa de documentos son los promedios de tf-idf para respectivos documentos y un promedio general sobre los cuales, lo cual representa al valor de la información relevante e importante que tiene un CV y puede servir como un valor rating para la recomendación de ofertas laborales.

Uno de los elementos más importante de sistemas de recomendación son los valores rating, que es el valor de satisfacción o gusto de los usuario sobre los productos o elementos, mientras para la recomendación de ofertas laborales, el elemento a valorar son las ofertas laborales a partir del contenido de términos en los currículums vitae, los promedios de tf-idf son indicadores que representan el valor de la relevancia de los CV con respecto a las ofertas laborales, entonces, el promedio general de tf-idf para la presente investigación se considera como valores rating para recomendación de ofertas laborales, ya que es valor calculado según al contenido de términos de CV.

Por lo tanto, para el objetivo del estudio se extrae los promedios de tf-idf detallados por documentos, como se puede ver en la tabla N° 10, en donde los promedios son para cada documentos. En la tabla N° 11 se

puede ver un solo valor de promedio de tf-idf, el cual es el hallado sobre los promedios individuales de los documentos.

En síntesis, se extrae los datos promedios de tfidf de manera detallada por documentos (tabla N° 10) y otro en general (tabla N° 11), los cuales para el presente trabajo se consideran como rating de los CV con respecto las ofertas laborales, de tal manera que esta pueda ser usada como valor rating en la recomendación de ofertas laborales.

Tabla 10: Promedios de valores relevantes de TF-IDF para respectivos documentos.

Id_Cv	Id_Job	Rating_Descripcion	Rating_Experiencia	Rating_Habilidades
0	0	0.04	0.109	0.127
0	1	0	0	0
0	2	0.005	0.006	0
0	3	0.002	0.003	0
0	4	0	0	0.037
0	5	0.006	0	0
0	6	0.007	0.012	0
0	7	0	0	0
0	8	0.027	0	0
0	9	0.005	0	0
0	10	0	0	0

Tabla 11: Promedio general de TF-IDF sobre los valores TFIDF de respectivos documentos como se muestra en la tabla anterior.

Id_Job	Id_Cv	Rating_general
0	0	0.092
0	1	0
0	2	0.004
0	3	0.002
0	4	0.012
0	5	0.002
0	6	0.006
0	7	0
0	8	0.009
0	9	0.002
0	10	0

4.4 Minería de datos

A partir de información procesada y estructurada en la etapa anterior se puede comenzar a analizar y visualizar algunos resultados con los datos extraídos y la utilidad que puede tener para diferentes fines.

En la tabla Nº 13 Se puede apreciar una matriz de 5000 currículums vitae por 9998 puestos de trabajo (5000x9998), los cuales son representados por su clave principal y en donde el valor de relevancia de cada currículum vitae es distribuido de forma matricial. Haciendo muy comprensible para los seres humanos y estructurada para ser usada en todo tipo de estudios con otros algoritmos de inteligencia artificial y/o afines.

A continuación, se aprecia en la tabla Nº 12 una matriz agrupada de 10 por 10, una muestra para demostrar la usabilidad de estos datos de forma gráfica, estos datos son flexibles para todo tipo de estudio y análisis de información.

Tabla 12: Resumen de la matriz de la Tabla 11.

```
invoices = pd.DataFrame(R_df,columns = [0,1,2,3,4,5,6,7,8,9,10])
invoices.head(10)
```

	0	1	2	3	4	5	6	7	8	9	10
Id_Job											
0	0.092	0.000	0.004	0.002	0.012	0.002	0.006	0.000	0.009	0.002	0.000
1	0.000	0.008	0.013	0.013	0.037	0.023	0.021	0.029	0.040	0.009	0.019
2	0.000	0.023	0.013	0.030	0.024	0.034	0.033	0.047	0.015	0.018	0.016
3	0.000	0.023	0.013	0.030	0.048	0.041	0.030	0.051	0.032	0.013	0.022
4	0.000	0.025	0.013	0.030	0.033	0.035	0.035	0.047	0.032	0.013	0.016
5	0.000	0.022	0.013	0.030	0.030	0.043	0.035	0.034	0.024	0.017	0.016

Tabla 13: Matriz de rating de los CV respecto a las ofertas laborales.

```
R_df = ratings_df.pivot(index = 'Id_Job', columns ='Id_Cv', values = 'Rating_general').fillna(0)
R_df.head(10)
```

Id_Cv	0	1	2	3	4	5	6	7	8	9	...	4990	4991	4992	4993	4994	4995	4996	4997	4998	4999
Id_Job																					
0	0.092	0.000	0.004	0.002	0.012	0.002	0.006	0.000	0.009	0.002	...	0.003	0.001	0.006	0.031	0.000	0.002	0.000	0.015	0.001	0.000
1	0.000	0.008	0.013	0.013	0.037	0.023	0.021	0.029	0.040	0.009	...	0.041	0.038	0.030	0.058	0.030	0.001	0.049	0.011	0.012	0.004
2	0.000	0.023	0.013	0.030	0.024	0.034	0.033	0.047	0.015	0.018	...	0.093	0.002	0.022	0.029	0.031	0.014	0.107	0.012	0.001	0.002
3	0.000	0.023	0.013	0.030	0.048	0.041	0.030	0.051	0.032	0.013	...	0.094	0.002	0.022	0.063	0.039	0.014	0.058	0.017	0.001	0.002
4	0.000	0.025	0.013	0.030	0.033	0.035	0.035	0.047	0.032	0.013	...	0.094	0.001	0.025	0.060	0.051	0.013	0.051	0.015	0.001	0.002
5	0.000	0.022	0.013	0.030	0.030	0.043	0.035	0.034	0.024	0.017	...	0.094	0.038	0.026	0.026	0.050	0.014	0.084	0.014	0.001	0.003
6	0.000	0.005	0.001	0.001	0.033	0.009	0.008	0.006	0.018	0.013	...	0.003	0.001	0.017	0.026	0.002	0.001	0.031	0.002	0.001	0.003
7	0.017	0.017	0.025	0.001	0.012	0.003	0.006	0.004	0.024	0.006	...	0.002	0.001	0.011	0.050	0.002	0.001	0.055	0.003	0.001	0.005
8	0.000	0.030	0.013	0.032	0.025	0.032	0.025	0.047	0.023	0.022	...	0.093	0.002	0.029	0.069	0.040	0.014	0.086	0.018	0.004	0.003
9	0.000	0.006	0.001	0.001	0.028	0.010	0.008	0.006	0.014	0.012	...	0.003	0.001	0.019	0.029	0.002	0.001	0.023	0.002	0.001	0.007

El modelo permite visualizar los datos con mucho más detalle, como en las figuras Nº 24 y 25, en la primera se puede visualizar la diferencia de los valores de relevancia de los currículums vitae con respecto a 10 primeras ofertas laborales.

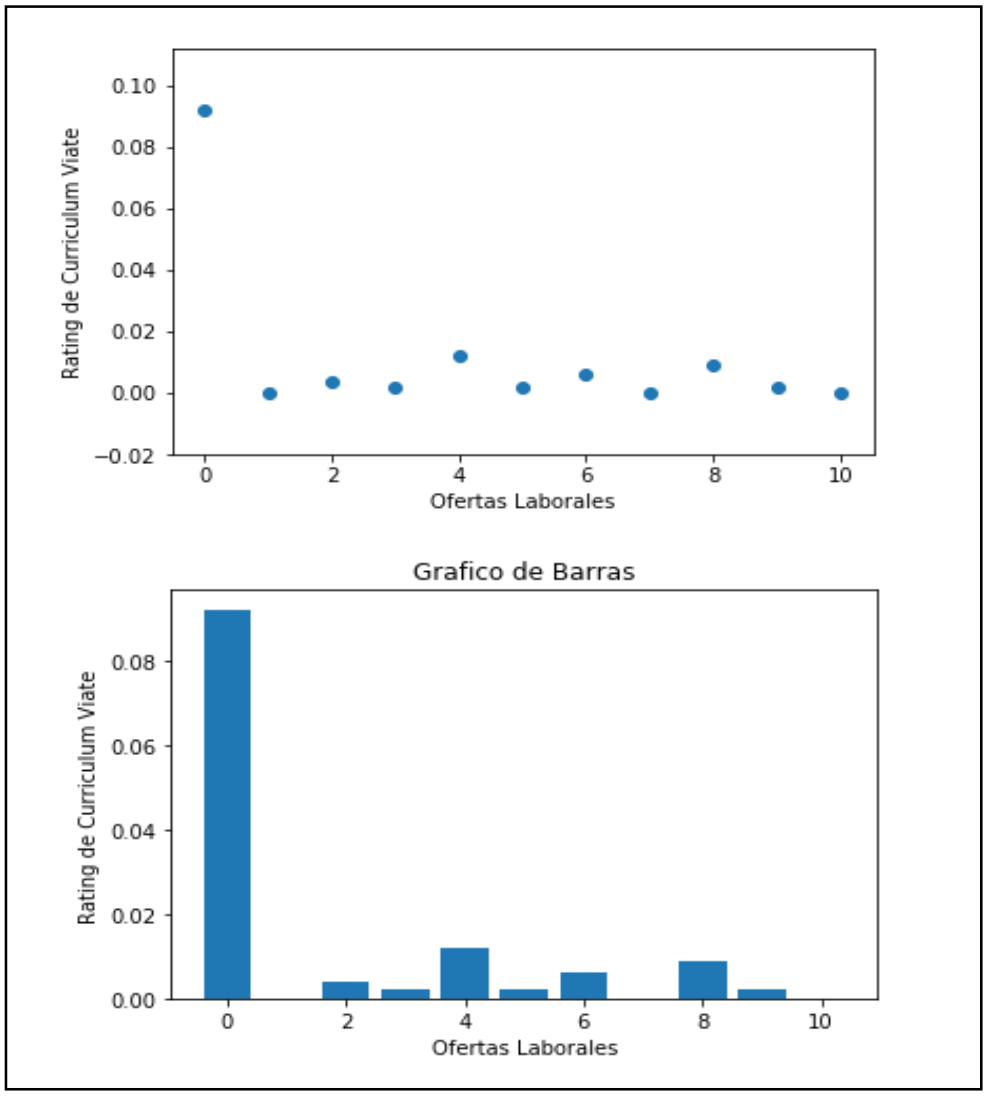


Figura 24: Ilustración de la relevancia del primer CV respecto a varias ofertas laborales.

Los promedios de tf-idf permiten diferenciar que algunos CV son más relevantes que otros en diferentes ofertas laborales, como se puede apreciar en la figura N° 25, esto es posible a tras el procesamiento de

términos escrito en lenguaje humano en los currículums vitae y ofertas laborales.

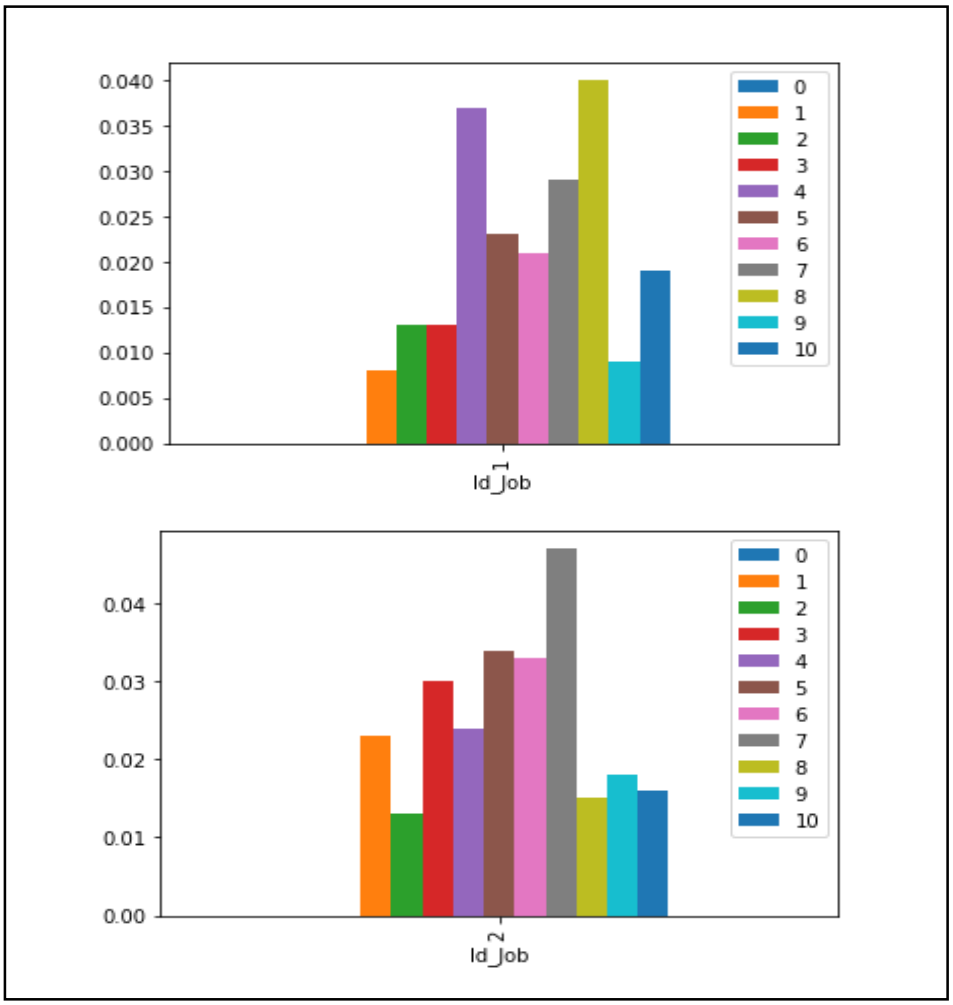


Figura 25: Ilustración grafica de relevancia de dos CV (1 y 2) con respecto a varias ofertas laborales.

