IEBS BUSINESS SCHOOL

TRABAJO DE FINAL DE MÁSTER

ALUMNOS

- DIEGO CUESTA JAIRO
- FARAVELLI RODRÍGUEZ MARC
- GONZÁLEZ GUSTAVO MARTÍN
- OLIVERA MATURANA FRANCISCO

PROGRAMA:

MASTER EN BUSINESS INTELLIGENCE

AND DATA SCIENCE

NOMBRE DEL PROYECTO

PREDICCIÓN DE PERSONALIDAD EN BASE AL INDICADOR MBTI UTILIZANDO HERRAMIENTAS DE INTELIGENCIA ARTIFICIAL

AÑO 2021

Contenido

1 RESUMEN

El presente trabajo tiene como finalidad, crear un algoritmo que, en base a comentarios de una persona, objeto de análisis, prediga cuál es su personalidad, utilizando el indicador MBTI, con miras a aplicarlo en el ámbito empresarial, específicamente a procesos relacionados con selección y formación de recursos humanos.

El Myers-Briggs Type Indicator (o MBTI por sus siglas en inglés) es un test de personalidad diseñado para ayudar a una persona a identificar algunas de sus preferencias personales más importantes.

El indicador es utilizado frecuentemente en campos tales como la pedagogía, dinámica de grupos, capacitación de personal, desarrollo de capacidades de liderazgo, asesoramiento matrimonial y desarrollo personal.

Este indicador se diferencia de otros tests estandarizados y otros parámetros de medida, tales como la inteligencia, en que no mide una característica, sino que clasifica los tipos de preferencias de las personas. De acuerdo a la teoría del MBTI®, mientras que los tipos y parámetros son innatos, las preferencias pueden ser mejoradas (de aquí su importancia y relevancia para aplicarse junto con el coaching), y si el individuo se encuentra en un medio ambiente sano, naturalmente se diferenciarán con el transcurso del tiempo.

Generalmente el test se presenta como un set de preguntas que la persona debe contestar. En función a los resultados, se encuadra a la persona en 16 tipos diferentes, cada uno con sus características propias.

Si bien es uno de los métodos más difundidos y utilizados, presenta algunas desventajas: No tiene en cuenta el contexto, el estado de ánimo de la persona en ese momento, su actitud positiva o negativa frente al test, lo cual puede ocasionar distorsiones en los resultados.

Los resultados pueden verse sesgados dado que la persona infiere que, dependiendo la respuesta que dé, el test puede arrojar conclusiones que pueden afectarla en determinada situación, por ejemplo, al momento de evaluar su aptitud para ser contratada en un nuevo trabajo.

Dados los problemas y limitaciones planteados, se tiene como objetivo desarrollar un algoritmo de inteligencia artificial, que, en base a comentarios y apreciaciones del sujeto, pueda predecir su tipo de personalidad, apoyando procesos decisorios empresariales, de contratación de personal, pero no limitando su uso a este ámbito.

Consideramos que esta solución minimiza los problemas de sesgo del test, agregando agilidad a los procesos desarrollados en los departamentos de recursos humanos, relacionados al reclutamiento y dinámica de grupos promoviendo el desarrollo y mejora continua de los diferentes colaboradores de la empresa en cuestión, agregando un aspecto profesional, innovador y tecnológico en la gestión de talentos.

2 INTRODUCCIÓN

2.1 RRHH Y EL PROCESO DE CONTRATACIÓN

Actualmente, existe una corriente que impulsa el desarrollo del talento, transformando el lugar de trabajo, la fuerza laboral y el trabajo como tal.

Deloitte, en su artículo llamado "Tendencias Globales en Capital Humano 2016 - La nueva organización: Un diseño diferente" (Deloitte University Press, 2016) muestra la necesidad por parte de los directivos de rediseñar las organizaciones, construyendo las mismas con base en equipos altamente empoderados, impulsados por un nuevo modelo de administración y dirigidos por una generación de líderes más jóvenes, globales y diversos.

Para lograr este cambio, los CEO y líderes de RRHH se están enfocando en comprender y crear una cultura compartida, así como en diseñar un ambiente de trabajo cautivador y en construir un nuevo modelo de liderazgo y desarrollo profesional.

En cuanto a la competencia de cara a buscar talento calificado, las organizaciones se disputan a los mejores candidatos en un mercado laboral altamente transparente, adoptando tecnologías digitales para reinventar el lugar de trabajo, enfocándose en la diversidad e inclusión como estrategia de negocio.

Las organizaciones están luchando para ser más digitales y ágiles con el fin de satisfacer las necesidades del cliente, sin embargo, también están cambiando su estructura buscando lograr equipos flexibles e interconectados donde las organizaciones construyen y empoderan equipos para que trabajen en proyectos específicos del negocio.

Esta nueva estructura requiere que se desarrollen actividades y programas tales como desarrollo de liderazgo, gestión del desempeño, aprendizaje y crecimiento profesional.

Dentro del ámbito de los Recursos Humanos, se utilizan diferentes técnicas y métodos para lograr **desarrollar a los colaboradores** de una organización y **contratar** a aquellos que reúnen las características necesarias que requiere cada puesto, con sus competencias y habilidades. Entre las mencionadas técnicas, podemos destacar al **test MBTI** (Myers Briggs Type Indicator, por sus siglas en inglés), el cual es un instrumento para identificar características, rasgos de personalidad y preferencias personales.

La naturaleza de la teoría de este indicador es que las muchas variaciones en el comportamiento que, al parecer son aleatorias, son realmente coherentes con las diversas formas de los individuos de usar su percepción y juicio.

En la evaluación MBTI se definen 16 tipos de personalidad, materializándose en un cuestionario introspectivo que ayuda a indicar diferentes preferencias psicológicas tanto por percepción como en la toma de decisiones.

Dentro de cada tipo de personalidad, se toman en cuenta varios factores como puntos débiles y fuertes, características, potenciales y preferencias. De esta manera, es posible entender mejor las acciones de cada uno en diferentes situaciones y momentos.

Para la empresa, la evaluación MBTI es de gran ventaja pues es un método fácil para entender los comportamientos psicológicos, particularidades de cada personalidad, preferencias y estilo de trabajo de sus empleados. Por lo tanto, la empresa puede trabajar en formas más apropiadas para desarrollar el potencial de cada uno.

Además, MBTI puede ayudar a identificar las mejores formas de desarrollar relaciones y destrezas, tanto para el desarrollo personal como el profesional.

Los recursos humanos son indiscutiblemente el activo más valioso de una empresa: son responsables de administrar todo el personal de la organización para obtener el mayor valor posible de sus empleados.

Han estado utilizando la analítica durante años. Sin embargo, la recopilación, el procesamiento y el análisis de datos fue en gran parte manual y, dada la naturaleza, la dinámica de sus KPIs y el enfoque que tenían, se quedaban limitados en esta labor. Por lo tanto, es sorprendente que los departamentos de recursos humanos se dieran cuenta de la utilidad del machine learning tan tarde.

No obstante, el machine learning ha entrado lentamente, pero convencidamente en el dominio de los recursos humanos y se han creado varios casos de uso, como la previsión de deserción, la toma de derechos y la predicción del éxito de un candidato potencial. Pronto también se descubrirán más casos de uso.

A diferencia del enfoque manual, el enfoque de machine learning es mucho más rápido, mucho más sensible a situaciones dinámicas y proporciona datos precisos, factibles y valiosos. Con la aparición del Big Data de fuentes web como foros y redes sociales, las organizaciones están encontrando a los candidatos correctos para los puestos adecuados. Al evaluar su aplicación, el machine learning considera sus calificaciones, experiencia, intereses, conexiones y registros profesionales, logros, discusiones en foros y más. Esto mejora significativamente las posibilidades de adaptar roles, aunque no se garantice. Un buen ejemplo sería el sitio de redes profesionales LinkedIn.

El machine learning reduce significativamente el esfuerzo manual en la gestión de aplicaciones y libera a los recursos humanos para así centrarse en esfuerzos más

productivos. Según Cristian Rennella, CEO y Cofundador de MejorTrato.com.mx, una empresa que compara productos financieros, "En el pasado, invertimos el 67.2 por ciento del tiempo de cada persona en Recursos Humanos leyendo los CV de cada candidato que se acercaba a nosotros a través de nuestro propio sitio web y de terceros. Gracias a la IA, este trabajo ahora lo realiza automáticamente nuestro sistema interno, que a través del aprendizaje profundo con TensorFlow, podemos automatizar esta tarea ".

2.2 TEST MBTI

La prueba MBTI, muy extendida especialmente en el mundo anglosajón y muy popular en Internet, es una herramienta poderosa para conocer mejor las características de la personalidad. Basado en una teoría psicológica simple, pero al mismo tiempo ingeniosa, revela de una manera formidable cómo cada uno de nosotros se relaciona con los demás y con el mundo exterior.

Myers y Briggs, las idealistas de esta teoría, la describen de la siguiente manera:

“La esencia de la teoría es que una diferencia de comportamiento aparentemente aleatoria es, en cambio, ordenada y coherente, debido a ciertas diferencias básicas en la forma en que los individuos prefieren usar su percepción y su juicio”.

La pista para la creación de este indicador llegó durante la Segunda Guerra Mundial, para ayudar a las mujeres a comprender qué trabajo era más adecuado para ellas, en un momento en que muchos hombres estaban ocupados luchando.

El hecho de realizar el test y leer el perfil resultante es realmente interesante para entender no sólo la personalidad, sino también qué trabajos son los más adecuados para cada uno. Tanto es así que muchas empresas, en los Estados Unidos y en otros países, piden a los candidatos que declaren su perfil MBTI, para conocerlos mejor y para comprender dónde pueden rendir más (Winterhalter, 2014).

Para comprender este Test debemos remontarnos a la Teoría del suizo Carl G. Jung quien en 1921 publicó “*Psychological Types*”, introduciendo la idea de que cada persona tiene un tipo psicológico. Quien menciona “*lo que parece ser un comportamiento aleatorio es en realidad el resultado de diferencias en la forma en que las personas prefieren usar sus capacidades mentales*”. Observó que las personas generalmente se involucran en una de dos funciones mentales: recibir información, que él llamó percibir, u organizar información y llegar a conclusiones, lo que llamó juzgar.

Además, observó que las personas prefieren realizar esa función de dos maneras, lo que llamó preferencias. Añadió que, aunque todos toman información y toman una decisión, algunas prefieren tomar más información (percibir) y otras prefieren tomar más decisión (juzgar). Luego mencionó que el *tipo psicológico* de una persona consiste en su preferencia en cada categoría.

"*Cada persona parece estar más energizada por el mundo externo (extraversión) o el mundo interno (introversión)"*

Sin embargo, sus estudios fueron muy académicos, dificulta la lectura para el común de las personas y se limitaba solo a algunos entendidos.

Con la finalidad de entender este importante hallazgo de Carl Jung, dos mujeres durante la segunda guerra mundial Isabel Briggs Myers y su madre lograron realizar el Test MBTI para utilizar la teoría de Carl Jung sin la necesidad de leer su teoría, lo cual es casi tan meritorio que la propia teoría, ya que da cuenta el nivel de entendimiento alto para poder traducir toda la teoría "Psychological Types" a unas cuantas preguntas y lograr obtener los resultados de diferentes personalidades (The Myers & Briggs Fundation, s.f.).

El test consiste en la identificación y descripción de los 16 tipos de personalidad distintivos que resultan de las interacciones entre las preferencias. Una preferencia es lo que te gusta o no, no existen preguntas buenas o malas. Si bien todas las preferencias son iguales, cada una tiene diferentes fortalezas y diferentes desafíos. Conocer estas fortalezas y desafíos de la personalidad puede ayudar a la persona, a comprender y apreciar cómo todos contribuyen a una situación, una tarea o la solución de un problema.

Según MBTI ® Manual: A Guide to the Development and Use of the Myers-Briggs Type Indicator (Briggs Myers & McCaulley, 1962), las preferencias se pueden clasificar en:

> **Mundo favorito:** ¿Prefieres centrarte en el mundo exterior o en tu propio mundo interior? Esto se llama Extraversión o Introversión. **I-E:** Clasificación binaria, Introversión (I) – Extroversión (E)
>
> **Información:** ¿Prefieres centrarte en la información básica que recibes o prefieres interpretar y añadir significado? Esto se llama detección (S) o intuición (N). **N-S:** Clasificación binaria, Intuition (N) – Sensing (S)
>
> **Decisiones:** al tomar decisiones, ¿prefiere mirar primero a la lógica y la coherencia o primero a las personas y las circunstancias especiales? Esto se llama Pensar (T) o Sentir (F). **T-F:** Clasificación binaria, Thinking (T) – Feeling (F)

Estructura: al tratar con el mundo exterior, ¿prefiere que las cosas se decidan o prefiere permanecer abierto a nueva información y opciones? Esto se llama Juzgar (J) o Percibir (P). **J-P:** Clasificación binaria, Judging (J) – Perceiving (P)

Su tipo de personalidad: cuando decide su preferencia en cada categoría, tiene su propio tipo de personalidad , que puede expresarse como un código de cuatro letras.

Con esta información se clasifican 16 tipos de personalidad, las cuales se describen a continuación (The Myers & Briggs Fundation, s.f.)

ISTJ: Tranquilo, serio, gana el éxito con minuciosidad y confiabilidad. Práctico, realista y responsable. Decida lógicamente qué se debe hacer y trabaje para lograrlo de manera constante, independientemente de las distracciones. Disfrute de hacer todo ordenado y organizado: su trabajo, su hogar, su vida. Valora las tradiciones y la lealtad.