Para el presente trabajo se busca plantear un modelo minería de datos para recomendación de ofertas laborales, que básicamente consiste en una data estructurada con valores rating para que puedan ser de utilidad para los algoritmos de sistemas de recomendación.

La propuesta del presente trabajo se concluye con una forma de procesar la información no estructurada de currículums vitae y ofertas

laborales. El proceso detallado desde la recolección de data no estructurada desde los sitios web, estructuración, reducción y extracción de data relevante con técnicas de NLP y TM hasta plantear el modelo minería de datos para algoritmos de sistema de recomendación.

Resultado final se puede observar en la tabla N° 14, en donde:

- Id_Job: Es la llave principal de las ofertas laborales.
- Id_Cv: Es la llave principal de los CV del candidato.
- Rating general: Es el valor de relevancia de los currículums vitae respecto a las ofertas laborales.
- Título: Campo de título de las ofertas laborales.
- Empresa: Campo de nombres de las empresas que ofertan los puestos laborales.
- Ciudad: Ciudad en el que se ubica el puesto laboral.
- Descripción: Descripción general sobre la oferta laboral.
- Requisitos: Especificaciones de requisitos de la oferta laboral.
- Funciones: Descripción de funciones de la oferta laboral.

Tabla 14: Modelo de datos detallado con rating de currículums vitae respecto a las ofertas laborales

Id_Job	Id_Cv	Rating_general	Titulo	Empresa	Ciudad	Descripcion	Requisitos	Funciones	Educacion	Conocimientos
0	0	0.092	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
0	1	0.000	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
0	2	0.004	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
0	3	0.002	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
0	4	0.012	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
0	5	0.002	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...

V. CONCLUSIONES

5.1 Conclusiones generales

El uso de las técnicas de minería de textos y procesamiento lenguaje natural para construir un modelo de minería de datos para la recomendación de ofertas laborales a partir de la data no estructurada de currículums vitae y ofertas laborales de sitios web, son de gran utilidad para la recuperación de información, procesamiento de los textos escritos en lenguaje humano, para la extracción de información relevante y análisis del mismo.

5.2 Conclusiones específicas

Frecuencia de término – frecuencia inversa de documentos (tf-idf) resultó ser eficaz para identificar información más relevante de un corpus de datos de currículums vitae y ofertas laborales en base a los términos escrito de forma textual y en lenguaje humano, finalmente los promedios generales de tf-idf se plantean como el valor rating de los currículums vitae para la recomendación de ofertas laborales.

VI. RECOMENDACIONES

Se recomienda continuar la investigación en línea de sistemas de recomendación para dar utilidad la dataset de la presente investigación, puesto que este estudio solo se centra en la primera etapa de sistema de recomendación, en donde el objetivo es obtener data pre procesada y estructurada para posteriormente aplicar diversos algoritmos de sistemas de recomendación.

El modelo de datos propuesto puede ser usado en una investigación relacionada a la empleabilidad de los egresados universitarios en el mercado laboral, buscando la relevancia del contenido de cursos que recibe el estudiante en las aulas universitarias.

VII. Referencias

Martınez Lopez, L., Castro Gallardo, J., & Barranco Garcıa, M. (2012). Un nuevo modelo ponderado para Sistemas de Recomendaci´on Basados en Contenido con medidas de contingencia y entropıa. *Universidad de Jaén*.

Okaka, R., Mwangi, W., & Okeyo, G. (2016). A Hybrid Approach for Personalized Recommender System Using Weighted TFIDF on RSS Contents. *International Journal of Computer Applications Technology and Research*, 11.

Takane, Y., & Young, F. (1977). NONMETRIC INDIVIDUAL DIFFERENCES MULTIDIMENSIONAL SCALING: AN ALTERNATING LEAST SQUARES METHOD WITH OPTIMAL SCALING FEATURES. *UNIVERSITY OF NORTH CAROLINA*, 61.

Adrade, J. (s.f.). El proceso de diseño del plan de estudios. 20.

Aggarwal, C. C. (2016). *Recommender Systems.* Springer International Publishing Switzerland.

Allahyari, M., Pouriyeh, S., Assef, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *[cs.CL] 28 Jul 2017*.

Almada, R. V., Elias, O. M., Gómez, C. E., Mendoza, M. D., & López, S. G. (2017). Natural Language Processing and Text Mining to Identify Knowledge Profiles for Software Engineering Positions. *2017 5th*

International Conference in Software Engineering Research and Innovation (CONISOFT).

Barbieri, N., Manco, G., & Ritacco, E. (2014). Probabilistic Approaches to Recommendations. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 38.

Brun, R. E., & Senso, J. A. (2004). Minería textual. *El profesional de la información*.

bugra. (25 de 09 de 2017). *bugra*. Obtenido de bugra: https://bugra.github.io/work/notes/2014-04-19/alternating-least-squares-method-for-collaborative-filtering/

Castellanos, Y. R. (2014). Sistema de recomendación por filtrado colaborativo para el sistema de publicación de contenido multimedia - VideoWeb 1.0. *Universidad de las Ciencias Informáticas UCI, La Habana, Cuba.*, 9.

Cedano, J. Á. (2015). *Modelo de minería de datos para indentificación de patrones que influyen en el aprovechamiento academico.* Instituto tecnológico de la paz.

Comon, P., Luciani, X., & De Almeida, A. (2009). Tensor Decompositions, Alternating Least Squares and other Tales. *Journal of Chemometrics, Wiley, 2009*, 32.

Cortez, A. V., Vega, H. H., & Pariona, J. Q. (2009). Procesamiento Lenguaje Natural. *Revista de Ingeniería de Sistemas e Informática vol. 6, N.º 2, Julio - Diciembre 2009*.

Ekstrand, M., & Riedl, J. (2010). Collaborative Filtering Recommender Systems. *Foundations and Trends inHuman–Computer Interaction*, 95.

Emmanuel, M., & Aufaure, M.-A. (2016). Bridge the Terminology Gap Between Recruiters and Candidates: A Multilingual Skills Base built from Social Media and Linked Data. *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Fernández, A. R., & Gavilanes, J. Á. (s.f.). DIAGNÓSTICO DE LA INSERCIÓN LABORAL DE GRADUADOS DE ADMINISTRACIÓN DE EMPRESAS DE LA UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES. 19.

Gao, L., & Eldin, N. (2014). Employers expectations: A probabilistic text mining model. *Creative Construction Conference 2014, CC2014*.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *University of Cambridge*, 24.

Hearst, M. A. (s.f.). Untangling Text Data Mining. *School of Information Management - Systems University of California, Berkeley*.

HERLOCKER, J., TERVEEN, L., KONSTAN, J., & RIEDL, J. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 49.

Internet World Stats. (30 de Junio de 2017). *Internet World Stats*. Obtenido de http://www.internetworldstats.com/stats.htm

Internet World Stats. (11 de 2018). *Internet World Stats*. Obtenido de https://www.internetworldstats.com/

Kao, A., & Poteet, S. R. (2007). *Natural Language Processing and Text Mining*. British Library.

Karakatsanis, I., AlKhader, W., MacCrory, F., Alibasic, A., Omar, M. A., Aung, Z., & Woon, W. L. (2016). Data Mining Approach to Monitoring The Requirements of the Job Market: A Case Study. *Electrical Engineering and Computer Science, Masdar Institute of Science and Technology, Abu Dhabi, United Arab Emirates*.

Kinoa, Y., Kurokia, H., Machidab, T., Furuyab, N., & Takanob, K. (2017). Text Analysis for Job Matching Quality Improvement. *International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2017, 6-8 September 2017, Marseille, France*.

Kwarter, T. (2010). *Text Mining in practice with R.* John Wiley y Sons, Ltd.

Lathia, N. K. (2010). Evaluating Collaborative Filtering Over Time. *Department of Computer Science - University College London*, 140.

Malherbe, E., Diaby, M., Cataldi, M., Viennet, E., & Aufaure, M. (2014). Field Selection for Job Categorization and Recommendation to Social Network Users. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*.

Montes, J. A. (2000). El perfil de competencias del hombre empleable. *Andalucía Económica*.

MOQUEGUA, U. N. (s.f.). *UNIVERSIDAD NACIONAL DE MOQUEGUA* . Obtenido de UNIVERSIDAD NACIONAL DE MOQUEGUA : http://www.unam.edu.pe/

Murakami, T., Mori, K., & Orihara, R. (s.f.). Metrics for Evaluating the Serendipity of Recommendation Lists. *Corporate Research and Development Center, Toshiba Corporation*, 2.

Murphy, K. P. (2012). *Machine Learning A Probabilistic Perspective.* The MIT Press Cambridge, Massachusetts London, England: LATEX programming language.

NAJAFI , S., & SALAM, Z. (2016). Evaluating Prediction Accuracy for Collaborative Filtering Algorithms in Recommender Systems. *KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION*, 37.

Nieto, S. M. (s.f.). Filtrado Colaborativo y Sistemas de Recomendación. *Universidad Calos III de Madrid*, 8.

Owen, S. (25 de 09 de 20107). *quora*. Obtenido de quora: https://www.quora.com/What-metrics-are-used-for-evaluating-recommender-systems

Piacente, P. (24 de Marzo de 2009). *IngenieriaQuimica.org*. Obtenido de IngenieriaQuimica.org

Quispe, L. V. (2016). Un enfoque probabilístico de Sistemas de Recomendación basado en contenido. *Universidad Nacional de San Agustin de Arequipa*, 94.

Rajaraman, A., & Ullman, J. D. (2014). *Recommendation Systems.* California: Cambredge University Press.

Romero Fernández., A., & Álvarez Gavilanes, J. (2015). Diagnóstico de la inserción laboral de graduados de administración de empresas de la Universidad Regional Autónoma de los Andes. *Universidad de Matanzas "Camilo Cienfuegos*, 19.

Rudliff, N. I. (2015). Sistemas De Recomendación Basados En Métodos De Filtrado Colaborativo. *UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA*, 146.

Satanovich, K. (2007). *How To Think Straight About Psychology* .

Shani , G., & Gunawardana, A. (s.f.). Evaluating Recommendation Systems. 41.

Significados. (24 de 09 de 2017). *significados*. Obtenido de significados: https://www.significados.com/item/

Su, X., & Khoshgoftaar , T. (s.f.). Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms. *Florida Atlantic University* , 8.

Torre, M. d. (2017). *Nuevas Técnicas de Minería de Textos: Aplicaciones.* Granada: DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL.

TORRES RUDLOFF, N. I. (2015). “SISTEMAS DE RECOMENDACIÓN BASADOS EN MÉTODOS DE FILTRADO COLABORATIVO”. *UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA DEPARTAMENTO DE INFORMÁTICA SANTIAGO – CHILE*, 146.

Torres, E. J. (2007). EVALUACIÓN DEL USO DE ALGORITMOS COLABORATIVOS PARA ORIENTAR ACADÉMICAMENTE AL ALUMNADO EN BACHILLERATO. *Departamento de Informática-Universidad de Jaén*, 143.

Universidad Autónoma de Chihuahua. (2010). *Estudio de seguimiento de egresados de economía internacional.*

Vásquez, A. C., Huerta, H. V., & Quispe, J. P. (2009). Procesamiento Lenguaje Natural. *Revista de Ingeniería de Sistemas e Informática vol. 6, N.º 2, Julio - Diciembre 2009*.