ISFJ: Tranquilo, amable, responsable y concienzudo. Comprometidos y constantes en el cumplimiento de sus obligaciones. Completo, minucioso y preciso. Leales, considerados, noten y recuerdan los detalles de las personas que son importantes para ellos, preocupados por cómo se sienten los demás. Esfuércese por crear un entorno ordenado y armonioso en el trabajo y en el hogar.

INFJ: Busque significado y conexión en ideas, relaciones y posesiones materiales. Quieren comprender qué motiva a las personas y son perspicaces sobre los demás. Concienzudos y comprometidos con sus firmes valores. Desarrolla una visión clara sobre la mejor manera de servir al bien común. Organizado y resolutivo en la implementación de su visión.

INTJ: Tienen mentes originales y un gran impulso para implementar sus ideas y lograr sus objetivos. Ven rápidamente patrones en eventos externos y desarrolla perspectivas explicativas de largo alcance. Cuando está comprometido, organiza un trabajo y lo lleva a cabo. Escépticos e independientes, tienen altos estándares de competencia y desempeño, para ellos mismos y para los demás.

ISTP: Observadores tolerantes, flexibles y silenciosos hasta que aparezca un problema, luego actúan rápidamente para encontrar soluciones viables. Analiza qué hace que las cosas funcionen y obtienen fácilmente grandes cantidades de datos para aislar el núcleo de los problemas prácticos. Interesado en causa y efecto, organizan hechos usando principios lógicos, valora la eficiencia.

ISFP: Tranquilo, amigable, sensible y amable. Disfruta el momento presente, lo que sucede a su alrededor. Les gusta tener su propio espacio y trabajar dentro de su propio marco de tiempo. Leales y comprometidos con sus valores y con las personas que son importantes para ellos. No le gustan los desacuerdos y los conflictos, no impone sus opiniones o valores a los demás.

INFP: Idealista, fiel a sus valores y a las personas que le son importantes. Quieren una vida externa que sea congruente con sus valores. Las posibilidades curiosas, rápidas de ver, pueden ser catalizadores para implementar ideas. Busque comprender a las personas y ayudarlas a desarrollar su potencial. Adaptable, flexible y tolerante a menos que un valor se vea amenazado.

INTP: Buscar desarrollar explicaciones lógicas para todo lo que les interese. Teórico y abstracto, interesado más en las ideas que en la interacción social. Silencioso, contenido, flexible y adaptable. Tiene una capacidad inusual para concentrarse en profundidad para resolver problemas en su área de interés. Escéptico, a veces crítico, siempre analítico.

ESTP: Flexibles y tolerantes, adoptan un enfoque pragmático centrado en resultados inmediatos. Las teorías y las explicaciones conceptuales los aburren: quieren actuar con energía para resolver el problema. Concéntrese en el aquí y ahora, espontáneo, disfrute cada momento en el que pueda estar activo con los demás. Disfrute de las comodidades y el estilo de los materiales. Aprende mejor haciendo.

ESFP: Extrovertido, amigable y tolerante. Amantes exuberantes de la vida, las personas y las comodidades materiales. Disfruta trabajando con otros para hacer que las cosas sucedan. Aporta sentido común y un enfoque realista a su trabajo y hacen que el trabajo sea divertido. Flexible y espontáneo, se adapta fácilmente a nuevas personas y entornos. Aprenden mejor probando una nueva habilidad con otras personas.

ENFP: Calurosamente entusiasta e imaginativo. Ven la vida llena de posibilidades. Establecen conexiones entre eventos e información muy rápidamente y proceden con confianza según los patrones que vean. Desea mucha afirmación de los demás y presten apoyo y aprecio de buena gana. Espontáneos y flexibles, suelen depender de su capacidad para improvisar y su fluidez verbal.

ENTP: Rápido, ingenioso, estimulante, alerta y franco. Ingenioso para resolver problemas nuevos y desafiantes. Experto en generar posibilidades conceptuales y

luego analizarlas estratégicamente. Bueno para leer a otras personas. Aburrido por la rutina, rara vez hará lo mismo de la misma manera, apto para volverse hacia un nuevo interés tras otro.

ESTJ: Práctico, realista. Decisivo, se mueve rápidamente para implementar decisiones. Organiza proyectos y personas para hacer las cosas, se concentra en obtener resultados de la manera más eficiente posible. Cuida los detalles de la rutina. Tiene un conjunto claro de estándares lógicos, los sigue sistemáticamente y desea que otros también lo hagan. Enérgico en la implementación de sus planes.

ESFJ: Afectuoso, concienzudo y cooperativo. Quiere armonía en su entorno, trabaja con determinación para establecerlo. Le gusta trabajar con otros para completar las tareas con precisión y a tiempo. Leal, cumple incluso en los pequeños asuntos. Observa lo que los demás necesitan en su vida diaria y trata de proporcionarles. Quieren ser apreciados por quienes son y por lo que aportan.

ENFJ: Cálido, empático, receptivo y responsable. Muy en sintonía con las emociones, necesidades y motivaciones de los demás. Encuentra potencial en todos, quiere ayudar a otros a desarrollar su potencial. Puede actuar como catalizador para el crecimiento individual y grupal. Leal, receptivo a los elogios y críticas. Sociable, facilita a los demás en un grupo y proporciona un liderazgo inspirador.

ENTJ: Franco, decidido, asume el liderazgo de buena gana. Vea rápidamente procedimientos y políticas ilógicas e ineficientes, desarrolle e implemente sistemas integrales para resolver problemas organizacionales. Disfrute de la planificación a largo plazo y el establecimiento de objetivos. Por lo general, están bien informados, leen bien, disfrutan ampliando sus conocimientos y transmitiendo a los demás. Enérgico en la presentación de sus ideas.

Estos códigos sobre la clasificación de la personalidad corresponden y estarán en concordancia a los resultados de clasificación del presente trabajo.

2.3 DEBILIDADES DETECTADAS

En esta sección del presente trabajo final se busca exponer las debilidades y problemas detectados en dos frentes:

Por un lado, aquellos puntos débiles que denota específicamente el test MBTI, por otro, debilidades y problemas que se observan en procesos propios del área de recursos

humanos, tales como la selección del talento y su posterior formación, estando convencidos de que la inteligencia artificial puede contribuir a disminuir los mencionados puntos débiles y errores en los frentes mencionados.

Enfocándonos en el **test MBTI**, específicamente, en la medida de que se respondan a las preguntas con la mayor sinceridad, se podrá obtener inmediatamente un resultado fiel a la realidad. La prueba tiene su principal debilidad al basarse en dicotomías: puede ser blanco o negro, pero no gris. Esto significa que es bastante común no obtener el resultado correcto al principio. Y, como explicaron los creadores, una pequeña variación cambia significativamente el perfil psicológico de un individuo.

No siempre es fácil determinar infaliblemente la posición de uno en una de las 4 dicotomías. Reconocemos de inmediato al extrovertido que se pasaría la vida hablando con 20 personas a la vez, así como al meticuloso organizador de su propia vida y la de los demás. Pero ... ¿y todos los demás?

Como se trata de una prueba, entra en juego el factor de **deseabilidad social**: muchas personas tienden a responder de acuerdo con lo que creen que es más aceptado socialmente. Es decir, pensando en cómo deberían ser, y no en cómo son. Resultado: obtiene ENFP cuando tal vez sea ESTP. Por eso es bueno repetir la prueba más de una vez y verificar empíricamente el resultado, siempre que se tenga un buen conocimiento de las bases teóricas del MBTI.

Por otro lado, no solo es difícil predecir una relación ideal entre diferentes tipos, sino que también hay que tener en cuenta que, con el mismo tipo psicológico, hay muchos factores "extra-MBTI" que pueden hacer que las personas que han obtenido un resultado idéntico en la prueba sean diferentes:

- Edad
- Género de pertenencia
- Cultura de origen
- Trasfondo social
- Nivel educacional
- Valores y creencias
- Temperamento (por ejemplo, tranquilo o inquieto)

Respecto a los **procesos propios del área de Recursos Humanos**, detectamos como problema, la falta de definición del perfil para cubrir el puesto que se está buscando. Limitarse a filtrar a los candidatos por el nivel de formación y la experiencia **sin tener en**

cuenta otras características como las aptitudes, el perfil psicológico, las habilidades sociales, las destrezas en la comunicación y la forma en la que el candidato encaja en la empresa, acorde a los valores organizacionales, puede tener consecuencias muy negativas, desde la **contratación de personal sobre cualificado hasta dar con alguien con serias dificultades para alcanzar el nivel exigido**. La urgencia por encontrar un buen profesional desembocará en una elección apresurada e improvisada que puede traer desventajas y pérdidas. Se tenderá a acudir a técnicas de reclutamiento muy manidas y poco creativas, sin darle oportunidad a nuevas técnicas que valoran mejor el potencial que tiene un candidato.

Otro de los errores comunes es no rastrear **talento interno** en la empresa ya que se podría estar desaprovechando el talento que pueda haber dentro de la organización.

Sin perjuicio de lo expuesto, en las organizaciones es común observar falta de motivación y desarrollo de sus equipos de trabajo. Esto afecta directamente a la productividad de la empresa y los resultados obtenidos ya que la persona trabaja por "cumplir" en vez de trabajar de manera conjunta para conseguir el mejor producto, lo que al final del día hace la diferencia de los servicios prestados por la empresa respecto de otros proveedores. Por este motivo, actualmente se suma a las funciones de áreas de Recursos Humanos, las de capacitación y formación del talento interno desarrollando competencias relacionadas a la gestión del tiempo, trabajo en equipo y liderazgo, entre otras.

Los problemas y errores explicados en el párrafo anterior pueden ser solucionados con un verdadero plan y desarrollo del área de recursos humanos que abarque los siguientes puntos:

- Definir un plan de reclutamiento
- Crear políticas de búsquedas internas
- Evaluar las aptitudes y competencias del personal de la empresa y de nuevos candidatos.
- Definir planes y políticas de desarrollo de habilidades y competencias

2.4 MOTIVACIONES

En este trabajo nos proponemos investigar, analizar y desarrollar un algoritmo de inteligencia artificial que permita predecir la personalidad de una persona, utilizando el indicador MBTI, a partir de **comentarios** realizados por el sujeto, obtenidos a partir de

diferentes fuentes, tales como comentarios de redes sociales, mails enviados, o comentarios que haya realizado el sujeto luego de observar un video o imagen.

Este algoritmo permitirá predecir el tipo de personalidad de un sujeto, disminuyendo posibles sesgos que pueden existir cuando una persona está completando el test correspondiente, y así, llegar a diferentes conclusiones de cara a eficientizar un proceso de reclutamiento o realizar dinámicas de grupo que permitan empoderar y dotar de herramientas a los trabajadores de cara a lograr objetivos organizacionales, reduciendo, de esta manera, tiempo y costes.

Para lograr el objetivo que nos planteamos, se ha tomado un set de datos que contiene una base con los posts de más de 8 mil usuarios y su resultado del indicador Myers-Briggs.

Con foco en dicho dataset, se aplicarán diferentes algoritmos de machine learning, comparando los resultados o seleccionando el algoritmo con mayor precisión.

2.5 ESTRUCTURA DEL DOCUMENTO

El presente trabajo final se estructurará en capítulos, en donde se buscará abordar de manera clara y ordenada los diferentes conceptos y teorías, necesarias para dar una respuesta al problema planteado.

Comenzaremos con el “estado del arte” donde se realizará una breve descripción de trabajos y ensayos donde sus autores e investigadores abordaron la construcción de modelos predictivos basados en el test MBTI. A su vez, se dejará de manifiesto la innovación o punto diferenciador que se plantea en este trabajo.

El lector continuará con el apartado de “objetivos” donde se exponen los objetivos generales y específicos que se buscarán cumplir con el desarrollo de este trabajo final de máster.

Seguidamente se desarrollará la solución con la que buscaremos predecir el resultado del test MBTI de una persona, a partir de comentarios que realice la misma.

Se comenzará explicando las herramientas utilizadas para abordar la solución.

Como siguiente punto, se procederá a realizar un primer análisis exploratorio de datos, instrumentando un fuerte trabajo de pre procesamiento de datos, para luego, estar en condiciones de construir los modelos de ML que luego serán evaluados de forma de elegir aquel algoritmo que arroje mejores resultados predictivos.

Culminando este trabajo se expondrán los trabajos futuros que, a nuestro entender, son necesarios si se quiere mejorar la precisión de los algoritmos utilizados.

Por último, se presentarán las conclusiones generales, productos de todo el desarrollo e investigación realizada, así como también apreciaciones personales de cada uno de los autores de este trabajo.

3 ESTADO DEL ARTE

3.1 APROXIMACIONES ACTUALES

Hoy en día las técnicas de machine learning se utilizan en casi todos los sectores, desde los modernos smartphones hasta los diagnósticos médicos, aprovechando su reducido tiempo computacional, gran capacidad de generalización y flexibilidad operativa. En el campo de la contratación de personal en los departamentos de recursos humanos, sin embargo, este tipo de innovación aún no se ha expandido mucho, aunque en los últimos años se están comenzando a difundir posibles aplicaciones destinadas a simplificar tanto la recopilación y tratamiento de los datos como el estudio de la personalidad de los candidatos. Este cambio se ha implementado a través de un uso cada vez mayor de la inteligencia artificial que ha transformado la selección y contratación de personal en una práctica que se llevará a cabo a través de Internet, o en línea.

Muchas de las técnicas de machine learning se basan en el deep learning, y en particular en las redes neuronales artificiales, siendo la herramienta ideal para el análisis impulsado por datos con grandes cantidades de elementos, logrando reconocer características implícitas y tendencias intrínsecas en los datos y por tanto clasificarlos en diferentes categorías.

John McCarthy, quien acuñó el término "inteligencia artificial" en 1956, lo define como "la ciencia y la ingeniería de la fabricación de máquinas inteligentes".

Es un sistema habilitado por computadora, como un sistema robótico diseñado para procesar información de manera similar a lo que hace la fuerza laboral en la organización, utilizando su capacidad para aprender, tomar decisiones y resolver problemas.

Jonathan Kestenbaum (Kestenbaum, 2016) dice, en cambio, que la implementación del software de IA simplemente elimina las tareas mundanas y el análisis de datos que requieren mucho tiempo, para servir como un solucionador continuo de problemas para los recursos humanos.

La inteligencia artificial, sin embargo, requiere un seguimiento constante y una mejora continua de los procesos de negocio, que pueden ser soportados por nuevas tecnologías, pero que ante todo requieren del capital humano adecuado para llevar a cabo las actualizaciones continuas que requiere el software.

Por tanto, se puede destacar que el proceso es de naturaleza cíclica, en particular para implementar y actualizar las herramientas soportadas por la inteligencia artificial es

necesario contar con un capital humano adecuado y competente, que, sin embargo, será seleccionado precisamente por las herramientas que implemente.

Todo el proceso de adquisición de talento vuelve a ser un elemento clave para la empresa, por lo que debe implementarse con herramientas especialmente sofisticadas e innovadoras.

De hecho, si este proceso se lleva a cabo con el máximo cuidado y con los medios adecuados, se podrían lograr muchos beneficios, que incluyen (Bhanu et al., 2016):

- Ahorro de tiempo: la inteligencia artificial ahorra tiempo ya que, al tener una memoria interna, lleva a no repetir la misma tarea.
- Mapeo de talentos: la IA ayuda a los recursos humanos a adquirir el mejor talento requerido para la organización, ya que es capaz de analizar rápidamente las habilidades de los empleados y sus conocimientos.
- Reducción de costes: la tarea de adquirir el personal adecuado para la organización se realiza de forma cualitativa e internamente dentro de la empresa, por lo que se reducen las solicitudes a la agencia de contratación subcontratada.
- Contratación con equidad: la herramienta de inteligencia artificial funciona para utilizar grandes datos para la contratación y realiza una selección y una evaluación imparciales. Esto conduce a la contratación de candidatos de calidad, sin tener en cuenta las características externas del individuo que podrían dar lugar a un juicio subjetivo.
- Corrección de consultas: Los empleados reciben información actualizada y obtienen respuestas inmediatas a sus preguntas. En definitiva, la inteligencia artificial conduce a la satisfacción de los empleados porque se sienten considerados y parte de la empresa y, por tanto, un mayor compromiso de su parte.
- Aspirantes de calidad: las herramientas tecnológicas ayudan a seleccionar aspirantes a empleados de calidad. De hecho, la inteligencia artificial ayuda a identificar las habilidades, competencias y características de los candidatos que corresponden al puesto solicitado.

Mohammad Hossein Amirhosseini y Hassan Kazemian escribieron un artículo llamado Machine Learning Approach to Personality Type Prediction Based on the Myers–Briggs Type Indicator (Mohammad & Kazemian, 2020) donde buscan aplicar algoritmos de inteligencia artificial de forma de predecir el tipo de personalidad de un sujeto basándose en el test MBTI.

Los autores explican que se han aplicado diferentes técnicas para predecir la personalidad, cada una con un nivel diferente de precisión, sin embargo, sostienen que no se ha utilizado Gradient Boosting en el campo.

El gradient boosting es una técnica de aprendizaje automático para problemas de regresión y clasificación, que produce un modelo de predicción en forma de un conjunto de modelos de predicción débiles, generalmente árboles de decisión. Construye el modelo por etapas como lo hacen otros métodos de impulso, y los generaliza al permitir la optimización de una función de pérdida diferenciable arbitraria.

XGBoost es una de las implementaciones del concepto Gradient Boosting, pero lo que hace que sea único es que utiliza "una formalización de modelo más regularizada para controlar el sobreajuste, lo que le da un mejor rendimiento", según el autor del algoritmo, Tianqi Chen. Por tanto, ayuda a reducir el sobreajuste[1]

Los citados autores concluyen en que los resultados obtenidos con el modelo construido en base a la técnica XGBoost presentan mayor precisión que los algoritmos usualmente utilizados en esta rama.

Por su parte, Anthony Ma y Gus Liu (Liu & Ma) investigaron la manera de desarrollar algoritmos utilizando redes neuronales para predecir la personalidad de autores de libros a partir del análisis de los mismos.

Los autores exploraron una variedad de métodos para abordar el problema de predicción de la personalidad. Comenzaron construyendo manualmente un gran corpus de extractos de novelas famosas con la correspondiente clasificación de MBTI de cada autor.

Finalmente, construyen una red neuronal recurrente que tiene como objetivo lograr sistemas más generalizables que no solo tengan en cuenta el contenido de una oración, sino también su estructura y progresión. Para poder cumplir el objetivo planteado, desarrollan redes neuronales recurrentes del tipo Long Short Term memory (LSTM), obteniendo una precisión del 37%.

Los autores concluyen que el modelo construido hace bien en predecir las tendencias generales de personalidad, pero hay que trabajar más en el desarrollo de redes neuronales más sofisticadas, métricas de evaluación completas y conjuntos de datos expansivos.

En conclusión, la tecnología de IA tiene un gran impacto en la actividad de reclutamiento, ya que permite al reclutador alinear todos los datos biológicos de los candidatos no

[1] (https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7)

estructurados, construir el perfil de manera uniforme, identificar y combinar las habilidades requeridas para la industria (Geetha et al., 2016), ya que las tecnologías son capaces no solo de aumentar la eficiencia del proceso de selección, sino también de hacerlo automáticamente y más eficazmente que los métodos de selección tradicionales.
Para ello, las empresas están desarrollando sistemas cada vez más avanzados e innovadores de adquisición de personal, con el fin de agilizar y facilitar el proceso de contratación, tanto desde el punto de vista del empresario como del trabajador.
La herramienta de contratación y gestión de talento en la que nos queremos focalizar será analizada en los siguientes párrafos, teniendo en cuenta que no todas las empresas utilizan metodologías de última generación, sino que están evolucionando y adaptándose al contexto de referencia.

3.2 INNOVACIÓN PLANTEADA

En párrafos anteriores se explicaba que, para las empresas, la prioridad de hoy es que los empleados sepan adaptarse a las características del contexto y esto implica que se debe realizar un gran trabajo de captación de talento y formación donde las evaluaciones de las características de la personalidad del individuo tendrán un rol central.
Nos proponemos presentar un **mecanismo o metodología** que pueda mejorar el procedimiento convencional de evaluación del tipo de personalidad (por ejemplo, encuesta por cuestionario) hacia la determinación automática de los rasgos de la misma. Específicamente buscamos construir un algoritmo con niveles de precisión aceptables que, a partir de comentarios, pueda predecir la personalidad del sujeto evaluado.
Los comentarios o apreciaciones pueden obtenerse por diferentes medios, para luego ser ingestados por el algoritmo.
Según los autores del paper “The Present Situation and the Prospect of Determining the Personality Type of Text Author with Machine Learning” (Cerkez, Vrdoljak, & Skansi), dado que la mayor parte de la población humana pasa una cantidad considerable de tiempo interactuando en las **redes sociales**, esto genera una cantidad masiva de datos durante su comunicación. Asimismo, las redes sociales son una parte esencial de la comunicación en Internet, y como plataforma, brindan una tremenda oportunidad en la evaluación del participante con respecto a su personalidad. Las redes sociales tienen un volumen significativo de fuentes de datos en diferentes formatos, algunos de ellos generados en tiempo real.

Tomando como ejemplo el estudio hecho por Brandom Cui y Calvin Qi titulado “Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction” (Cui & Qi) nos interesa saber, en un mundo donde la comunicación se basa cada vez más en las redes sociales, si existe una fuerte relación entre el **uso del lenguaje en línea y su personalidad real**. Hay dos implicaciones principales de este estudio. En primer lugar, está la posibilidad de que la "persona en línea" sea distinta de la que es en realidad, lo que sugiere que tienen probabilidades de comportarse de una manera completamente diferente en línea. En segundo lugar, los mensajes en las redes sociales, tienen un método de comunicación peculiar y estilos de lenguaje distintos en el escrito respecto al hablado, contienen una cierta cantidad de poder de representación y reflejan la personalidad del autor. Sin embargo, las redes sociales tienen solo unos pocos conjuntos de datos públicos disponibles debido a problemas de privacidad y costos de etiquetado, que también serán un desafío en el futuro.

También se pueden esperar nuevos modelos, no solo para tener mejores resultados de predicción, sino también para mitigar los sesgos en los datos.

Sin embargo, los resultados propuestos son alentadores y dado que las redes sociales son parte de la vida cotidiana de la mayoría de la población, se puede esperar que sean la plataforma para avances en el campo de la predicción automatizada de la personalidad.

Otra de las herramientas para tener en cuenta y que nos gustaría introducir en las empresas gracias a la difusión de la inteligencia artificial es la gamificación: se refiere a la incorporación de **elementos de juego**, en actividades ajenas a ello y en cualquier contexto, como una entrevista de trabajo, dando lugar a evaluaciones basadas en el juego, que se puede clasificar según el nivel de características que tiene (Hawkes, Cek, & Handler, 2017)

Basándonos en el concepto de juego y queriendo mostrar cual sería la idea que tenemos pensada para tal propósito, un ejemplo sería desarrollar una plataforma social a través de la cual cada candidato se enfrenta a pruebas que ponen de manifiesto sus fortalezas y debilidades, pero sobre todo siendo conscientes de estos. La tarea de evaluación se encomendaría a algoritmos que se integrarían con las más válidas pruebas de psico-aptitudes.

Las habilidades que se prueban a través de esta aplicación de gamificación, dependiendo de cada puesto de trabajo, pueden incluir autoconciencia e imagen de sí mismo, negociación y orientación al cliente, habilidades analíticas, reconocimiento de patrones,

precisión y orientación a resultados hasta el trabajo en equipo y la capacidad de crear y gestionar redes con otras personas.

Al comienzo de todo el proceso, se plantean al usuario preguntas de conocimiento general que investigan los conocimientos de historia, geografía, matemáticas y el idioma interesado. Al final, se asigna una puntuación que se sumará a las puntuaciones de las fases posteriores.

El siguiente camino se divide en cuatro fases donde el usuario también puede optar por realizarlas sin ningún orden en particular:

• ¿Quién eres tú?

En esta fase, se pide al usuario que se presente de tres formas: una primera descripción de sí mismo en 140 caracteres (longitud máxima de una publicación de Twitter), la creación de su propio avatar personal, con las características que mejor representan al candidato, y finalmente la grabación de una presentación en video de 30 segundos. En esta fase se estudia principalmente la capacidad de implicación, yendo a ver si la persona tiende a ser más o menos introvertida, y su coherencia en las tres fases de presentación. Particular importancia a nivel de selección es cubierta por la video-entrevista que es una especie de entrevista en la que el usuario habla de sí mismo. Esta fase proporciona al reclutador una idea aproximada de las características del candidato a nivel de personalidad, sin sustituir la entrevista de selección individual real, que tendrá lugar en una fecha posterior.

• ¿Cómo te comportas?

En este nivel se ponen a prueba las habilidades de negociación del candidato, es decir, cómo se comporta en relación con otras personas y su predisposición comercial. La evaluación se realiza mediante una simulación de ventas, por ejemplo, en una agencia de viajes con preguntas encrucijadas (dos opciones de elección) colocadas dentro de un video, que abordan la experiencia de manera diferente en función de las respuestas proporcionadas. El objetivo es probar el comportamiento del candidato hacia un cliente, evaluando habilidades como la orientación al cliente o la persuasión (incluso si las preguntas encrucijadas y el patrón de respuesta definido limitan la interacción). Por cada respuesta correcta, el video pasa a una situación diferente, hasta el punto de terminar en el mejor de los casos con la venta del paquete de viaje al cliente, si el usuario ha respondido a las preguntas de la mejor manera posible.

Se puede decir que la atención al cliente, la escucha de sus necesidades y la orientación del servicio son recompensadas consistentemente, en lugar de la venta inmediata del paquete más ventajoso para la empresa.
Además, en esta fase se envían preguntas con una puntuación de respuesta de 0 (nunca) a 5 (siempre), a través de las cuales se sondean las habilidades relacionales con los compañeros en el lugar de trabajo.

• ¿Qué puedes hacer?

Mediante el uso de tres mini juegos se profundizan las habilidades matemáticas, lingüísticas y lógicas, así como la paciencia y precisión de los candidatos.
A continuación, se explican dos ejemplos de posibles juegos. En el primero, el objetivo es crear la mayor cantidad de palabras con las letras disponibles, moviendo sólo una casilla, en el menor tiempo posible. En el segundo, sin embargo, utilizando una grúa oscilante, es necesario poder construir un edificio colocando los pisos del edificio uno encima del otro. El propósito de estos juegos es poner a prueba no solo las habilidades básicas de los usuarios sino también su perseverancia en el desafío, ya que solo completando todas las pruebas se puede obtener una puntuación alta. En este nivel, las habilidades analíticas y de resolución de problemas son el objetivo principal del análisis.

• ¿Qué red tienes?