Verdejo, V. G. (2012). Sistemas de Recomendación: Filtrado Colaborativo . *Leganés, 7 de noviembre de 2012*, 48.

Wikipedia. (24 de 09 de 2017). *Wikipedia* . Obtenido de Wikipedia : https://es.wikipedia.org/wiki/Algoritmo

Witten, I. H. (s.f.). Text Mining. *Computer Science, University of Waikato, Hamilton, New Zealand*.

Yera Toledo, R. (2015). Métodos de preprocesamiento de datos para sistemas recomendadores de filtrado colaborativo, con aplicación en un escenario de e-learning. *UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS*

VILLAS FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN, 140.

Zanini, N., & Dhawan, V. (2015). Text Mining: An introduction to theory and some applications. *Research Matters*.

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (s.f.). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. *HP Labs, 1501 Page Mill Rd, Palo Alto, CA, 94304*.

VIII. Anexos

Los anexos se encuentran adjunto desde la siguiente página.

Matriz de consistencia:

Problema general	**Objetivo general**	**Hipótesis general**
¿Cómo estructurar un modelo de minería de datos usando técnicas de minería de textos para recomendación de ofertas laborales a partir de currículum vitae y ofertas laborales de sitios web?	Proponer un modelo de minería de datos a partir de currículum vitae y ofertas laborales de sitios web para recomendación de ofertas laborales.	Sí, es posible construir un modelo minería de datos con técnicas minería de datos para recomendación de ofertas laborales a partir de currículum vitae y ofertas laborales de sitios web.
Problemas específicos	**Objetivos específicos**	**Hipótesis específicas**
¿Cómo aplicar técnicas de minería de textos en el procesamiento de los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales?	Aplicar las técnicas de minería de textos para procesar los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales.	Es eficiente la aplicabilidad de técnicas de minería de textos para procesamiento de los currículums vitae y ofertas laborales de sitios web para construir un modelo de minería de datos para recomendación de ofertas laborales.

Recuperacion de informacion

May 22, 2019

1 Extracción de datos de los currículums vitae

El siguiente código de programación es para extraer datos de los archivos de HTML a formato CSV, el marco de trabajo BeautifulSoup permite acceder a etiquetas HTML, en donde se encuentra la información sobre los CV.

```
In [ ]: #!/usr/bin/env python
        # -*- coding: utf-8 -*-
        from bs4 import BeautifulSoup
        l= "/media/honorio/Honorio LNX/AAA/Date Base/CANDIDATEVCS/CV"
        w = ".html"
        UserCV=[]
        for i in range(20000):
            datauser=[]
            files=("{}{}{}".format(l,i,w))
            #print files
            html_file=open(files,'r')
            page=html_file.read()
            soup=BeautifulSoup(page,'html.parser')
            reviews=soup.find_all('br')
            #print reviews
            for p in reviews:
                datauser.append(p.get_text())
                #print "============================",datauser

            UserCV.append(datauser)
            #print "::::::::::::::::::::::::::::::::::::::::::::::::::",UserCV
```

```
In [ ]: import csv
        import unicodecsv as csv
        with open('CVBRexp.csv', 'w') as csvfile:
            fieldnames = ['Id_User','Content']
            writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
            writer.writeheader()
            Id_user=0
            c=0
            hono=[]
```

```
for j in UserCV:
    DataCVList=[]
    for k in j:
        #print k
        DataCVList.append(k.split())
    #print DataCVList
    ListReducida=[]
    for h in DataCVList:
        for m in h:
            ListReducida.append(m)
            #print ListReducida
    #print ListReducida
    [x.encode('UTF8') for x in ListReducida]

    Cont=0
    CVWordsUser=[]
    for i in ListReducida:
        #print ListReducida[c],c
        if Cont>45:
            #print ListReducida[c]
            CVWordsUser.append(ListReducida[Cont])
        Cont+=1
        if Cont == len(ListReducida):
            CVWordsUser.append(" ")


    #print CVWordsUser
    cadena = " ".join(ListReducida)
    cadena =[cadena]
    #print cadena
    for g in cadena:
        writer.writerow({'Id_User': Id_user, 'Content':g })
        #print Id_user, " ", g, "\n"
    Id_user+=1
```

2 Recuperación de información de las ofertas laborales

El siguiente código de programación permite extraer información pública de sitios web para fines de investigación, nos permite acceder a los dominios de nuestro interes para extraer información que se encuentra incrustada en las etiquetas HTML.

```
In [ ]: # import sys
        # reload(sys)
        # sys.setdefaultencoding('utf8')

        import scrapy
        from scrapy.spider import CrawlSpider, Rule
```

```
from scrapy.linkextractors import LinkExtractor
from scrapy.exceptions import CloseSpider
from mercado.items import MercadoItem

class MercadoSpider(CrawlSpider):
        name = 'mercado'
        item_count = 0
        allowed_domain = ['www.dominio.com.pe']
        start_urls = ['https://www.dominio.com.pe/ofertas-de-trabajo/']

        rules = {
                # Para cada item
                Rule(LinkExtractor(allow = (),
                           restrict_xpaths = ('//li[@class="siguiente"]/a'))),
                Rule(LinkExtractor(allow =(),
                           restrict_xpaths = ('//h2[@class="tO"]/a')),
                                callback = 'parse_item', follow = False)
        }

        def parse_item(self, response):
                ml_item = MercadoItem()
                #info de producto
                ml_item['titulo'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/header/h1/text()'
        ).extract()
                ml_item['empresa'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/div[1]/div[2]/h2/text()'
        ).extract()
                ml_item['ciudad'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/header/p/span[1]/text()'
        ).extract()
                ml_item['descripcion'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[1]/text()'
        ).extract()
                ml_item['requisitos'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[1]/text()'
        ).extract()
                ml_item['funciones'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[5]/text()'
        ).extract()
                ml_item['educacion'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[1]/div[2]/ul/li[4]/text()'
        ).extract()
                ml_item['conocimiento'] = response.xpath(
            '//*[@id="MainContainer"]/article/section[4]/ul/li[5]/p/text()'
        ).extract()

                self.item_count += 1
```

```
                    if self.item_count > 10000:
                            raise CloseSpider('item_exceeded')
                    yield ml_item
```

```
In [ ]: # -*- coding: utf-8 -*-

        # Define here the models for your scraped items
        #
        # See documentation in:
        # http://doc.scrapy.org/en/latest/topics/items.html

        import scrapy


        class MercadoItem(scrapy.Item):
            # define the fields for your item here like:
            # name = scrapy.Field()

            #info de producto
            titulo = scrapy.Field()
            empresa = scrapy.Field()
            ciudad = scrapy.Field()
            descripcion = scrapy.Field()
            requisitos = scrapy.Field()
            funciones = scrapy.Field()
            educacion = scrapy.Field()
            conocimiento = scrapy.Field()
```

2.1 Exportación de datos de las Ofertas Laborales

El siguiente código de programación permite exportar los datos extraídos del formato HTML al formato CSV, de tal manera que se pueda usar más adelante en el procesamiento de lenguaje natural.

```
In [ ]: # -*- coding: utf-8 -*-

        import scrapy
        from scrapy import signals
        from scrapy.exporters import CsvItemExporter
        from scrapy.pipelines.images import ImagesPipeline
        from scrapy.exceptions import DropItem
        from scrapy import Request
        import csv

        class MercadoPipeline(object):
            def __init__(self):
                self.files = {}
```

```
@classmethod
def from_crawler(cls, crawler):
    pipeline = cls()
    crawler.signals.connect(pipeline.spider_opened, signals.spider_opened)
    crawler.signals.connect(pipeline.spider_closed, signals.spider_closed)
    return pipeline

def spider_opened(self, spider):
    file = open('%s_items.csv' % spider.name, 'w+b')
    self.files[spider] = file
    self.exporter = CsvItemExporter(file)
    self.exporter.fields_to_export = ['titulo', 'empresa', 'ciudad',
                                      'descripcion','requisitos',
                                      'funciones', 'educacion','conocimiento']
    self.exporter.start_exporting()

def spider_closed(self, spider):
    self.exporter.finish_exporting()
    file = self.files.pop(spider)
    file.close()

def process_item(self, item, spider):
    self.exporter.export_item(item)
    return item
```

Procesamiento Lenguaje Natural & Extraccion de Datos

May 26, 2019

1 Procesamiento Lenguaje Natural

1.1 Importación de librerias NLTK

Importamos librerías necesarias para la limpieza de datos con NLTK.

```
In [1]: import nltk
        import re, string, unicodedata
        import inflect
        from bs4 import BeautifulSoup
        from nltk import word_tokenize, sent_tokenize
        from nltk.corpus import stopwords
        from nltk.stem import LancasterStemmer, WordNetLemmatizer
        import math
```

1.2 Funciones de limpieza de datos con NLTK

Se crean funciones para cada propósito, por ejemplo: para eliminar caracteres que no son de ASCII, para convertir todos los términos en minúsculas para que todos sean iguales, eliminación de puntuaciones y acentos, etc. Todo este codigo esta bajo la función CleanWOrds.

```
In [2]: def CleanWOrds(words,user,Clean_Words):

            def remove_non_ascii(words):
                #"""Elimine los caracteres que no son ASCII de la lista de palabras"""
                new_words = []
                for word in words:
                    new_word = unicodedata.normalize(
                        'NFKD', word).encode('ascii', 'ignore').decode('utf-8', 'ignore')
                    new_words.append(new_word)
                return new_words
            def to_lowercase(words):
            #"""Convierte todos los caracteres a minúsculas de la lista de palabras """
                new_words = []
                for word in words:
                    new_word = word.lower()
                    new_words.append(new_word)
                return new_words
```

```
def remove_punctuation(words):
    #"""Eliminar puntuación de la lista de palabras tokenizadas"""
    new_words = []
    for word in words:
        new_word = re.sub(r'[^\w\s]', '', word)
        if new_word != '':
            new_words.append(new_word)
    return new_words

def replace_numbers(words):
    #"""Reemplace todas las apariciones de enteros en la lista
    #de palabras con token con representación textual"""
    p = inflect.engine()
    new_words = []
    for word in words:
        if word.isdigit():
            new_word = p.number_to_words(word)
            #new_words.append(new_word)
        else:
            new_words.append(word)
    return new_words

def remove_stopwords(words):
    #"""Eliminar palabras vacias de la lista de palabras en token"""
    new_words = []
    for word in words:
        if word not in stopwords.words('spanish'):
            new_words.append(word)
    return new_words


def lemmatize_verbs(words):
    #"""Lemmatizar verbos en lista de palabras tokenizadas"""
    lemmatizer = WordNetLemmatizer()
    lemmas = []
    for word in words:
        lemma = lemmatizer.lemmatize(word, pos='v')
        lemmas.append(lemma)
    return lemmas

def normalize(words):
    words = remove_non_ascii(words)
    words = to_lowercase(words)
    words = remove_punctuation(words)
    words = replace_numbers(words)
    words = remove_stopwords(words)
    words = lemmatize_verbs(words)
```

```
            return words
        words = normalize(words)
        #print (user,"\n",words)

        Clean_Words=Clean_Words.append(words)
        return (words,user,Clean_Words)
```

1.3 Importación de datos de los currículums vitae

Se importa datos de currículums vitaes.

```
In [3]: import pandas as pd
        CV_Data = pd.read_csv("../BD/CurriculumVitaes(5k).csv",
                              delimiter=",", error_bad_lines=False, encoding="utf-8")
        CV_Data.head()
```

```
Out[3]:    Id_User                            Nombre  \
        0        0  Miguel Angel Garcia del Castillo
        1        1    Jesús Augusto Valqui Barraza
        2        2          Lizbeth Huaman Ventura
        3        3                    Jorge Romero
        4        4           Marjori Santos Sedano

                                              Descripcion  \
        0  Soy un profesional capacitado para llevar adel...
        1  Cursante del IV ciclo de Administración de Neg...
        2  Soy una joven con grandes expectativas, respon...
        3  Abogado con cuatro años de experiencia profesi...
        4  En ciencias de la comunicaciones cursando el 4...

                                              Experiencia  \
        0  publicaciones sobre sistema de recomendacion y...
        1
        2  CONCAR S.A. Grupo Graña y Montero Desde Mayo 2...
        3  PerfilSoy una persona 100% creativa actitud pr...
        4

                                              Habilidades
        0                    python, pln, nltk, scrapy, tec
        1  Asociación Civil Sol y Mar /  Club Sol y Mar  ...
        2  Concar s.a. Grupo Graña y Montero / Informátic...
        3  ESTUDIO JURÍDICO ROMERO / Legal / Asesoría / L...
        4  Delosi / Atención a clientes / Venta al consum...
```

1.4 Lectura de documentos

Descripción, experiencia y habilidades son los campos que se importan para el tratamiento de información, en dichos campos hay un conjunto de términos escritos en forma textual referente a

la descripción, habilidades y experiencia de los candidatos. Cada campo de datos son importadas en tres listas diferentes por candidato, es decir cada uno de los términos se lee como elementos en tres listas por candidato (descripción, experiencia y habilidades).

```
In [4]: for j in CV_Data:
            CV_Descripcion = CV_Data.Descripcion.str.split(' ')
            CV_Experiencia = CV_Data.Experiencia.str.split(' ')
            CV_Habilidades = CV_Data.Habilidades.str.split(' ')
        #print CV_Descripcion
```

1.5 Limpieza de datos de los currículums vitae

Cada una de las listas generadas en el anterior punto (descripción, experiencia y habilidades) pasan sin excepción por las funciones de limpieza de datos con NLTK.