La última pieza del rompecabezas tiene como objetivo involucrar a la red de contactos del usuario en el proyecto. En la práctica, el candidato tendrá que compartir la plataforma a través de las redes sociales o por correo electrónico para aumentar aún más su puntuación. Por cada nuevo candidato reclutado de esta forma, el usuario recibirá una bonificación equivalente al 10% del rendimiento individual de los usuarios que trajo a la plataforma que se sumará a su puntuación actual en el ranking. El propósito de esta fase es por un lado evaluar el liderazgo del talento en las redes sociales y por otro hacer viral la iniciativa de la empresa. Tampoco se debe subestimar el análisis de comportamientos de relación con los demás, como la construcción de redes, que podría ser predictivo de la posesión de habilidades para el trabajo en equipo.
La dinámica de gamificación acabada de exponer, nos serviría, sobre todo, mediante los comentarios proporcionados por los candidatos, para darnos los inputs necesarios para el posterior análisis con nuestros algoritmos.

Esta herramienta se utiliza también porque en una entrevista presencial, el gerente no siempre es capaz de captar las habilidades blandas ("soft skills") del candidato que tiene frente a él, ya que no son habilidades técnicas y no se pueden probar con simples preguntas cognitivas.

La gamificación se aplicaría a los entornos de selección y formación de empleados con el fin de que los métodos de evaluación sean más parecidos a un juego, mejorando así las reacciones de los candidatos y actuales empleados, aumentando, posiblemente, la predicción del desempeño laboral (Armstrong, Ferrell, Collmus, & Landers, 2016).

Insertar este elemento permite al gerente evaluar el comportamiento de la persona en una situación aparentemente real pero ciertamente más relajada y menos estresante para que el candidato sea capaz de expresar sus cualidades de la mejor manera posible sin temer el juicio del gerente de recursos humanos.

Además, el uso de métodos de selección gamificados podría conducir a una mayor implicación y percepciones positivas de la organización, ya que estas herramientas están a la vanguardia de la tecnología y ofrecen una ventaja competitiva en la guerra por el talento (Fetzer, McNamara, & Geimer, 2017); de hecho, la herramienta podría congraciarse con los futuros empleados del nuevo milenio que, queriendo probar esta nueva forma de contratación, quieran ser incluidos en la empresa que la práctica.

Finalmente, el uso de la gamificación resuelve el problema que generan las redes sociales, o la falta de transparencia que las caracteriza: el candidato puede no ser del todo sincero al informar a la empresa de su experiencia laboral previa y sus conocimientos y habilidades.

La ejecución de evaluaciones gamificadas en línea, por otro lado, podría simular situaciones en las que se muestran las intenciones y los comportamientos de los individuos porque, según el tipo de diseño del juego y los elementos utilizados en las evaluaciones, la atención de los solicitantes podría desviarse del hecho que son evaluados, mostrando así sus comportamientos reales y, en consecuencia, se reducirán los falsos prejuicios de deseabilidad social (Armstrong, Landers, & Collmus, 2017).

Para ello, esta metodología no solo es innovadora y apreciada por los trabajadores, sino que también tiene la capacidad de mejorar la previsión de desempeño laboral, y esto representa una ventaja considerable para el empleador y para la empresa.

Sin embargo, es necesario considerar que las empresas siempre deben continuar investigando e implementando software cada vez más sofisticado y moderno.

4 OBJETIVOS

El **objetivo general** de este Trabajo Final de Máster es diseñar un algoritmo de Machine Learning que nos permita predecir la personalidad de un sujeto en base al indicador MBTI a partir de comentarios realizados por el mismo, de forma de aplicar los resultados a procesos de selección y formación de recursos

Para lograr lo mencionado anteriormente se presentan los siguientes **objetivos específicos:**

1) Seleccionar el set de datos a utilizar y realizar un análisis exploratorio
2) Realizar el pre procesamiento de datos
3) Crear y poner en marcha diferentes modelos de Machine Learning
4) Comparar los diferentes modelos de ML y análisis cuál de ellos se ajusta mejor a los datos, para comenzar con su aplicación real.

Con el término de este proyecto se puede lograr o plantear objetivos a largo plazo que nos proporcionen una visión más global de los beneficios que podrían conllevar, mediante su implementación, en la labor de contratación en los departamentos de Recursos Humanos. A continuación, se detallan los siguientes objetivos:

- Contratar y gestionar los talentos más capaces para gobernar la realidad empresarial que les rodea ya que con ellos se construyen e implementan las dinámicas de negocio que permiten a la organización obtener una posición destacada y prestigiosa frente a sus competidores.
- Seleccionar el mejor candidato para la empresa, asegurando que la atención del gerente se centre más en otros objetivos empresariales que requieren una mayor comprensión y estudio.
- Facilitar y hacer que la cuestión de selección y contratación de personal requiera menos tiempo respecto al método tradicional implementado en los últimos años.
- Atraer, mediante estas técnicas de selección vanguardistas tal como se ha comentado en el apartado anterior del “Estado del arte”, y encontrar aquellos talentos con las habilidades adecuadas para cumplir ciertos roles específicos en la organización ya que a las empresas les genera una gran dificultad para mantenerse activas y reactivas en el mercado.

5 SOLUCIÓN PLANTEADA

Adaptar algoritmos de IA que permitan predecir el tipo de personalidad de un sujeto en función del test MBTI, disminuyendo posibles sesgos que pueden existir cuando una persona está completando el test correspondiente. Y compararlos entre ellos para conocer cual se se ajusta mejor a los datos y nos entrega mejores predicciones.

5.1 METODOLOGÍA

5.1.1 Tecnologías y Herramientas

Nuestra elección recayó en Python porque es particularmente cómodo, permitiendo escribir código limpio, de fácil lectura y condensando funciones complejas en unas pocas líneas. Para la realización del caso de estudio, hemos utilizado un Jupyter Notebook: un documento interactivo que nos permite escribir y ejecutar código en *chunks* (fragmentos). Se considera como una de las herramientas más populares en la ciencia de datos y en el aprendizaje automático porque permite realizar todos los pasos necesarios para completar un análisis de datos eficiente y productivo en un solo documento. Además, un documento de este tipo admite varios lenguajes de programación. Por último, lo más importante es que un Jupyter Notebook se puede compartir fácilmente.

Hay que considerar que, para poder utilizar este documento interactivo, es necesario haber instalado previamente Anaconda, una distribución gratuita y de código abierto de los lenguajes de programación Python y R.

Para el correcto desarrollo de nuestro caso práctico en Python, hemos necesitado importar diferentes librerías, es decir, un conjunto de rutinas y funciones escritas que realizan una tarea específica y que se pueden utilizar según sea necesario. Estas librerías se han recogido en 3 apartados: misceláneo, procesamiento del Lenguaje Natural y Aprendizaje Automático (Machine Learning).

A continuación, se detallan las librerías escogidas:

Misceláneo

Pandas: proporciona las herramientas para el análisis de datos. El paquete es de código abierto y viene con diferentes estructuras de datos que se pueden usar para diferentes tareas de manipulación de datos. Es una librería muy popular para recuperar y preparar datos para su uso futuro en otras librerías de ML como Scikit-learn o Tensorflow. También permite recuperar fácilmente datos de diferentes

fuentes: base de datos SQL, texto, CSV, Excel, archivos JSON y muchos otros formatos menos populares.

Numpy: significa Numeric Python y se considera una de las librerías de cálculo científico y matemático más grandes para Python. Una de las características más importantes de NumPy es su interfaz Array. Esta interfaz se puede utilizar para expresar imágenes, ondas de sonido u otros flujos binarios sin procesar como matrices de números reales con tamaño N. El conocimiento de NumPy es muy importante para el aprendizaje automático y la ciencia de datos.

Matplotlib: hasta el mejor y más sofisticado proyecto de aprendizaje automático no tiene sentido si no lo se puede comunicar a otras personas. Entonces, ¿cómo transforma realmente el valor de todos estos datos que tiene? Aquí es donde Matplotlib viene al rescate. Es una librería estándar de Python que se utiliza para crear tablas y gráficos en 3D. Es de nivel bastante bajo, lo que significa que requiere más comandos para generar gráficos y figuras agradables que algunas librerías avanzadas. Sin embargo, la desventaja es la flexibilidad. Con suficientes comandos, se pueden crear prácticamente cualquier tipo de gráfico que se desee: desde gráficos de columnas y de dispersión hasta gráficos con coordenadas no cartesianas.

Seaborn: es una extensión estadística de Matplotlib. La librería está repleta de muchas funciones, con el objetivo de ayudar a comprender mejor los datos. Esta librería está poniendo la visualización y exploración de datos en el centro del análisis de datos. Una de las cosas que hacen que el paquete sea excelente es el amplio espectro de gráficos disponibles para analizar las relaciones entre múltiples variables. Seaborn también funciona bien con variables categóricas, muestras estadísticas agregadas y gráficos de conteo.

Defaultdict: es una subdivisión de la clase dict. Su importancia radica en el hecho de que permite que a cada nueva clave se le asigne un valor predeterminado en función del tipo de diccionario que se esté creando.

Re: es la abreviación de “expresiones regulares” y son patrones, descritos con sintaxis formal, para buscar coincidencias en un texto. Los patrones se interpretan como un conjunto de instrucciones, que luego se ejecutan con una cadena como entrada para producir un subconjunto de coincidencias o una versión modificada de la cadena original.

Procesamiento del Lenguaje Natural

NLTK: es una librería que permite el procesamiento de lenguaje y el análisis de texto en general. El primer paso para construir el vocabulario lexical en un documento es la transformación del texto en una secuencia de tokens. La función de NLTK para hacer la tokenización de palabras es "**word_tokenize**". En cambio, para la tokenización de tweets usamos la función "**TweetTokenizer**".

Para implementar la lematización en Python, existen numerosos paquetes, como por ejemplo la clase "**WordNetLemmatizer**" que forma parte del paquete NLTK. Estas técnicas utilizan un vocabulario para definir la raíz correcta de los términos.

Por finalizar este apartado, las últimas dos librerías importadas son "**stopwords**" y "**wordnet**". La primera nos permite eliminar o filtrar aquellas palabras que no son útiles para nuestro análisis, no tienen sentido en el aprendizaje porque no tienen conexiones con los sentimientos. Por lo tanto, eliminarlos ahorra poder computacional y aumenta la precisión del modelo. La segunda, en cambio, se utiliza para buscar sinónimos y eliminar la ambigüedad de términos. WordNet es una base de datos semántico-lexical para el idioma inglés que tiene como objetivo organizar, definir y describir los conceptos expresados de las palabras.

Aprendizaje Automático (Machine Learning)

Sklearn: Scikit-learn es una librería de aprendizaje automático de código abierto para el lenguaje de programación Python. Contiene algoritmos de clasificación, regresión y agrupamiento y admite máquinas vectoriales, regresión logística, clasificador bayesiano, k-mean y DBSCAN, y está diseñado para funcionar con librerías como NumPy entre otras.

Antes de dejar que nuestros datos se entrenen, tenemos que representar numéricamente los datos pre-procesados. Las técnicas más conocidas para la vectorización de palabras en el procesamiento del lenguaje natural son: "**CountVectorizer**" que nos permite transformar las palabras de un texto en un número y "**TfidfTransformer**" que tiene como objetivo definir mejor la importancia de una palabra para un documento, al tiempo que tiene en cuenta la relación con otros documentos del mismo corpus.

La siguiente librería "**train_test_split**" nos permite dividir el conjunto de datos en conjunto de entrenamiento y conjunto de test.

“StratifiedKFold” es una variación de k-fold que devuelve pliegues estratificados: cada conjunto contiene aproximadamente el mismo porcentaje de muestras de cada clase objetivo que el conjunto completo.

“Cross_val_score” nos permite calcular la puntuación de validación cruzada directamente utilizando el ayudante cross_val_score. Dado un estimador, el objeto de validación cruzada y el conjunto de datos de entrada, cross_val_score divide los datos repetidamente en un conjunto de entrenamiento y de prueba, entrena al estimador utilizando el conjunto de entrenamiento y calcula las puntuaciones según el conjunto de pruebas para cada iteración de la validación cruzada.

“**MultinomialNB**” es adecuado para la clasificación con características discretas (por ejemplo, recuento de palabras para la clasificación de texto). La distribución multinomial normalmente requiere recuentos de características enteras. Sin embargo, en la práctica, los recuentos fraccionarios como tf-idf también pueden funcionar.

Las “**SVM**” son modelos de clasificación cuyo objetivo es encontrar la línea de separación de las clases que maximiza el margen entre las clases en sí, donde el margen es la distancia mínima desde la línea a los puntos de las dos clases.

El “**LogisticRegression**” es un modelo de clasificación que se utiliza en el aprendizaje automático para entrenar un algoritmo para clasificar correctamente los datos. Es un modelo lineal de clasificación binaria o multiclase.

El “**KNeighborsClassifier**” es uno de los algoritmos más conocidos en el aprendizaje automático que, además de su sencillez, produce buenos resultados en una gran cantidad de dominios. Su propósito es predecir una nueva instancia conociendo los datos que se separan en diferentes clases.

El **“RandomForestClassifier”** es un tipo de algoritmo de aprendizaje automático supervisado basado en el aprendizaje de múltiples modelos de pronóstico para formar un modelo de pronóstico único y más potente. Cada modelo utilizado por el pronóstico de Random Forest suele ser un árbol de decisiones. Esto significa que un bosque aleatorio combina muchos árboles de decisión en un modelo. Individualmente, las predicciones hechas por los árboles de decisión individuales pueden no ser precisas, pero combinadas, las predicciones estarán en promedio más cerca del resultado. El algoritmo de bosque aleatorio se puede utilizar para problemas de regresión y clasificación.

Xgboost: este algoritmo se ha convertido en una herramienta muy poderosa y muy popular en el aprendizaje automático. Esta librería en sí contiene una variedad de funciones y métodos. Se combinan en un solo paquete y nos permite modelar varias actividades de clasificación o regresión utilizando árboles y funciones lineales, aplicando diferentes esquemas de regularización y estableciendo muchos parámetros para resolver estos problemas de manera efectiva. En este caso nos hará la función de clasificación (**XGBClassifier**).

5.1.2 Desarrollo de la herramienta

Junto con este documento se entrega un documento en Jupyter como Anexo 1, el cual recoge el desarrollo realizado durante la construcción de la herramienta objetivo del propio Trabajo Fin de Máster. Pretende constituirse como un documento complementario al informe entregado, y está construido desde un enfoque didáctico y progresivo, justificando de esta forma las decisiones tomadas y acciones acometidas a lo largo de la etapa de desarrollo.

Este cuaderno, siguiendo una estructuración pareja a la expuesta en el propio informe, se divide en 4 puntos. En primer lugar, se presenta y obtiene el (1) conjunto de datos objetivo. Una vez importado en el entorno, se detalla la primera toma de contacto con el mismo a través de un breve (2) análisis exploratorio. Tras identificar las características e idiosincrasia del conjunto de datos, pasaremos a la etapa de (3) pre-procesamiento, por la cual aplicaremos las modificaciones necesarias sobre los datos originales de cara a la obtención de una nueva versión de estos, limpia y optimizada para su posterior tratamiento y consumo por un modelo de IA. A continuación, nos adentraremos en la (4) construcción y testeo del modelo en base a la aplicación de distintos algoritmos, lo que finalmente nos permitirá seleccionar el que mejor se ajuste a nuestro conjunto de datos estudiando los (5) resultados obtenidos.

5.1.3 Conjunto de datos

E conjunto de datos con los que vamos a trabajar es obtenido de la plataforma Kaggle, en el siguiente link: **https://www.kaggle.com/datasnaek/mbti-type**

El indicador MBTI sobre el que construye este conjunto de datos es un test de personalidad que asigna a cualquier individuo un tipo de personalidad recogido dentro de

un conjunto de 16 posibles. Cada uno de estos tipos se corresponde con una posible combinación de las 4 componentes que lo conforman, donde cada una de las mismas toma un valor respecto a un rango binario:

Primera Componente: Introversion (I) – Extroversion (E)

Segunda Componente: Intuition (N) – Sensing (S)

Tercera Componente: Thinking (T) – Feeling (F)

Cuarta Componente: Judging (J) – Perceiving (P)

Cada registro de este conjunto de datos se corresponde con una persona, identificando para cada una su tipo de personalidad en base al indicador MBTI, y detallando los últimos 50 posts realizados por la misma en redes sociales.

Para importar este conjunto de datos, formateado como un archivo separado por comas (CSV), a nuestro entorno y comenzar a trabajar con él, simplemente es necesario hacernos con el fichero proporcionado en la plataforma Kaggle, e importarlo sobre un objeto DataFrame por medio de la librería Pandas.

```
df = pd.read_csv("Data/MBTI.csv")
```

Figura 1. Importar datos.

5.1.4 Análisis exploratorio

Una vez cargado el conjunto de datos podemos comenzar a trabajar con el mismo. La primera fase sobre la que construiremos nuestro desarrollo es la de análisis exploratorio. Mediante la misma seremos capaces de examinar las propiedades y peculiaridades del conjunto de datos original, e ir bocetando en base a las mismas el trabajo a realizar en la posterior etapa de pre procesamiento.

Para ello, comenzaremos analizando de forma general el dataset o métricas generales, permitiéndonos entender el volumen y principales atributos con los que vamos a trabajar. A continuación, examinaremos en detalle las dos dimensiones sobre el que sustenta el mismo con la tipología de personalidad, y el listado de posts.

5.1.5 Métricas generales

A la hora de comenzar a trabajar con un conjunto de datos, una de las primeras operaciones que podemos realizar sobre el mismo es la extracción y visualización de una pequeña porción de los datos. Una forma de hacerlo es a través de la función head().
En este primer extracto podemos ver que, tal como se especificaba en la documentación adjunta al conjunto de datos, que contamos simplemente con dos propiedades (columnas) por cada persona (fila): el tipo de personalidad MBTI, y una concatenación de los últimos 50 posts publicados por la misma en redes sociales. De un simple vistazo podemos identificar alguna de las acciones que deberemos acometer en posteriores pasos. La columna posts responde a una naturaleza no estructurada, lo que requerirá de soluciones asociadas al procesamiento del lenguaje natural. En esta vemos cómo se pueden incluir URL's o caracteres especiales, además de los propios delimitadores de los diferentes posts, y que, por tanto, todos estos escenarios requerirán de un trabajo de pre-procesado.

```
df.head()
```

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...

Figura 2. Head de set de datos.

La propiedad shape de un dataframe nos indica las dimensiones de esta. Tal como especifica su documentación, este conjunto de datos cuenta con un total de 8675 ejemplos, asociado cada uno a una determinada persona, y donde tomando en consideración la construcción de la propiedad "posts", resultaría en un total de 8675 x 50 = 433750 interacciones en redes sociales. En este aspecto, podemos considerarlo como un volumen de información válido para la construcción de un modelo de inteligencia artificial.

```
df.shape
```

```
(8675, 2)
```

Figura 3. Cantidad de datos en dataframe.

Por otro lado, a través de la función info(), obtendremos una visión algo más extensa de la estructura del conjunto de datos. En este caso particular nos es de gran ayuda para observar como no necesitaremos implementar un mecanismo de gestión de valores nulos, ya que los datos originales no contienen valores sin especificar.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8675 entries, 0 to 8674
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   type    8675 non-null   object
 1   posts   8675 non-null   object
dtypes: object(2)
memory usage: 135.7+ KB
```

Figura 4. Información de dataframe.

5.1.6 Tipología de personalidad

En esta sección nos centraremos en la columna "Type", la cual define el tipo de personalidad MTBI asociada a cada persona.

Siendo esta tipología la variable dependiente de nuestro modelo, y, por tanto, el objetivo de predicción, es interesante observar cómo se encuentra distribuida a lo largo del conjunto de datos. Para ello, podemos simplemente contar el número de registros existentes por cada categoría.

Una primera conclusión que podemos obtener a partir del resultado mostrado es que la columna "type" no incluye ningún valor fuera de las 16 tipologías de personalidad existentes, por lo que no necesitaremos ningún trabajo de depuración en ese sentido.

```
[ ] count_by_type = df["type"].value_counts()
    count_by_type

    INFP    1832
    INFJ    1470
    INTP    1304
    INTJ    1091
    ENTP     685
    ENFP     675
    ISTP     337
    ISFP     271
    ENTJ     231
    ISTJ     205
    ENFJ     190
    ISFJ     166
    ESTP      89
    ESFP      48
    ESFJ      42
    ESTJ      39
    Name: type, dtype: int64
```

Figura 5. Cantidad de información por tipos de personalidad.

Con el objetivo de proporcionar un análisis más visual, podemos soportarnos sobre las librerías matplotlib y seaborn para conformar un gráfico que represente esta distribución. Podemos ver como desafortunadamente contamos con un dataset poco balanceado, donde el volumen de ejemplos por cada categorización de personalidad se encuentra desigualmente repartido. Este es un aspecto que deberemos tener en cuenta a la hora de evaluar los resultados del modelo generado, cuya efectividad podrá verse afectada por ello. Igualmente nos permite plantear un primer punto de trabajo futuro, que se basaría en la nueva recopilación de datos para la obtención de un conjunto más equilibrado.

```
[ ] # Representación gráfica de la distribución obtenida.
    plt.figure(figsize = (16, 4))
    plt.ylabel("Persons", fontsize = 12)
    plt.xlabel("Personality Type", fontsize = 12)
    sns.barplot(count_by_type.index, count_by_type.values, alpha = 0.8)
    plt.show()
```

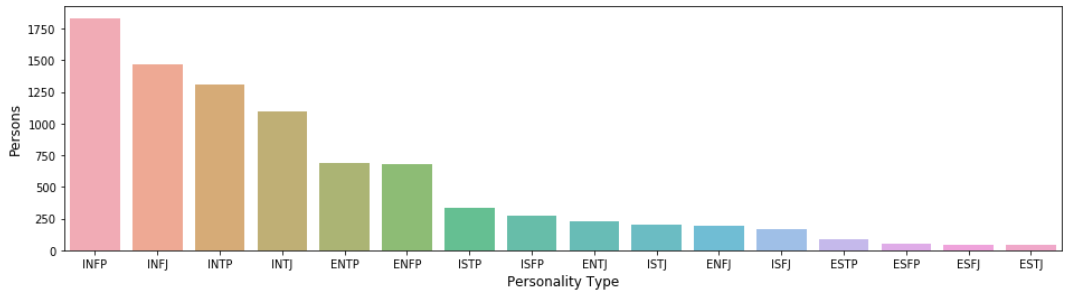

Figura 6. distribución de personalidades en el dataset.

Otro de los análisis que podemos sobre el tipo de personalidad es la distribución de la misma a lo largo del conjunto de datos, pero esta vez enfrentando los ejes de cada una de sus componentes.

```
[ ]  # Función que, dado un ejemplo del conjunto de datos, retorna una serie diseccionando las 4
     # componentes del tipo de personalidad.
     def get_types_detail(person):
         aux_type = person["type"]

         IE = aux_type[0]
         NS = aux_type[1]
         TF = aux_type[2]
         JP = aux_type[3]

         return pd.Series({"IE" : IE, "NS" : NS, "TF" : TF, "JP" : JP})

     # Aplicamos la función definida anteriormente sobre cada ejemplo del dataset, obteniendo
     # una nueva versión del mismo con el detalle del tipo de personalidad.
     types_detail_df = df.join(df.apply(lambda person: get_types_detail(person), axis = 1))
     types_detail_df.head()
```

	type	posts	IE	NS	TF	JP
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	I	N	F	J
1	ENTP	'I'm finding the lack of me in these posts ver...	E	N	T	P
2	INTP	'Good one _____ https://www.youtube.com/wat...	I	N	T	P
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	I	N	T	J
4	ENTJ	'You're fired.|||That's another silly misconce...	E	N	T	J

Figura 7. Modificación del conjunto de datos.

Con el conjunto de datos modificado, obtenemos los valores enfrentados por cada componente.

```
# Obtención de los valores por componente generados en el nuevo dataframe.
IE_I = types_detail_df["IE"].value_counts()["I"]
IE_E = types_detail_df["IE"].value_counts()["E"]

NS_N = types_detail_df["NS"].value_counts()["N"]
NS_S = types_detail_df["NS"].value_counts()["S"]

TF_T = types_detail_df["TF"].value_counts()["T"]
TF_F = types_detail_df["TF"].value_counts()["F"]

JP_J = types_detail_df["JP"].value_counts()["J"]
JP_P = types_detail_df["JP"].value_counts()["P"]

print ("IE:", IE_I, "/", IE_E)
print ("NS:", NS_N, "/", NS_S)
print ("TF:", TF_T, "/", TF_F)
print ("JP:", JP_J, "/", JP_P)
```

```
IE: 6676 / 1999
NS: 7478 / 1197
TF: 3981 / 4694
JP: 3434 / 5241
```

Figura 8. Obtención de valores del nuevo dataframe.

Y, de nuevo, representamos los resultados obtenidos a través de un gráfico para facilitar su comprensión.

```
[ ]  # Representación gráfica de la distribución obtenida.
     top = (IE_I, NS_N, TF_T, JP_J)
     bottom = (IE_E, NS_S, TF_F, JP_P)

     pos = np.arange(4)
     width = 0.7

     plt.ylabel("Persons", fontsize = 12)
     plt.xlabel("Personality Type Components", fontsize = 12)
     plt.xticks(pos, ("I/E",  "N/S", "T/F", "J/P"))
     plt.bar(pos, bottom, width)
     plt.bar(pos, top, width, bottom)
     plt.show()
```

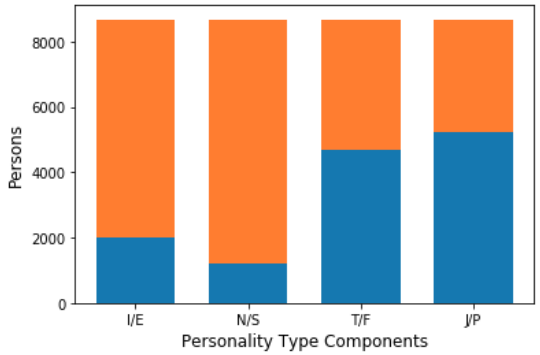

Figura 9. Gráfica de tipos de personalidades.

En este caso, podemos ver como el mal balanceo del conjunto de datos reside directamente sobre las dos primeras componentes de la tipología de personalidad. Mientras que los ejemplos disponibles para (Thinking - Feeling) y (Judging - Perceiving) se encuentran razonablemente parejos, disponemos de una notable disparidad entre I/E (Introversion - Extroversion) y N/S (Intuition - Sensing). Es por ello que en el trabajo futuro que referenciamos anteriormente, sería interesante focalizar sobre la equiparación de estas dos primeras componentes, sin que ello resulte en un desbalanceo del resto, obviamente; y que en el desempeño del algoritmo, será más factible que sean las predicciones asociadas a estas aquellas que se ven más afectadas.

5.1.7 Análisis columna “Post”

Una vez explorada la columna "Type", haremos lo mismo con "Posts". Esta columna recoge los 50 últimos comentarios en redes sociales de la persona asociada al registro, correspondiéndole por tanto con una naturaleza no estructurada.

En este sentido, el uso que vamos a darle a este dato está relacionado con las propias palabras contenidas en los textos proporcionados. Una primera muestra que puede sernos de interés es simplemente medir la extensión de texto asociada a cada ejemplo, y ver si la densidad resultante está razonablemente asentada. Al igual que antes, definimos un nuevo dataset auxiliar que implemente esta métrica.