A continuación, primero se hace lectura a las listas de términos por separado y en seguida se ejecuta a la función leanWOrds, a la que se adjunta las listas de los términos en la variable Words, clave principal del usuario de adjunta en la variable user y finalmente se envía una variable de vector vacío Clean_Words en la que devuelve los términos limpios.

```
In [5]: #Documento A
        user=0
        Clean_Words=[]
        for i in CV_Descripcion:
            UserList=[]
            data = ' '.join(i)
            words = nltk.word_tokenize(data)
            CleanWOrds(words,user,Clean_Words)
            CV_Description_list=Clean_Words
            user+=1
        #Documento B
        user=0
        Clean_Words=[]
        for i in CV_Experiencia:
            UserList=[]
            data = ' '.join(i)
            words = nltk.word_tokenize(data)
            CleanWOrds(words,user,Clean_Words)
            CV_Experiencia_list=Clean_Words
            user+=1
        #Documento C
        user=0
        Clean_Words=[]
        for i in CV_Habilidades:
            UserList=[]
            data = ' '.join(i)
            words = nltk.word_tokenize(data)
            CleanWOrds(words,user,Clean_Words)
            CV_Habilidades_list=Clean_Words
            user+=1
```

1.6 Importación de los Ofertas laborales

Se importan datos de ofertas laborales recolectados de setios web.

```
In [6]: import pandas as pd
        Job_Data = pd.read_csv("../BD/PuestosLaborales2.csv",
                               delimiter=",", error_bad_lines=False, encoding="utf-8")
        Job_Data.head()
```

Out[6]:

	Id_Job	Titulo	Empresa	Ciudad	Descripcion	Requisitos	Funciones	Educacion	Conocimientos
0	0	Investigador	Honorio's analytics	San Isidro Lima	Se necesita un profesional muy preparado para ...	Conocimientos en procesamiento lenguaje natura...	Desarrollador encargado del proyecto de sistem...	Universitario	python, scrapy, procesamiento lenguaje natural...
1	1	Vendedores rotativos para tiendas Baby Infanti...	SGA Consultores SAC	Miraflores Lima	Por encargo de las Tiendas Baby Infanti y Lego...	Secundaria completa. Manejo fluido a nivel usu...		Educación Secundaria	
2	2	Asistente Contable Técnico o Universitario Egr...	Adecco Perú S.A.	San Borja Lima	Por encargo de nuestro cliente empresa líder ...		Análisis de depreciaciones. Asientos contables...	Técnico	Microsoft Excel
3	3	Practicante de Administración (Apoyo)	Industrias Plasticas Reunidas S.A.C	Lima Lima	Industrias Plásticas Reunidas es una empresa 1...		Atención directa a gerencia en relación a coor...	Técnico	
4	4	Jefe de Producción	NaN	NaN	Importante empresa del sector se encuentra en ...		Supervisar y garantizar el cumplimiento del Pl...	Universitario	

```
3                                   Microsoft Excel
4                                   Microsoft Excel
```

1.7 Lectura de documentos de ofertas laborales

En este punto se importan datos de los campos: Titulo, Descripción, Requisitos, Funciones y Conocimientos, las cuales serán denominados: documento A, documento B, documento C, documento D y documento E respectivamente. Cada campo es importado en listas diferentes por documento, esto significa que los términos son elementos de las listas mencionadas.

```
In [7]: for i in Job_Data:
            Job_titulo = Job_Data.Titulo.str.split(' ')
            Job_Description = Job_Data.Descripcion.str.split(' ')
            Job_Requisitos = Job_Data.Requisitos.str.split(' ')
            Job_Funciones = Job_Data.Funciones.str.split(' ')
            Job_Conocimientos = Job_Data.Conocimientos.str.split(' ')
        #print Job_Funciones
```

1.8 limpieza de datos de las ofertas laborales

Toda la información de ofertas laborales convertidas en cinco listas diferentes pasan por la función CleanWOrds progrmando para limpieza de datos con NLTK. Todo el resto es igual al proceso de limpieza de datos de los currículums vitae.

```
In [8]: #Documento A
        user=0
        Clean_Words=[]
        for i in Job_titulo:
            UserList=[]
            data = ' '.join(i)
            words = nltk.word_tokenize(data)
            CleanWOrds(words,user,Clean_Words)
            Titulo_list=Clean_Words
            user+=1
        #Documento B
        user=0
        Clean_Words=[]
        for i in Job_Description:
            UserList=[]
            data = ' '.join(i)
            words = nltk.word_tokenize(data)
            CleanWOrds(words,user,Clean_Words)
            Description_list=Clean_Words
            user+=1
        #Documento C
        user=0
        Clean_Words=[]
        for i in Job_Requisitos:
```

```
        UserList=[]
        data = ' '.join(i)
        words = nltk.word_tokenize(data)
        CleanWOrds(words,user,Clean_Words)
        Requisitos_list=Clean_Words
        user+=1
    #Documento D
    user=0
    Clean_Words=[]
    for i in Job_Funciones:
        UserList=[]
        data = ' '.join(i)
        words = nltk.word_tokenize(data)
        CleanWOrds(words,user,Clean_Words)
        Funciones_list=Clean_Words
        user+=1
    #Documento E
    user=0
    Clean_Words=[]
    for i in Job_Conocimientos:
        UserList=[]
        data = ' '.join(i)
        words = nltk.word_tokenize(data)
        CleanWOrds(words,user,Clean_Words)
        Conocimientos_list=Clean_Words
        user+=1
```

1.9 Concatenación de documentos de currículums vitae

A continuación, todos los documentos se concatenan en una sola lista por oferta laboral, con el objetivo reducir el tiempo de procesamiento y redundancia de datos en las listas. Y en el siguiente punto se eliminan los términos que se repiten más de una vez y se guardan en una sola lista que es: CorpusjobUnduplicated.

```
In [9]: #Concatenacion de documentos en una sola lista
        JobCorpus=[]
        IdJobCorpus=[]
        Job=0
        for i in Job_titulo:

            A=Titulo_list[Job]
            B=Description_list[Job]
            C=Requisitos_list[Job]
            D=Funciones_list[Job]
            E=Conocimientos_list[Job]

            Job+=1
```

```
            for a in A:
                JobCorpus.append(a)
            for b in B:
                JobCorpus.append(b)
            for c in C:
                JobCorpus.append(c)
            for d in D:
                JobCorpus.append(d)
            for e in E:
                JobCorpus.append(e)

            IdJobCorpus.append(JobCorpus)
            JobCorpus=[]
```

```
In [10]: #ELiminacion de terminos repetidos
         CorpusjobUnduplicated=[]
         for i in IdJobCorpus:
             j=set(i)
             CorpusjobUnduplicated.append(j)
```

1.10 Calculando valor de relevancia de cada CV respecto a cada Oferta laboral con TFIDF

El siguiente código es del algoritmo TF-IDF, el cual nos permite calcular la relevancia de términos de un CV con respecto a las ofertas laborales, está diseñado y programado de acuerdo al caso, como resultado tenemos valores promedios de tf-idf de manera detallada para cada documento de CV (Descripción, Experiencia y Habilidades) y otra general, esta última es exportado como valor rating de un cv con respecto a las ofertas laborales. Incluso se puede obtener la lista de términos relevantes de cada uno de estos documentos. La función para ejecutar y calcular los valores tf-idf es: TFIDFCV.

```
In [11]: def TFIDFCV(
             campo,CV_Description_list,CV_Experiencia_list,CV_Habilidades_list,TFIDF_WORD):
             JOB=0
             for u in campo:
                 #print u
                 sumA=0
                 sumB=0
                 sumC=0
                 extranho=[]
                 for x in range(len(CV_Description_list)):
                     A=CV_Description_list[x]
                     B=CV_Experiencia_list[x]
                     C=CV_Habilidades_list[x]

                     WordList=[]
                     WordListA=[]
                     WordListB=[]
```

```
WordListC=[]
for q in u:
    ContDoc=0
    ContA=0
    ContB=0
    ContC=0

    for a in A:
        if q==a:
            ContA+=1
    if len(A)==0:
        TFA=0
    else:
        TFA=ContA/float(len(A))
    for b in B:
        if q==b:
            ContB+=1
    if len(B)==0:
        TFB=0
    else:
        TFB=ContB/float(len(B))
    for c in C:
        if q==c:
            ContC+=1
    if len(C)==0:
        TFC=0
    else:
        TFC=ContC/float(len(C))

    if ContA>=1:
        ContDoc+=1
    if ContB>=1:
        ContDoc+=1
    if ContC>=1:
        ContDoc+=1

    if ContDoc!=0:
        product=(3/float(ContDoc))
        IDF=math.log(product)
    else:
        IDF=0


    TFIDF_A=TFA*IDF
    TFIDF_B=TFB*IDF
    TFIDF_C=TFC*IDF
    #print (TFIDF_B)
    #print "Valor TF-IDF para Doc B =>",TFIDF_B
```

```
        TFIDF_WORD_A=str(round(TFIDF_A, 3))
        TFIDF_WORD_B=str(round(TFIDF_B, 3))
        TFIDF_WORD_C=str(round(TFIDF_C, 3))

        #print q," >>>>>>>>>>>>>>>>>>>>",TFIDF_WORD

        #WordList.append(q)
        if TFIDF_A>0:
            WordListA.append(q)
            WordListA.append(TFIDF_WORD_A)
            sumA=sumA+float(TFIDF_WORD_A)
        if TFIDF_B>0:
            WordListB.append(q)
            WordListB.append(TFIDF_WORD_B)
            sumB=sumB+float(TFIDF_WORD_B)
        if TFIDF_C>0:
            WordListC.append(q)
            WordListC.append(TFIDF_WORD_C)
            sumC=sumC+float(TFIDF_WORD_C)


if len(WordListA)>0:
    PromedioA=str(round(sumA/(len(WordListA)/2), 3))
    #PromedioA=str(round(sumA/(len(WordListA)/2), 3))
else:
    PromedioA=0
if len(WordListB)>0:
    PromedioB=str(round(sumB/(len(WordListB)/2), 3))
else:
    PromedioB=0
if len(WordListC)>0:
    PromedioC=str(round(sumC/(len(WordListC)/2), 3))
else:
    PromedioC=0

sumA=0
sumB=0
sumC=0

XX=(float(PromedioA)+float(PromedioB)+float(PromedioC))/3
promediogeneral=str(round(XX, 3))
extranho.append(JOB)
extranho.append(x)
#extranho.append(promediogeneral)
extranho.append(PromedioA)
extranho.append(PromedioB)
extranho.append(PromedioC)
```

```
            #print ("\n")
        JOB+=1
        OferList.append(extranho)
        #print OferList
    return OferList
```

1.11 Ejecución de función anterior TFIDFCV

El siguiente código es la que ejecuta la función anterior (TFIDFCV), también envía datos como: listas de términos de CV y ofertas laborales (campo, CV_Description_list, CV_Experiencia_list, CV_Habilidades_list,OferList) para calcular su valor de relevancia.

Dónde:

campo: es la variable en la que se adjunta las listas de términos de las ofertas laborales.

CV_Description_list,CV_Experiencia_list,CV_Habilidades_list: Son listas de términos de entrada de currículums vitae.