```
[ ]  # Función que, dado un ejemplo de datos, retorna una serie (valor único) que contiene el número
     # caracteres que compone cada concatenación de posts.
     def get_posts_length(person):
         aux_posts = person["posts"]

         return pd.Series({"posts_length" : len(aux_posts)})

     # Aplicamos la función definida anteriormente sobre cada ejemplo del dataset, obteniendo
     # una nueva versión del mismo con la extensión de cada conjunto de posts.
     posts_length_df = df.join(df.apply(lambda person: get_posts_length(person), axis = 1))
     posts_length_df.head()
```

	type	posts	posts_length
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krwIII...	4652
1	ENTP	'I'm finding the lack of me in these posts ver...	7053
2	INTP	'Good one _____ https://www.youtube.com/wat...	5265
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	6271
4	ENTJ	'You're fired.|||That's another silly misconce...	6111

Figura 10. Cantidad de post por tipos de personalidad.

Se representarán la cantidad de post del set de datos.

```
[ ]  # Representación gráfica de los resultados obtenidos.
     plt.figure(figsize = (16, 4))
     sns.distplot(posts_length_df.posts_length, bins = 50)
     plt.ylabel("Density", fontsize = 12)
     plt.xlabel("Posts Length", fontsize = 12)
     plt.show()
```

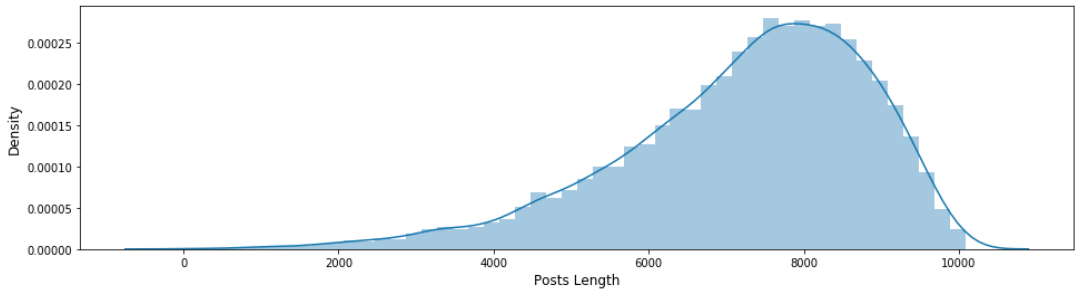

Figura 11. Representación gráfica de post.

Podemos observar como la mayoría de los posts se encuentran en un rango cercano a los 8000 caracteres, conformando de esta forma una muestra razonablemente uniforme en su totalidad.

Otra visión de gran utilidad que podemos obtener del dataset sobre la propiedad "posts" es la extracción de las palabras más empleadas en la totalidad del conjunto. Para ello, tal como mostramos a continuación, podemos conformar un diccionario con las diferentes palabras presentes a lo largo de los posts en redes sociales, y asociar un contador a cada una de ellas que marque su frecuencia.

```
[ ]  # Generamos un diccionario de palabras junto su frecuencia de aparición a lo largo del conjunto.
     posts_dictionary = defaultdict(int)

     for aux_post in df.posts.str.split():
         for aux_word in aux_post:
             posts_dictionary[aux_word] += 1

     # Del diccionario resultante, nos quedamos con las 15 de mayor aparición.
     top_words = sorted(posts_dictionary.items(), key = lambda entry:entry[1] , reverse = True)[:15]
     words, frequencies = zip(*top_words)

     # Representación gráfica de los resultados obtenidos.
     plt.figure(figsize = (16, 4))
     sns.barplot(list(words), list(frequencies))
     plt.ylabel("Frequency", fontsize = 12)
     plt.xlabel("Words", fontsize = 12)
     plt.show()
```

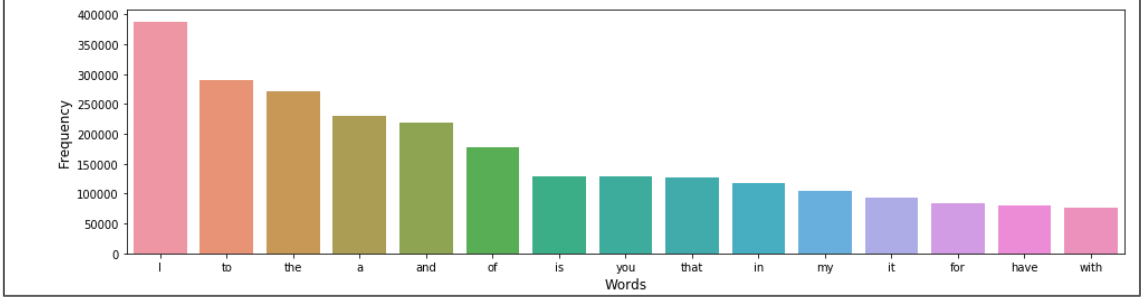

Figura 12. Palabras más utilizadas en post.

Como podemos observar en el gráfico resultante, las palabras más empleadas a lo largo de la colección son, naturalmente, palabras comunes cuyo uso se da transversalmente a cualquier discurso o forma de expresión, por lo que no aportan valor en contenido para realizar ningún tipo de clasificación. De esta forma, podemos concluir que será necesario un trabajo de pre-procesamiento en este sentido, donde nos aseguremos que el contenido de los posts con el que trabajamos realmente sea valioso de cara a la construcción de un mecanismo de asociación entre dicho contenido y la tipología de personalidad de la persona a su cargo.

5.1.8 Pre-procesamiento de los datos

Mediante la etapa de pre-procesamiento transformaremos el conjunto de datos original en una nueva versión del mismo, depurada y optimizada para su uso por algoritmos de inteligencia artificial, manteniendo únicamente la información relevante y de valor para nuestro cometido, así como su transformación a un formato unificado.

En nuestro caso, este trabajo de pre-procesado se centrará en sacar el máximo partido a la propiedad "posts" del conjunto de datos, transformando el espacio de texto libre a un conjunto estructurado que poder tratar. Para ello, tomaremos un ejemplo de la colección completa a modo guía, y comenzaremos con operaciones de Limpieza general, con las que depuramos ciertas de las particularidades observadas en el análisis exploratorio. Una vez obtenida una versión aseada del texto aplicaremos POS Tagging sobre el mismo, que nos permitirá extraer aquel subcontenido que aporte valor real para nuestro cometido. A continuación, mediante el proceso de Lematización, seremos capaces de unificar las diferentes formas en las que puede aparecer una misma palabra, lo que nos permitirá potenciar la efectividad del modelo a desarrollar. Finalmente, una vez definido este flujo de pre-procesamiento para el post extraído, pasaremos a Transformar el conjunto de datos original al completo.

A continuación, se presenta el post de ejemplo sobre el que iremos construyendo el flujo de pre-procesado.

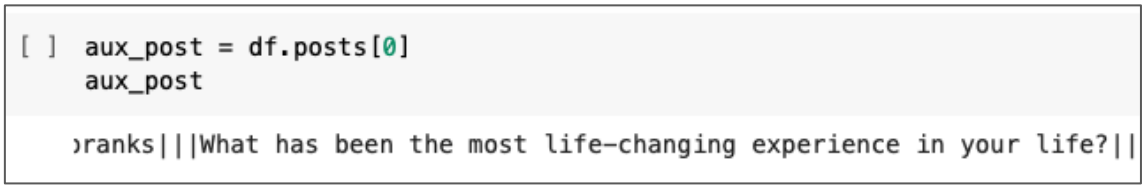

Figura 13. Ejemplo post.

Un ejemplo de post es el siguiente:

> *"'http://www.youtube.com/watch?v=qsXHcwe3krw|||http://41.media.tumblr.com /tumblr_lfouy03PMA1qa1rooo1_500.jpg|||enfp and intj moments https://www.youtube.com/watch?v=iz7lE1g4XM4 sportscenter not top ten plays https://www.youtube.com/watch?v=uCdfze1etec pranks|||What has been the most life-changing experience in your life?|||http://www.youtube.com/watch?v=vXZeYwwRDw8*

http://www.youtube.com/watch?v=u8ejam5DP3E On repeat for most of today.|||May the PerC Experience immerse you.|||The last thing my INFJ friend posted on his facebook before committing suicide the next day. Rest in peace~ http://vimeo.com/22842206|||Hello ENFJ7. Sorry to hear of your distress. It's only natural for a relationship to not be perfection all the time in every moment of existence. Try to figure the hard times as times of growth, as...|||84389 84390 http://wallpaperpassion.com/upload/23700/friendship-boy-and-girl-wallpaper.jpg http://assets.dornob.com/wp-content/uploads/2010/04/round-home-design.jpg ...|||Welcome and stuff.|||http://playeressence.com/wp-content/uploads/2013/08/RED-red-the-pokemon-master-32560474-450-338.jpg Game. Set. Match.|||Prozac, wellbrutin, at least thirty minutes of moving your legs (and I don't mean moving them while sitting in your same desk chair), weed in moderation (maybe try edibles as a healthier alternative...|||Basically come up with three items you've determined that each type (or whichever types you want to do) would more than likely use, given each types' cognitive functions and whatnot, when left by...|||All things in moderation. Sims is indeed a video game, and a good one at that. Note: a good one at that is somewhat subjective in that I am not completely promoting the death of any given Sim...|||Dear ENFP: What were your favorite video games growing up and what are your now, current favorite video games? :cool:|||https://www.youtube.com/watch?v=QyPqT8umzmY|||It appears to be too late. :sad:|||There's someone out there for everyone.|||Wait... I thought confidence was a good thing.|||I just cherish the time of solitude b/c i revel within my inner world more whereas most other time i'd be workin... just enjoy the me time while you can. Don't worry, people will always be around to...|||Yo entp ladies... if you're into a complimentary personality,well, hey.|||... when your main social outlet is xbox live conversations and even then you verbally fatigue quickly.|||http://www.youtube.com/watch?v=gDhy7rdfm14 I really dig the part from 1:46 to 2:50|||http://www.youtube.com/watch?v=msqXffgh7b8|||Banned because this thread requires it of me.|||Get high in backyard, roast and eat marshmellows in backyard while conversing over something intellectual, followed by massages and kisses.|||http://www.youtube.com/watch?v=Mw7eoU3BMbE|||http://www.youtube.com/watch?v=4V2uYORhQOk|||http://www.youtube.com/watch?v=SlVmgFQ

Q0TI|||Banned for too many b's in that sentence. How could you! Think of the B!|||Banned for watching movies in the corner with the dunces.|||Banned because Health class clearly taught you nothing about peer pressure.|||Banned for a whole host of reasons!|||http://www.youtube.com/watch?v=IRcrv41hgz4|||1) Two baby deer on left and right munching on a beetle in the middle. 2) Using their own blood, two cavemen diary today's latest happenings on their designated cave diary wall. 3) I see it as...|||a pokemon world an infj society everyone becomes an optimist|||49142|||http://www.youtube.com/watch?v=ZRCEq_JFeFM|||http://discovermagazine.com/2012/jul-aug/20-things-you-didnt-know-about-deserts/desert.jpg|||http://oyster.ignimgs.com/mediawiki/apis.ign.com/pokemon-silver-version/d/dd/Ditto.gif|||http://www.serebii.net/potw-dp/Scizor.jpg|||Not all artists are artists because they draw. It's the idea that counts in forming something of your own... like a signature.|||Welcome to the robot ranks, person who downed my self-esteem cuz I'm not an avid signature artist like herself. :proud:|||Banned for taking all the room under my bed. Ya gotta learn to share with the roaches.|||http://www.youtube.com/watch?v=w8IgImn57aQ|||Banned for being too much of a thundering, grumbling kind of storm... yep.|||Ahh... old high school music I haven't heard in ages. http://www.youtube.com/watch?v=dcCRUPCdB1w|||I failed a public speaking class a few years ago and I've sort of learned what I could do better were I to be in that position again. A big part of my failure was just overloading myself with too...|||I like this person's mentality. He's a confirmed INTJ by the way. http://www.youtube.com/watch?v=hGKLI-GEc6M|||Move to the Denver area and start a new life for myself.'"

Para medir el impacto de nuestras tareas de pre-procesamiento es interesante, antes y después de cada operación, tomar una muestra del número de palabras existentes en el post tomado como ejemplo y sobre el que desarrollaremos este flujo, de forma que podamos ir apreciando progresivamente el volumen de palabras poco relevantes en contenido que vamos desechando. Vemos como partimos con un total de 779 en este post seleccionado.

```
[ ]  current_words = len(TweetTokenizer().tokenize(aux_post))
     current_words

     779
```

Figura 14. Ejemplo longitud post.

5.1.9 Limpieza general

La primera etapa del pre-procesamiento es una limpieza general de los datos, en base a diversos puntos que hemos detectado en el análisis exploratorio, y con el objetivo de pasar de un texto totalmente crudo a uno ligeramente más unificado. En esta primera transformación empleamos Regex y la sustitución como principales herramientas.

```
[ ]  # Aplica un conjunto de operaciones de limpieza general sobre un post pasado como parámetro.
     def post_general_cleaning(post):
         # Convertimos todo el texto a minúsculas.
         post = post.lower()

         # Sustituimos todas las URL por la palabra clave "Link"
         post = re.sub("http[s]?://(?:[a-z]|[0-9]|[$-_@.&+]|(?:%[0-9a-f][0-9a-f]))+", "link", post)

         # Mantenemos únicamente letras y apóstrofes.
         post = re.sub("[^a-z']", " ", post)

         # Descartamos aquellos apóstrofes que no conformen una expresión.
         post = re.sub("(^|[^a-z])'|'([^a-z]|$)", " ", post)

         # Descartamos todas aquellas palabras de un único caracter que no coincidan con "a" o "i" (únicas existentes)
         post = re.sub("(^|[\s])[^ai](?=([\s]|$))", " ", post)

         # Normalizamos los espacios.
         post = re.sub("\s+", " ", post)

         return post

     # Lazamos la función implementada sobre un post sin ningún tratamiento.
     cleaned_post = post_general_cleaning(aux_post)
     cleaned_post
```

Figura 15. Limpieza de datos post.

Podemos observar que el post quedo de la siguiente forma:

> *"link link enfp and intj moments link sportscenter not top ten plays link pranks what has been the most life changing experience in your life link link on repeat for most of today may the perc experience immerse you the last thing my infj friend posted on his facebook before committing suicide the next day rest in peace link hello enfj sorry to hear of your distress it's only natural for a relationship to not be perfection all the time in every moment of existence try to figure the hard times as times of growth as link link welcome and stuff link game set match prozac*

wellbrutin at least thirty minutes of moving your legs and i don't mean moving them while sitting in your same desk chair weed in moderation maybe try edibles as a healthier alternative basically come up with three items you've determined that each type or whichever types you want to do would more than likely use given each types cognitive functions and whatnot when left by all things in moderation sims is indeed a video game and a good one at that note a good one at that is somewhat subjective in that i am not completely promoting the death of any given sim dear enfp what were your favorite video games growing up and what are your now current favorite video games cool link it appears to be too late sad there's someone out there for everyone wait i thought confidence was a good thing i just cherish the time of solitude i revel within my inner world more whereas most other time i'd be workin just enjoy the me time while you can don't worry people will always be around to yo entp ladies if you're into a complimentary personality well hey when your main social outlet is xbox live conversations and even then you verbally fatigue quickly link i really dig the part from to link banned because this thread requires it of me get high in backyard roast and eat marshmellows in backyard while conversing over something intellectual followed by massages and kisses link link link banned for too many b's in that sentence how could you think of the banned for watching movies in the corner with the dunces banned because health class clearly taught you nothing about peer pressure banned for a whole host of reasons link two baby deer on left and right munching on a beetle in the middle using their own blood two cavemen diary today's latest happenings on their designated cave diary wall i see it as a pokemon world an infj society everyone becomes an optimist link link link link not all artists are artists because they draw it's the idea that counts in forming something of your own like a signature welcome to the robot ranks person who downed my self esteem cuz i'm not an avid signature artist like herself proud banned for taking all the room under my bed ya gotta learn to share with the roaches link banned for being too much of a thundering grumbling kind of storm yep ahh old high school music i haven't heard in ages link i failed a public speaking class a few years ago and i've sort of learned what i could do better were i to be in that position again a big part of my failure was just overloading myself with too i like this person's mentality he's a confirmed intj by the way link move to the denver area and start a new life for myself "

5.1.10 POS Tagging

El siguiente paso es el etiquetado gramatical. Mediante este proceso seremos capaz de marcar una palabra en el texto gramaticalmente, en función de la parte del discurso en la que se encuentre y su contexto. Esta funcionalidad está contenida dentro de la librería NLTK, siendo esta la librería más estandarizada de procesado del lenguaje natural en Python.

El objetivo del POS Tagging es la identificación de las palabras más relevantes en cuanto a contenido respecto al documento. Por ello, una vez etiquetadas, nos quedaremos únicamente con aquellas palabras asociadas a nombres, adjetivos, verbos y adverbios, representando estas las categorías más significativas en valor de discurso. En la misma dirección, aprovecharemos para eliminar del conjunto de datos las denominadas "stop words", que podrían definirse como aquellas palabras carentes de significado por sí mismas.

```
[ ] # Aplica POS Tagging a un post pasado como parámetro, resultando una lista de palabras significantes en contenido.
    def post_pos_tagging(post):
        # Tokenizamos el texto.
        post = TweetTokenizer().tokenize(post)

        # Etiquetamos gramaticalmente cada palabra presente en el texto. Empleamos el tagset universal.
        post = nltk.pos_tag(post, tagset = "universal")

        # Nos quedamos únicamente con aquellas palabras que aporten valor en contenido.
        empty_words = stopwords.words("english")
        significant_tags = ["NOUN", "ADJ", "VERB", "ADV"]
        post = list(filter(lambda x: x[0] not in empty_words and x[1] in significant_tags, post))

        return post

    # Lazamos la función implementada sobre un post previamente limpiado.
    pos_tagged_post = post_pos_tagging(cleaned_post)
    pos_tagged_post

    [('link', 'NOUN'),
     ('link', 'NOUN'),
     ('enfp', 'NOUN'),
     ('intj', 'ADJ'),
     ('moments', 'NOUN'),
     ('link', 'VERB'),
     ('sportscenter', 'ADJ'),
     ('top', 'ADJ'),
     ('ten', 'NOUN')
```

Figura 16. POS Tagging

Tras el etiquetado gramatical, conseguimos reducir el número de palabras a tomar en cuenta de las 596 post limpieza, a estas 321.

```
[ ] current_words = len(pos_tagged_post)
    current_words

    321
```

Figura 17. Post limpieza.

5.1.11 Lematización

Mediante la lematización no incurrimos en una reducción de palabras explícita, pero sí conseguiremos unificar el conjunto actual y potenciar el desempeño del modelo predictor. La lematización es un proceso por el cual, dada una palabra en cualquier de sus formas, obtenemos su raíz o lema. Este lema es el aceptado como representante de todas las palabras que deriven de ella, permitiéndonos agrupar una misma palabra que se encontrase expresada en diferentes formas, pero que al fin y al cabo para nuestro propósito deben ser consideradas como idénticas, ya que proporcionan el mismo valor en contenido. Esta transformación permitirá que el trabajo realizado en la construcción del modelo, expuesto más adelante en este mismo documento, sea más efectivo, valorando la relevancia de una palabra en un documento y en la colección total en base a su raíz.

La funcionalidad de lematización empleada se encuentra contenida, de nuevo, a través de la librería *NTLK*, e implementada a través del denominado *WordNetLemmatizer*.

Como podemos observar, para un correcto desempeño del lematizador es necesario incluir el etiquetado gramatical de la palabra, obtenido en el paso anterior.

```
[ ]  # Función auxiliar para transformar el tag universal al requerido por el lematizador empleado.
     def universal_to_wn(tag):
         if tag == "ADJ":
             return wn.ADJ
         elif tag == "VERB":
             return wn.VERB
         elif tag == "ADV":
             return wn.ADV
         else:
             return wn.NOUN

     # Aplica Lematización a un post pasado como parámetro, resultando una lista de palabras unificadas sobre
     # forma original.
     def post_lemmatization(post):
         post = [(WordNetLemmatizer().lemmatize(word, universal_to_wn(tag))) for (word, tag) in post]

         return post

     # Lazamos la función implementada sobre un post previamente limpiado y etiquetado gramaticalmente.
     lemmatized_post = post_lemmatization(pos_tagged_post)
     lemmatized_post

     ['link',
      'link',
      'enfp',
      'intj',
```

Figura 18. Lematización.

Aun manteniendo el mismo número de palabras, podemos ver que aquellas con distinta raíz o lema se reducen a 244.

```
[ ]  current_words = len(set(lemmatized_post))
     current_words

     244
```

Figura 19. Post lematización.

5.1.12 Aplicación al Conjunto de Datos

Finalmente, en esta sección construimos el algoritmo de pre-procesado integrando las funciones implementadas anteriormente en una única lógica. Este algoritmo tomará como parámetro un conjunto de posts crudos, y retornará un nuevo conjunto con la operativa de transformación aplicada para cada uno de ellos. De la misma forma, se implementará una transformación del listado de variables dependientes para adecuarlo a las necesidades de los algoritmos y a las distintas pruebas a realizar.

```
[ ]  # Función que, dado un post pasado como parámetro, implementa el flujo de pre-procesamiento definido anteriormente
     # para los diferentes posts del conjunto.
     def pre_process_post(post):
         post = post_general_cleaning(post)
         post = post_pos_tagging(post)
         post = post_lemmatization(post)

         return post

     # Función que, para el dataframe objetivo, aplica de forma iterativa el flujo de pre-procesamiento definido
     # para los posts publicados, y genera un nuevo dataframe de tipos de personalidad con la variable original
     # y su correspondiente descomposición por componentes, retornando una nueva versión del conjunto original
     # subdividida en un listado de posts procesados y una tabla de personalidades objetivo.
     def pre_process_dataframe():
         # Subdividimos el conjunto original.
         types = df.type
         posts = df.posts

         # Generamos un nuevo dataframe de tipologías de personalidad, manteniendo el valor original y añadiendo la
         # descomposición por componentes. Aprovechamos la función definida previamente.
         personality_types = pd.DataFrame(types)
         personality_types = personality_types.join(personality_types.apply(lambda full_type: get_types_detail(full_type), axis = 1))
         personality_types = personality_types.rename(columns = {"type": "FullType"})

         # Procesamos los posts en base al flujo definido.
         processed_posts = []
         current_post_index = 0
         total_posts = len(posts)

         print("0 / %s Posts Processed" % (total_posts))

         for aux_post in posts:
             current_post_index += 1

             processed_posts.append(pre_process_post(aux_post))

             if current_post_index % 100 == 0 or current_post_index == total_posts:
                 print("%s / %s Posts Processed" % (current_post_index, total_posts))

         return personality_types, processed_posts

     # Lazamos la función implementada sobre el total de posts presentes en el dataset.
     personality_types, processed_posts = pre_process_dataframe()

     0 / 8675 Posts Processed
     100 / 8675 Posts Processed
```

Figura 20. Aplicación al set completo de datos.

A continuación, podemos observar un extracto de la tabla de tipos de personalidad resultante, cuyas propiedades conformarán la variable dependiente del modelo en los diferentes algoritmos a testar.

```
[ ]  personality_types.head()
```

	FullType	IE	NS	TF	JP
0	INFJ	I	N	F	J
1	ENTP	E	N	T	P
2	INTP	I	N	T	P
3	INTJ	I	N	T	J
4	ENTJ	E	N	T	J

Figura 21. Tipos de personalidades.

Igualmente, se presenta un ejemplo de post el trabajo de pre-procesamiento, donde las palabras resultantes servirán de base para la configuración de las variables independientes.

```
[ ]  processed_posts[1]

     ["i'm",
      'find',
      'lack',
      'post',
      'alarm',
      'sex',
      'boring',
      'position',
      'often',
      'example',
      'girlfriend',
      'currently',
      'environment',
      'creatively',
      'use',
      'cowgirl',
      'missionary',
      'enough',
```

Figura 22. Ejemplo post de pre-procesamiento.

5.2 CONSTRUCCIÓN DEL MODELO

Una vez tratado el conjunto de datos original y procesado hacia a una versión reducida y optimizada del mismo, podemos pasar a la construcción del modelo. En esta sección, en primer lugar, aplicaremos la última transformación necesaria a nuestro conjunto de variables independientes o predictoras a través de la medida Tf-Idf, que resultará una matriz de valores lista para ser consumida por cualquier algoritmo de inteligencia artificial. A continuación, presentaremos los criterios de Entrenamiento y testeo de los diferentes algoritmos considerados, para, por último, aplicar los mismos sobre los conjuntos de datos obtenidos y medir su efectividad, constituyendo de esta forma la etapa final de Machine Learning, donde desemboca todo el trabajo realizado hasta el momento.

5.2.1 Transformación Tf-Idf

Tf-Idf (Term frequency - Inverse document frequency) es una métrica que estima cómo de relevante es una determinada palabra en un documento respecto a una colección completa. A groso modo, de cara a la obtención de esta relevancia, calcula el número de veces que la palabra aparece en el documento, y compensa esta cantidad con la frecuencia de aparición de la misma en la colección. De esta forma, una palabra repetida múltiples veces en un documento, pero presente de forma frecuente a lo largo de la colección, supondría una menor relevancia que otra, que con un número idéntico de repeticiones en el documento, fuese menos presente en el conjunto total.

Para aplicar esta métrica a nuestro conjunto de datos, lo primero que debemos definir es el conjunto de palabras que va a representar la colección, y para cada documento, cuantificar el número de veces que cada palabra de esta colección aparece en el mismo. Ambas operaciones quedan soportadas por la función CountVectorizer(). En ella parametrizamos el número máximo de palabras que deseamos que tenga nuestra colección, 1000 en este caso, resultantes tras aplicar el rango de frecuencia de repetición especificado, que hemos definido entre el 10% y el 50%. Esto quiere decir que únicamente formarán parte de nuestra colección aquellas palabras que aparezcan en un mínimo del 10% de documentos, y en un máximo del 50%. Esta configuración nos va a permitir excluir del conjunto final aquellas palabras demasiado frecuentes, que no aporten valor contextual relevante (underfitting), y también aquellas que puedan condicionar demasiado el funcionamiento del algoritmo dada su rareza (overfitting). Es debido a esta primera operación por lo que la lematización realizada en la etapa de pre-procesado era

de suma importancia. La transformación de cada palabra a su lema nos permitirá medir la relevancia de la misma para con el documento en base a su forma original, independientemente de las distintas formas en las que pudiera aparecer expresada.