OferList: Es la variable en la que retorna listas de términos y susterpectivas valores if-idf.

```
In [12]: OferList=[]
         TFIDF_OferDescription_list=[]
         campo=CorpusjobUnduplicated
         TFIDFCV(
             campo,CV_Description_list,CV_Experiencia_list,CV_Habilidades_list,OferList)
         TFIDF_CVwords_list=OferList
```

2 Extracción de datos

Con el siguiente código exportamos los resultados obtenidos en el anterior punto, las misma se rencuentran guardadas en la variable (TFIDF_CVwords_list). Tenemos dos tipos de exportación: la primera que exporta promedios de TFIDF detallados por cada documento (descripción, experiencia y habilidades) de currículums vitae y la segunda parte nos permite exportar el promedio general de tfidf. Es decir promedio de promedios de if-idf de documentos respectivos.

```
In [13]: #rtacion de rating detallados por documentos
         import csv
         with open('RatingDetail_CV_JOB180519.csv', 'w') as csvfile:
             fieldnames = ['Id_Cv','Id_Job','Rating_Descripcion',
                           'Rating_Experiencia','Rating_Habilidades']
             writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
             writer.writeheader()

             for i in TFIDF_CVwords_list:
                 word=0
                 for j in range(len(i)/5):
                     Id_CV=i[word]
                     Id_Job=i[word+1]
                     #Rating_general=i[word+2]
                     Rating_descripcion=i[word+2]
                     Rating_experiencia=i[word+3]
```

```
                    Rating_habilidades=i[word+4]
                    writer.writerow({'Id_Cv': Id_CV,
                                     'Id_Job':Id_Job,
                                     'Rating_Descripcion':Rating_descripcion,
                                     'Rating_Experiencia':Rating_experiencia,
                                     'Rating_Habilidades':Rating_habilidades })
                    cont=word+4
                    word=cont+1
```

In [48]:
```
#exportacion de rating generales
import csv
with open('RatingGeneral_CV_JOBS180519.csv', 'w') as csvfile:
    fieldnames = ['Id_Job','Id_Cv','Rating_general']
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()

    for i in TFIDF_CVwords_list:
        word=0
        for j in range(len(i)/3):
            Id_Job=i[word]
            Id_Cv=i[word+1]
            Rating_general=i[word+2]
            writer.writerow({'Id_Job': Id_Job,
                             'Id_Cv':Id_Cv,
                             'Rating_general':Rating_general })
            cont=word+2
            word=cont+1
```

Mineria de Datos & Visualizacion de Resultados

May 26, 2019

1 Importación de datos

En los siguientes líneas de código de programación se importa los datos de promedio tf-idf calculados en el anterior punto (procesamiento lenguaje natural y extracción de datos), los cuales nos permiten hacer un análisis y visualización de resultados de manera más comprensible.

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        df = pd.read_csv("RatingGeneral_CV_JOBS180519P.csv")

In [2]: ratings_df = pd.DataFrame(df, columns = ['Id_Job', 'Id_Cv', 'Rating_general'])

In [3]: ratings_df.head()
```

Out[3]:

	Id_Job	Id_Cv	Rating_general
0	0	0	0.092
1	0	1	0.000
2	0	2	0.004
3	0	3	0.002
4	0	4	0.012

A continuación, convertimos en una matriz de los promedios tf-idf, los valores de relevancia calculados en base a los terminos de los CV con relación a cada oferta laboral, los cuales se consideran como valor rating.

Teniendo en cuenta que la primera columna que es "Id_Job" corresponde la clave principal de cada uno de las ofertas laborales y el encabezado de las columnas o campos desde la segunda hasta 4999 son los claves principales de los CV.

```
In [4]: R_df = ratings_df.pivot(
            index = 'Id_Job', columns ='Id_Cv', values = 'Rating_general').fillna(0)
        R_df.head(10)
```

Out[4]:

Id_Cv	0	1	2	3	4	5	6	7	8	9	\
Id_Job											
0	0.092	0.000	0.004	0.002	0.012	0.002	0.006	0.000	0.009	0.002	
1	0.000	0.008	0.013	0.013	0.037	0.023	0.021	0.029	0.040	0.009	
2	0.000	0.023	0.013	0.030	0.024	0.034	0.033	0.047	0.015	0.018	
3	0.000	0.023	0.013	0.030	0.048	0.041	0.030	0.051	0.032	0.013	

```
4       0.000  0.025  0.013  0.030  0.033  0.035  0.035  0.047  0.032  0.013
5       0.000  0.022  0.013  0.030  0.030  0.043  0.035  0.034  0.024  0.017
6       0.000  0.005  0.001  0.001  0.033  0.009  0.008  0.006  0.018  0.013
7       0.017  0.017  0.025  0.001  0.012  0.003  0.006  0.004  0.024  0.006
8       0.000  0.030  0.013  0.032  0.025  0.032  0.025  0.047  0.023  0.022
9       0.000  0.006  0.001  0.001  0.028  0.010  0.008  0.006  0.014  0.012

Id_Cv    ...   4990   4991   4992   4993   4994   4995   4996   4997   4998  \
Id_Job   ...
0        ...  0.003  0.001  0.006  0.031  0.000  0.002  0.000  0.015  0.001
1        ...  0.041  0.038  0.030  0.058  0.030  0.001  0.049  0.011  0.012
2        ...  0.093  0.002  0.022  0.029  0.031  0.014  0.107  0.012  0.001
3        ...  0.094  0.002  0.022  0.063  0.039  0.014  0.058  0.017  0.001
4        ...  0.094  0.001  0.025  0.060  0.051  0.013  0.051  0.015  0.001
5        ...  0.094  0.038  0.026  0.026  0.050  0.014  0.084  0.014  0.001
6        ...  0.003  0.001  0.017  0.026  0.002  0.001  0.031  0.002  0.001
7        ...  0.002  0.001  0.011  0.050  0.002  0.001  0.055  0.003  0.001
8        ...  0.093  0.002  0.029  0.069  0.040  0.014  0.086  0.018  0.004
9        ...  0.003  0.001  0.019  0.029  0.002  0.001  0.023  0.002  0.001

Id_Cv    4999
Id_Job
0       0.000
1       0.004
2       0.002
3       0.002
4       0.002
5       0.003
6       0.003
7       0.005
8       0.003
9       0.007

[10 rows x 5000 columns]
```

En seguida extraemos una muestra de 10 primeros CV y 10 primeros ofertas laborales para una visualización clara de resultados y análisis del mismo.

```
In [5]: invoices = pd.DataFrame(R_df,columns = [0,1,2,3,4,5,6,7,8,9,10])
        invoices.head(10)
```

```
Out[5]:            0      1      2      3      4      5      6      7      8      9   \
        Id_Job
        0       0.092  0.000  0.004  0.002  0.012  0.002  0.006  0.000  0.009  0.002
        1       0.000  0.008  0.013  0.013  0.037  0.023  0.021  0.029  0.040  0.009
        2       0.000  0.023  0.013  0.030  0.024  0.034  0.033  0.047  0.015  0.018
        3       0.000  0.023  0.013  0.030  0.048  0.041  0.030  0.051  0.032  0.013
        4       0.000  0.025  0.013  0.030  0.033  0.035  0.035  0.047  0.032  0.013
```

```
5       0.000  0.022  0.013  0.030  0.030  0.043  0.035  0.034  0.024  0.017
6       0.000  0.005  0.001  0.001  0.033  0.009  0.008  0.006  0.018  0.013
7       0.017  0.017  0.025  0.001  0.012  0.003  0.006  0.004  0.024  0.006
8       0.000  0.030  0.013  0.032  0.025  0.032  0.025  0.047  0.023  0.022
9       0.000  0.006  0.001  0.001  0.028  0.010  0.008  0.006  0.014  0.012

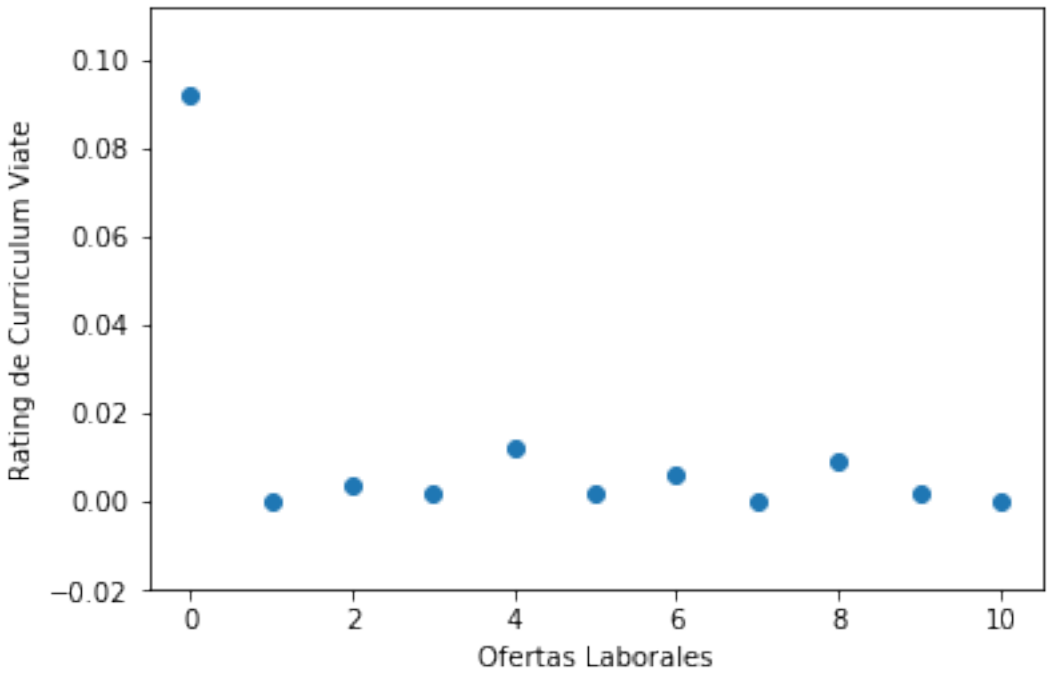
           10
Id_Job
0       0.000
1       0.019
2       0.016
3       0.022
4       0.016
5       0.016
6       0.014
7       0.014
8       0.016
9       0.049
```

2 Visualización de datos

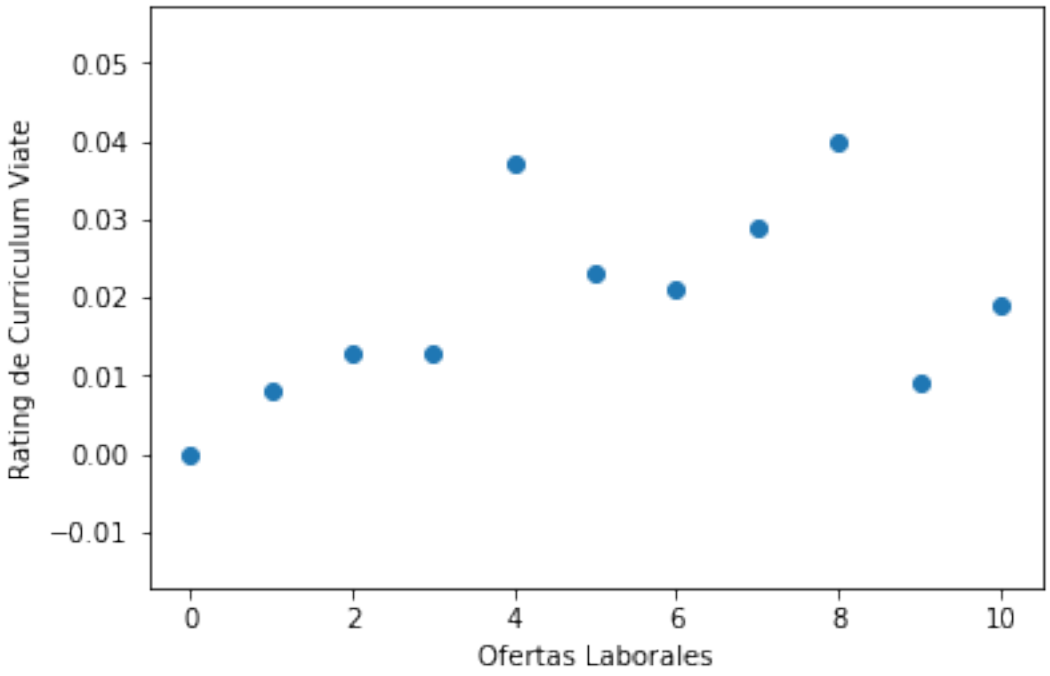
A continuación, visualizaremos algunos datos en gráficas. En el siguiente grafico se muestra la relevancia de un CV en 10 ofertas laborales, la relevancia está basada en términos o palabras escritas en los currículums vitae, de tal manera que se puede observar que algunos son muy relevantes, mientras que otros tienen poco valor de relevancia.

```
In [9]: import numpy as np
        a=np.array(invoices)
        coont=0
        for i in a:
            print ("cv Nro", coont)
            y=i
            x = np.arange(len(i))
            plt.xlabel("Ofertas Laborales")
            plt.ylabel("Rating de Curriculum Viate")
            plt.scatter(x, y)
            plt.show()
            coont+=1
            if coont>=2:
                break

('cv Nro', 0)
```

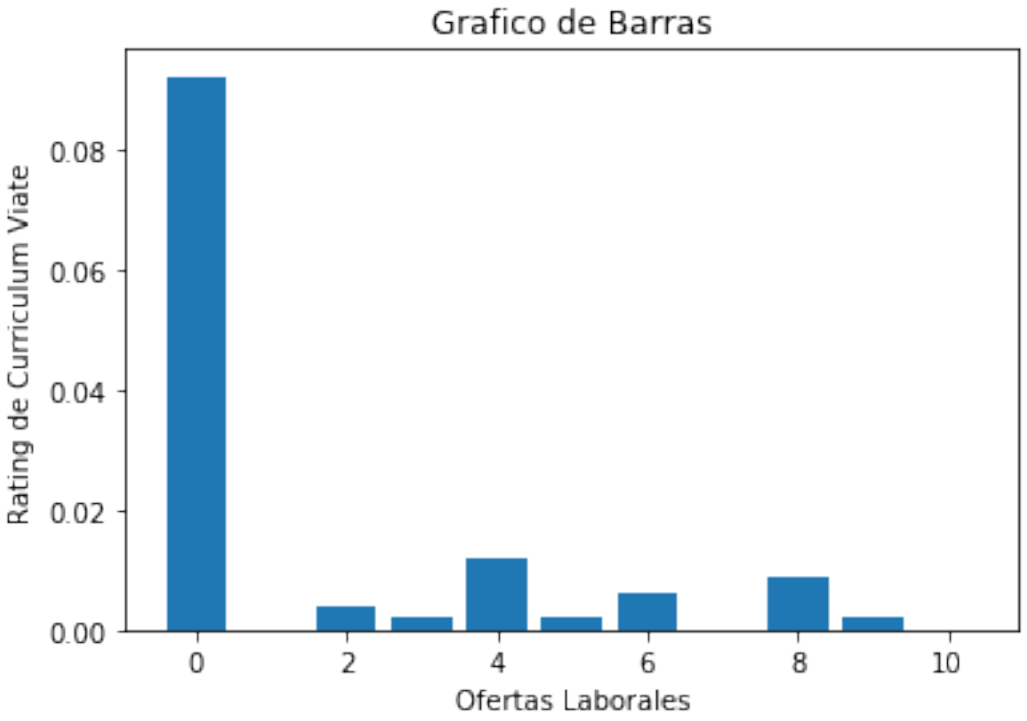

Rating de Curriculum Viate
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
0
2
4
6
8
10
Ofertas Laborales

```
('cv Nro', 1)
```

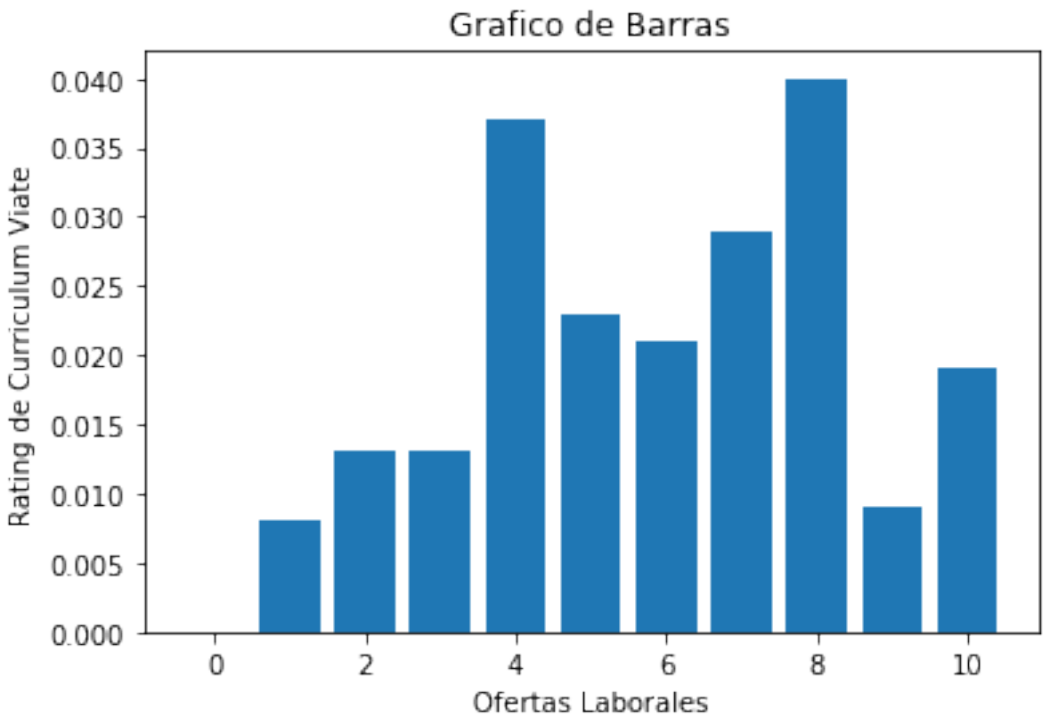

Rating de Curriculum Viate
0.05
0.04
0.03
0.02
0.01
0.00
−0.01
0
2
4
6
8
10
Ofertas Laborales

```
In [11]: coont=0
         for i in a:
             print ("cv Nro", coont)
             #print (i)
             y=i
             x = np.arange(len(i))
             plt.xlabel("Ofertas Laborales")
             plt.ylabel("Rating de Curriculum Viate")
             plt.bar(x, y, align="center")
             plt.title("Grafico de Barras")
             plt.show()
             coont+=1
             if coont>=2:
                 break
```

```
('cv Nro', 0)
```

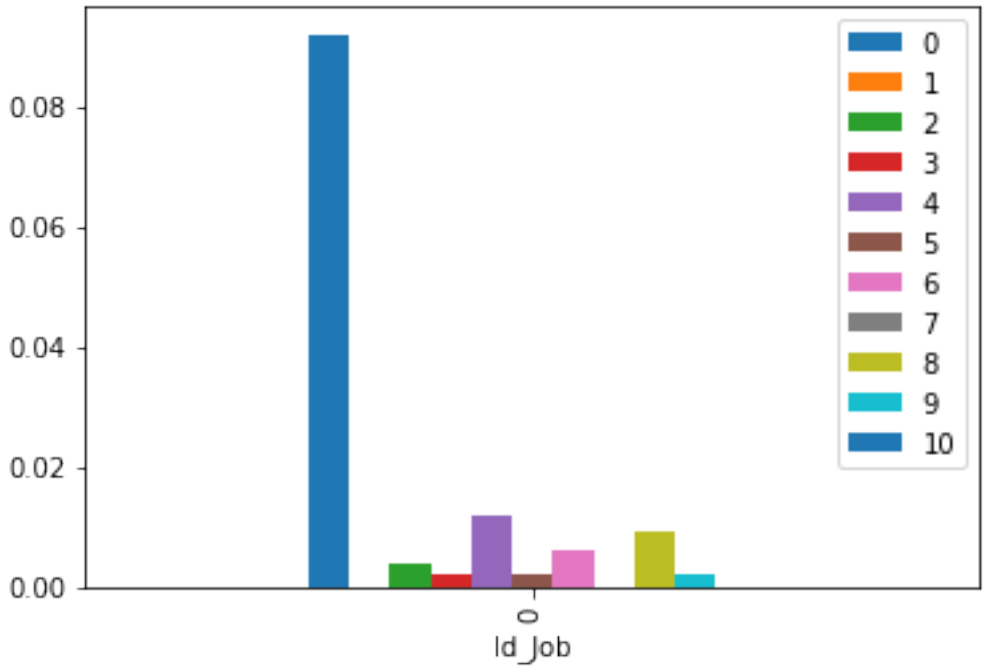


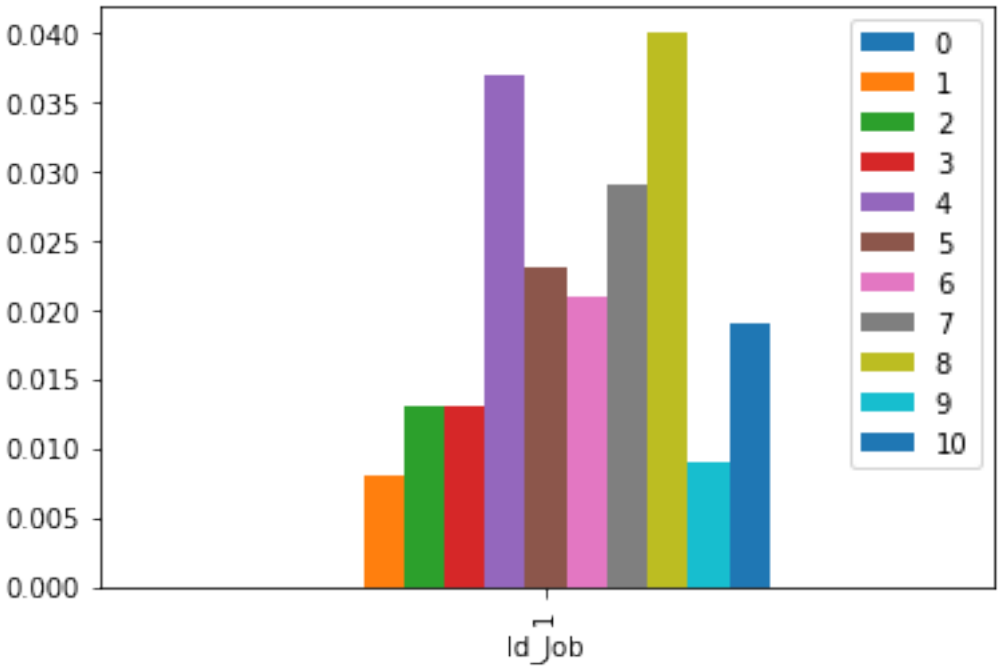

```
('cv Nro', 1)
```

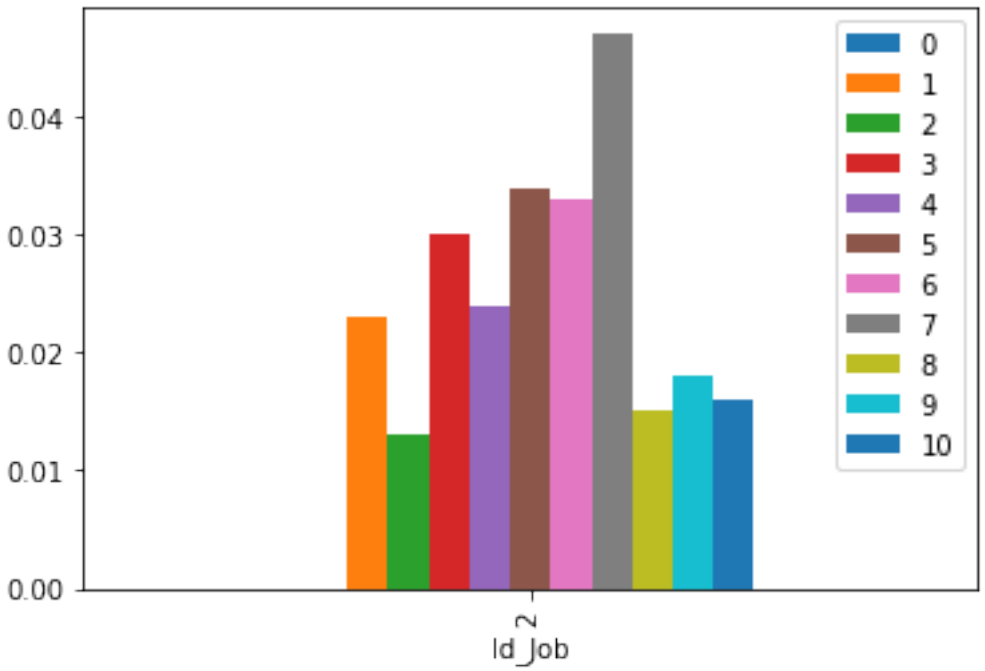


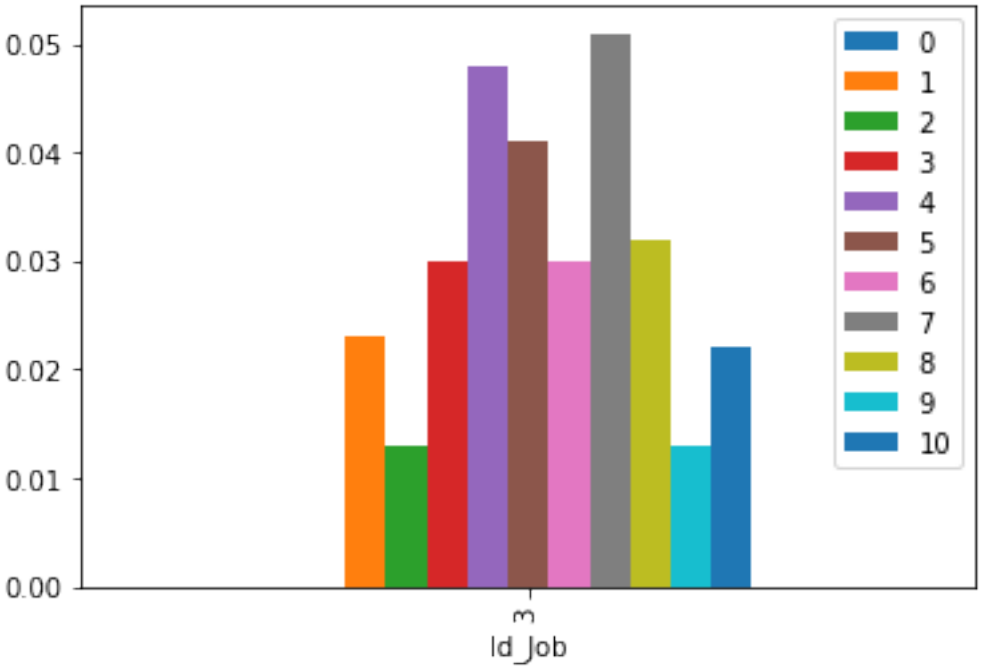
En seguida, se aprecia de forma grafica la relevancia de los 10 primeros CV y ofertas laborales.

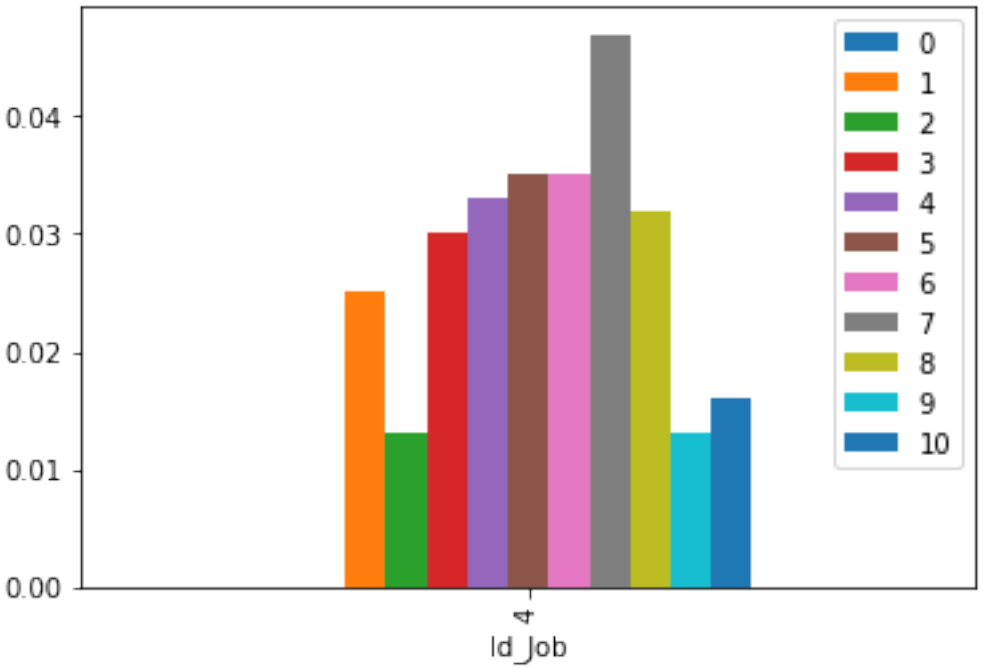
```
In [8]: plotbar=invoices.groupby(['Id_Job',0,1,2,3,4,5,6,7,8,9,10])
        plotbar.plot(kind='bar')
        plt.show()
```

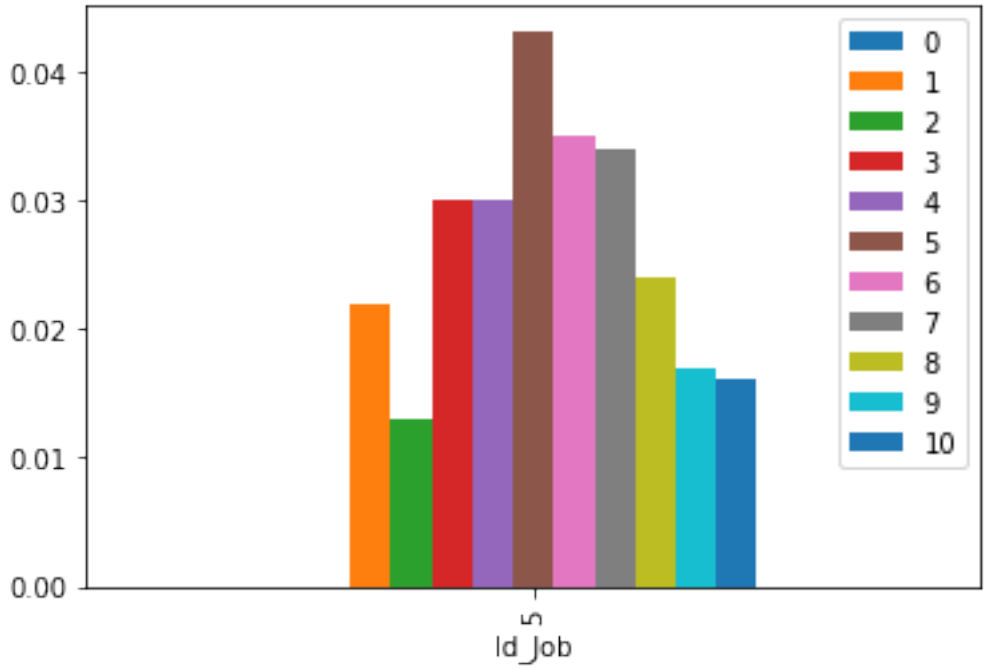


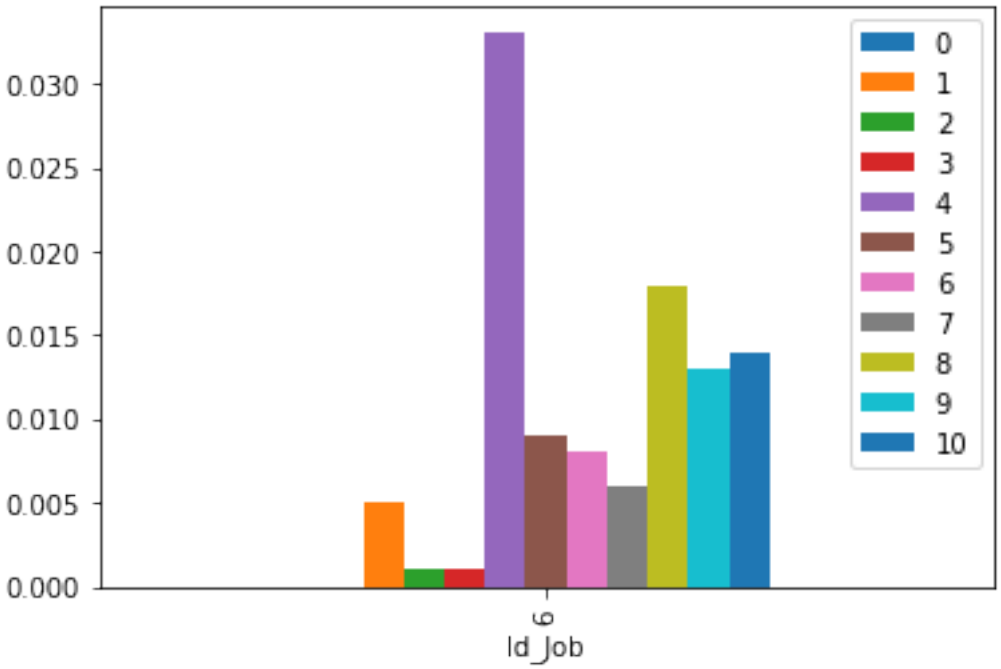

0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1
2
3
4
5
6
7
8
9
10
Id_Job

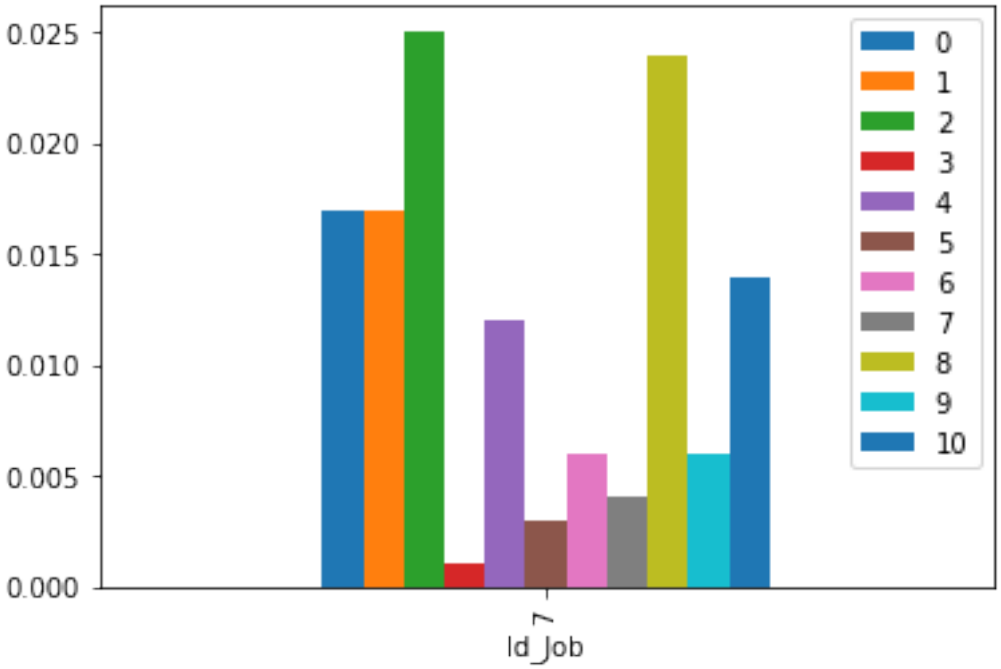

0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
Id_Job

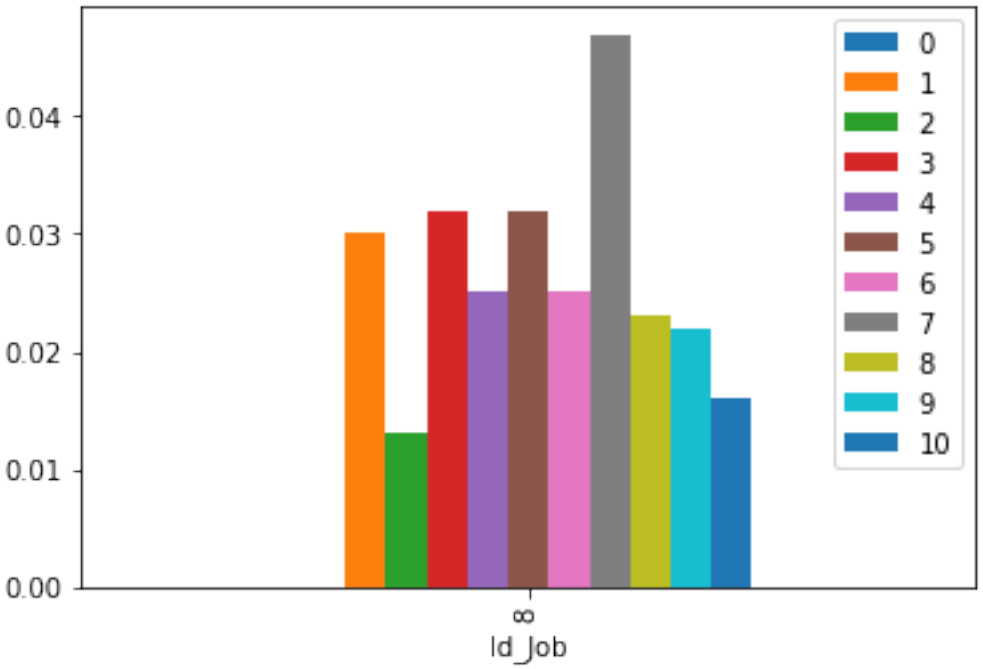

0.05
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
3
Id_Job

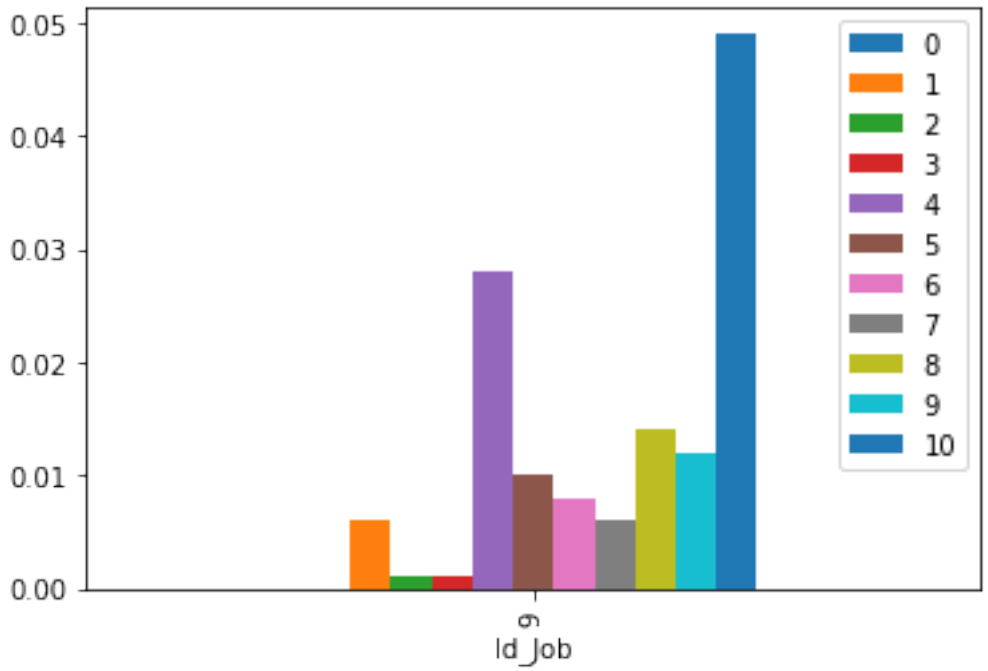

0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
4
Id_Job

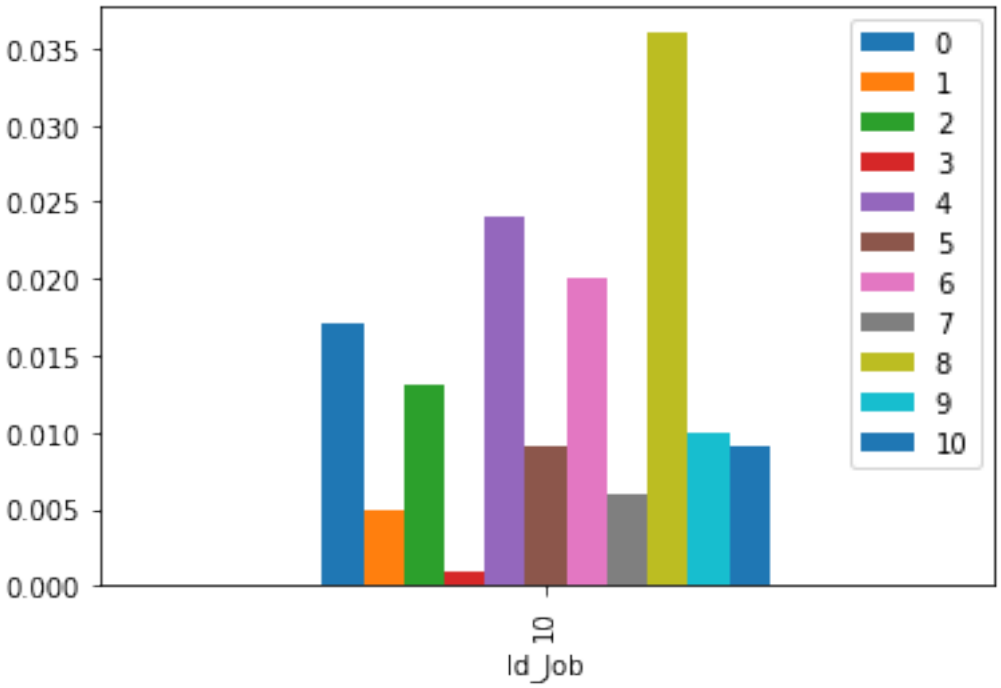

0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
Id_Job
5


0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1
2
3
4
5
6
7
8
9
10
Id_Job
6


0.025
0.020
0.015
0.010
0.005
0.000
0
1
2
3
4
5
6
7
8
9
10
Id_Job
7


0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
Id_Job
8


0.05
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
9
10
6
Id_Job


0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
1
2
3
4
5
6
7
8
9
10
10
Id_Job

Estructuracion de datos para recomendacion de ofertas laborales

May 26, 2019

1 Preparación de datos para recomendación de ofertas laborales

Los siguientes líneas de código es para estructurar el modelo de datos para la recomendación de ofertas laborales, el modelo está inspirada en Dataset de libros, seleccionado por Cai-Nicolas Ziegler.

A continuación, se importan datos de rating tf-idf generadas en la etapa de procesamiento lenguaje natural, en seguida también se importan datos originales de currículums vitae y finalmente de ofertas laborales.

```
In [1]: #Lectura de datos de valoraciones
        import pandas as pd
        import matplotlib.pyplot as plt
        df = pd.read_csv("../BD/RatingGeneral_CV_JOBS180519.csv")

In [2]: #Lectura de curriculums vitae
        dfcv = pd.read_csv("../BD/CurriculumVitaes(5k).csv")

In [3]: #Lectura de datos sobre ofertas laborales
        dfjob = pd.read_csv("../BD/PuestosLaborales2.csv")

In [4]: ratings_df = pd.DataFrame(df, columns = ['Id_Job', 'Id_Cv', 'Rating_general'])
```

A continuación, se puede apreciar los datos de promedios o rating de tf-idf obtenidos en la fase de extracción de datos después haber procesado con técnicas procesamiento lenguaje natural y precisamente con la técnica frecuencia de términos y frecuencia inversa de documentos (TF-IDF).

```
In [5]: ratings_df.head()
```

Out[5]:

	Id_Job	Id_Cv	Rating_general
0	0	0	0.092
1	0	1	0.000
2	0	2	0.004
3	0	3	0.002
4	0	4	0.012

En seguida se puede visualizar y apreciar los datos de los currículums vitae, la data original con sus respectivas campos de información.

```
In [6]: dfcv.head()
```

```
Out[6]:    Id_User                            Nombre  \
        0        0  Miguel Angel Garcia del Castillo
        1        1     Jesús Augusto Valqui Barraza
        2        2           Lizbeth Huaman Ventura
        3        3                     Jorge Romero
        4        4            Marjori Santos Sedano

                                              Descripcion  \
        0  Soy un profesional capacitado para llevar adel...
        1  Cursante del IV ciclo de Administración de Neg...
        2  Soy una joven con grandes expectativas, respon...
        3  Abogado con cuatro años de experiencia profesi...
        4  En ciencias de la comunicaciones cursando el 4...

                                              Experiencia  \
        0  publicaciones sobre sistema de recomendacion y...
        1
        2  CONCAR S.A. Grupo Graña y Montero Desde Mayo 2...
        3  PerfilSoy una persona 100% creativa actitud pr...
        4

                                              Habilidades
        0                     python, pln, nltk, scrapy, tec
        1  Asociación Civil Sol y Mar /  Club Sol y Mar  ...
        2  Concar s.a. Grupo Graña y Montero / Informátic...
        3  ESTUDIO JURÍDICO ROMERO / Legal / Asesoría / L...
        4  Delosi / Atención a clientes / Venta al consum...
```

En el siguiente punto se visualiza los datos importados de las ofertas laborales, los datos son originales y con sus respectivos campos de información.

```
In [7]: dfjob.head()
```

```
Out[7]:    Id_Job                                            Titulo  \
        0       0                                        Investigador
        1       1  Vendedores rotativos para tiendas Baby Infanti...
        2       2  Asistente Contable Técnico o Universitario Egr...
        3       3              Practicante de Administración (Apoyo)
        4       4                                 Jefe de Producción

                                  Empresa            Ciudad  \
        0                 Honorio's analytics  San Isidro  Lima
        1                 SGA Consultores SAC  Miraflores  Lima
        2                    Adecco Perú S.A.   San Borja  Lima
        3  Industrias Plasticas  Reunidas S.A.C       Lima  Lima
        4                                 NaN         NaN
```

```
                                          Descripcion  \
0  Se necesita un profesional muy preparado para ...
1  Por encargo de las Tiendas Baby Infanti y Lego...
2  Por encargo de nuestro cliente  empresa líder ...
3  Industrias Plásticas Reunidas es una empresa 1...
4  Importante empresa del sector se encuentra en ...

                                          Requisitos  \
0  Conocimientos en procesamiento lenguaje natura...
1  Secundaria completa. Manejo fluido a nivel usu...
2
3
4

                                           Funciones             Educacion  \
0  Desarrollador encargado del proyecto de sistem...           Universitario
1                                                    Educación Secundaria
2  Análisis de depreciaciones. Asientos contables...                Técnico
3  Atención directa a gerencia en relación a coor...                Técnico
4  Supervisar y garantizar el cumplimiento del Pl...           Universitario

                                       Conocimientos
0  python, scrapy, procesamiento lenguaje natural...
1
2                                  Microsoft Excel
3                                  Microsoft Excel
4                                  Microsoft Excel
```

En seguida se aglomeran los datos de ofertas laborales (dfjob) y rating de relevancia de términos de cada CV (ratings_df) con respecto a todas las ofertas laborales, estos son acoplados a través de sus claves principales, "Id_Job" de ofertas laborales y "Id_Job" de rating de Cvs.

```
In [8]: merged_inner1 = pd.merge(
            left=ratings_df,right=dfjob, left_on='Id_Job', right_on='Id_Job')
        merged_inner1.shape

Out[8]: (49990000, 11)

In [14]: merged_inner1.head()

Out[14]:    Id_Job  Id_Cv  Rating_general        Titulo              Empresa  \
         0       0      0           0.092  Investigador  Honorio's analytics
         1       0      1           0.000  Investigador  Honorio's analytics
         2       0      2           0.004  Investigador  Honorio's analytics
         3       0      3           0.002  Investigador  Honorio's analytics
         4       0      4           0.012  Investigador  Honorio's analytics

                  Ciudad                                          Descripcion  \
         0  San Isidro  Lima  Se necesita un profesional muy preparado para ...
```

```
1  San Isidro  Lima  Se necesita un profesional muy preparado para ...
2  San Isidro  Lima  Se necesita un profesional muy preparado para ...
3  San Isidro  Lima  Se necesita un profesional muy preparado para ...
4  San Isidro  Lima  Se necesita un profesional muy preparado para ...

                                          Requisitos  \
0  Conocimientos en procesamiento lenguaje natura...
1  Conocimientos en procesamiento lenguaje natura...
2  Conocimientos en procesamiento lenguaje natura...
3  Conocimientos en procesamiento lenguaje natura...
4  Conocimientos en procesamiento lenguaje natura...

                                            Funciones      Educacion  \
0  Desarrollador encargado del proyecto de sistem...  Universitario
1  Desarrollador encargado del proyecto de sistem...  Universitario
2  Desarrollador encargado del proyecto de sistem...  Universitario
3  Desarrollador encargado del proyecto de sistem...  Universitario
4  Desarrollador encargado del proyecto de sistem...  Universitario

                                       Conocimientos
0  python, scrapy, procesamiento lenguaje natural...
1  python, scrapy, procesamiento lenguaje natural...
2  python, scrapy, procesamiento lenguaje natural...
3  python, scrapy, procesamiento lenguaje natural...
4  python, scrapy, procesamiento lenguaje natural...
```