```
[ ]  # Definimos la función de conteo.
     count_vectorizer = CountVectorizer(
         analyzer = lambda word : word,
         max_features = 1000,
         min_df = 0.1,
         max_df = 0.5
     )

     # Y la aplicamos sobre los posts previamente procesados.
     processed_posts_count = count_vectorizer.fit_transform(np.array(processed_posts, dtype = "object"))
```

Figura 23. Función de Conteo.

Podemos observar como, tras la función aplicada y su correspondiente filtrado, resulta un diccionario basado en 714 palabras distintas, que serán las variables independientes de nuestro algoritmo.

```
[ ]  processed_posts_count.shape

     (8675, 714)
```

Figura 24. Conteo de palabras después del proceso.

Si queremos conocer el listado, es el siguiente como ejemplo.

```
[ ]  count_vectorizer.get_feature_names()

     ['ability',
      'able',
      'absolutely',
      'accept',
      'accurate',
      'act',
      'action',
      'actual',
      'add',
      'admit',
      'advice',
      'affect',
      'afraid',
      'age',
      'ago',
      'ah',
      'allow',
      'almost',
      'alone',
      'along',
      'already',
```

Figura 25. Listado de palabras.

Y un pequeño extracto de la matriz resultante, donde para cada documento de la colección (fila, post), se marca el número de veces que aparece en el mismo la palabra en cuestión (columna, variable independiente).

```
[ ]  print(processed_posts_count.toarray())

     [[0 0 0 ... 0 0 0]
      [0 1 0 ... 0 0 0]
      [2 1 2 ... 0 0 0]
      ...
      [0 0 0 ... 0 0 0]
      [0 2 0 ... 0 0 0]
      [0 1 0 ... 0 0 1]]
```

Figura 26. Matriz resultante de palabras.

Una vez obtenida la matriz de frecuencias podemos aplicar la transformación Tf-Idf, que medirá la relevancia de una palabra para un documento en función de su importancia en el conjunto global. Estos valores se encontrarán entre el rango de 0 y 1, donde a mayor valor, mayor relevancia.

```
[ ]  # Definimos la función de transformación.
     tfidf_transformer = TfidfTransformer()
     processed_posts_tfidf = tfidf_transformer.fit_transform(processed_posts_count).toarray()

     # Mostramos un extracto de la matriz resultante.
     print(processed_posts_tfidf)

     [[0.         0.         0.         ... 0.         0.         0.        ]
      [0.         0.04825483 0.         ... 0.         0.         0.        ]
      [0.16105196 0.05839248 0.14053007 ... 0.         0.         0.        ]
      ...
      [0.         0.         0.         ... 0.         0.         0.        ]
      [0.         0.07352055 0.         ... 0.         0.         0.        ]
      [0.         0.04779301 0.         ... 0.         0.         0.05066328]]
```

Figura 27. Función de transformación y matriz resultante.

5.2.2 Entrenamiento y Test: Cross Validation

Una vez hemos obtenido el conjunto final de variables dependientes e independientes para nutrir a nuestros algoritmos, debemos definir la metodología de entrenamiento y testeo, así como las diferentes pruebas con las que vamos a medir la eficacia de cada una de nuestras aproximaciones.

En primer lugar, para entrenar y analizar el rendimiento de cada prueba, vamos a emplear validación cruzada o "Cross Validation". Mediante este mecanismo podremos evaluar el resultado de cada ejecución, con la seguridad de que este no se encuentra influido por la partición efectuada entre el conjunto de entrenamiento y test. Para ello, la validación cruzada divide el conjunto de datos en un número N de particiones, donde se realizarán N ejecuciones del algoritmo, que será entrenado con N-1 de las mismas, para acabar testeando contra la partición restante, hasta que hayamos testado el mismo contra cada una de las N particiones. Finalmente, se calculará la media aritmética del resultado obtenido por cada una de las N ejecuciones, lo que resultará en una métrica fiable del desempeño del algoritmo sobre el conjunto de datos.

En nuestro caso hemos seleccionado 5 particiones. Esto quiere decir que se efectuarán 5 ejecuciones de cada uno de los algoritmos, donde el conjunto de entrenamiento supondrá el 80% del conjunto de datos, y el 20% restante conformará el conjunto de test. Además, hemos seleccionado la variante "stratified" de la validación cruzada, donde nos aseguramos que el ratio de ejemplos por categoría en el conjunto original se mantenga en cada una de las particiones realizadas. Igualmente, seleccionamos una semilla concreta para hacer las particiones, simplemente para fijar que todos los algoritmos y todas las pruebas que hagamos se realicen sobre la misma distribución de ejemplos a lo largo de las mismas, y que la comparativa al fin y al cabo sea lo más objetiva posible.

```
[ ]  # Definimos el mecanismo de validación cruzada descrito.
     skf = StratifiedKFold(n_splits = 5, shuffle = True, random_state = 20210331)

     # Definimos una función auxiliar para entrenar y medir el rendimiento de cada modelo sobre los conjuntos
     # de entrenamiento y test.
     def train_and_score(model, x, y):
         scores = cross_val_score(model, x, y, cv = skf)
         score = scores.mean()

         return score
```

Figura 28. Definición del mecanismo de validación cruzada.

A continuación, definimos los distintos conjuntos de entrenamiento y test que vamos a emplear en las diferentes pruebas. Naturalmente, el conjunto de variables independientes siempre será común a cada una de las pruebas por modelo, pero no así las variables dependientes u objetivo. Esto se debe a que hemos definido 5 diferentes escenarios para testar cada uno de nuestros modelos. El primero de ellos se corresponderá con la variable objetivo original al conjunto de datos, es decir, el tipo de personalidad completo,

resultando de esta forma en una clasificación en 16 diferentes categorías. Pero, además de ello, mediremos la eficacia de los algoritmos para cada una de las componentes de personalidad de forma independiente, siendo de esta forma capaces de evaluar qué componentes son los que mejor se ajustan a nuestras predicciones, y cuales peor. De esta forma, resultaron los 4 escenarios restantes, que se corresponderá con una tarea de clasificación binaria. En resumen, esta sería cada una de las pruebas a realizar por modelo:

Tipo Completo: Clasificación en las 16 categorías de personalidad.

I-E: Clasificación binaria, Introversión (I) – Extroversión (E)

N-S: Clasificación binaria, Intuition (N) – Sensing (S)

T-F: Clasificación binaria, Thinking (T) – Feeling (F)

J-P: Clasificación binaria, Judging (J) – Perceiving (P)

```
[ ] # Definimos los distintos conjuntos de entrenamiento y test.
    x = processed_posts_tfidf

    y = personality_types.FullType

    y_IE = personality_types.IE
    y_NS = personality_types.NS
    y_TF = personality_types.TF
    y_JP = personality_types.JP

    # Definimos una función auxiliar para formatear los resultados obtenidos en cada modelo para las diferentes
    # pruebas realizadas.
    def print_score(model_name, score, score_IE, score_NS, score_TF, score_JP):
        print("_____________________________________________________________________________\n")
        print("Algorithm:", model_name)
        print("_____________________________________________________________________________\n")

        print("Full Personality Type Classification Score (16 Categories): %0.2f" % (score), "\n")
        print("    * Introversion - Extroversion Classification Score (2 Categories): %0.2f" % (score_IE))
        print("    * Intuition - Sensing Classification Score (2 Categories): %0.2f" % (score_NS))
        print("    * Thinking - Feeling Classification Score (2 Categories): %0.2f" % (score_TF))
        print("    * Judging - Perceiving Classification Score (2 Categories): %0.2f" % (score_JP))
        print("_____________________________________________________________________________\n")
```

Figura 29. Definición de conjunto de entrenamiento y test

5.3 MACHINE LEARNING

En la etapa de aprendizaje automático materializamos todo el trabajo realizado hasta el momento, generando los distintos modelos y midiendo su rendimiento contra el conjunto de datos en base a las pruebas previamente definidas. El objetivo es probar un conjunto de algoritmos, para acabar determinando cual es el más apropiado para nuestra solución. En nuestro caso hemos seleccionado un total de 6 aproximaciones distintas, basadas en 6 modelos típicamente empleados en la resolución de problemas relacionados con el procesamiento del lenguaje natural o NLP, los cuales se mencionan a continuación:

1. Multinomial Naive Bayes.

2. Support Vector Machines.
3. Logistic Regression.
4. Extreme Gradient Boosting.
5. K-Nearest Neighbors.
6. Random Forest.

En el desarrollo de esta sección emplearemos la configuración por defecto de los distintos algoritmos pre-definida en las librerías asociadas, con el objetivo de comprobar el desempeño de los mismos sin parametrizaciones a medida. Igualmente, no entraremos en el detalle matemático sobre el que sustenta cada uno de los algoritmos, ya que consideramos que no es el objetivo de este trabajo. En su lugar, nos centraremos en un enfoque puramente ensayístico, donde analizaremos cada algoritmo en base a la efectividad arrojada por cada uno.

A continuación se proporciona una pequeña reseña de cada modelo, obtenido de Pedregosa et al. 2011.

5.3.1 Multinomial Naive Bayes

Se basa en la aplicación de la Regla de Bayes para predecir la probabilidad condicional de que un documento pertenezca a una clase a partir de la probabilidad de los documentos dada la clase y la probabilidad a priori de la clase en el conjunto de entrenamiento.

5.3.2 Support Vector Machines

Es un conjunto de métodos de aprendizaje supervisado que se utilizan para la clasificación, la regresión y la detección de valores atípicos.

Las principales ventajas son:

Eficaz en espacios de gran dimensión.

Sigue siendo eficaz en los casos en que el número de dimensiones es mayor que el número de muestras.

Utiliza un subconjunto de puntos de entrenamiento en la función de decisión (llamados vectores de soporte), por lo que también es eficiente en la memoria.

Versátil: se pueden especificar diferentes funciones del núcleo para la función de decisión. Se proporcionan núcleos comunes, pero también es posible especificar núcleos personalizados.

5.3.3 Logistic Regresion

La regresión logística, a pesar de su nombre, es un modelo lineal para clasificación en lugar de regresión. La regresión logística también se conoce en la literatura como regresión logit, clasificación de máxima entropía (MaxEnt) o clasificador log-lineal. En este modelo, las probabilidades que describen los posibles resultados de un solo ensayo se modelan utilizando una función logística.

5.3.4 Extreme Gradient Boosting

Este método forma una clase de algoritmos que construyen varias instancias de un estimador de caja negra en subconjuntos aleatorios del conjunto de entrenamiento original y luego agregan sus predicciones individuales para formar una predicción final. Estos métodos se utilizan como una forma de reducir la varianza de un estimador base (por ejemplo, un árbol de decisión), al introducir la aleatorización en su procedimiento de construcción y luego hacer un conjunto a partir de él.

5.3.5 K-Nearest Neighbors

El principio detrás de los métodos de vecino más cercano es encontrar un número predefinido de muestras de entrenamiento más cercanas en distancia al nuevo punto y predecir la etiqueta a partir de ellas. El número de muestras puede ser una constante definida por el usuario (aprendizaje del vecino más cercano k) o variar según la densidad local de puntos (aprendizaje del vecino basado en el radio). En general, la distancia puede ser cualquier medida métrica: la distancia euclidiana estándar es la opción más común. Los métodos basados en vecinos se conocen como métodos de aprendizaje automático no generalizantes, ya que simplemente "recuerdan" todos sus datos de entrenamiento (posiblemente transformados en una estructura de indexación rápida como un árbol de bolas o un árbol KD).

A pesar de su simplicidad, los vecinos más cercanos han tenido éxito en una gran cantidad de problemas de clasificación y regresión, incluidos dígitos escritos a mano y escenas de imágenes de satélite. Al ser un método no paramétrico, a menudo tiene éxito en situaciones de clasificación donde el límite de decisión es muy irregular.

5.3.6 Random Forest

Este algoritmo utiliza técnicas de perturbación y combinación diseñadas específicamente para árboles. Esto significa que se crea un conjunto diverso de clasificadores mediante la introducción de aleatoriedad en la construcción del clasificador. La predicción del conjunto se da como la predicción promedio de los clasificadores individuales.

Como otros clasificadores, los clasificadores de bosque deben estar equipados con dos matrices: una matriz X de forma dispersa o densa que contiene las muestras de entrenamiento, y una matriz Y de forma que contiene los valores objetivo (etiquetas de clase) para las muestras de entrenamiento.

6 EVALUACIÓN

A continuación, se presentan los modelos a evaluar junto a sus resultados en porcentajes de aciertos del set de datos de entrenamiento v/s set de datos de prueba.

6.1 MULTINOMIAL NAIVE BAYES

El modelo y su evaluación es el siguiente:

```
[ ]  model_name = "Multinomial Naive Bayes"
     model = MultinomialNB()

     score_MNB = train_and_score(model, x, y)

     score_MNB_IE = train_and_score(model, x, y_IE)
     score_MNB_NS = train_and_score(model, x, y_NS)
     score_MNB_TF = train_and_score(model, x, y_TF)
     score_MNB_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_MNB, score_MNB_IE, score_MNB_NS, score_MNB_TF, score_MNB_JP)

     ____________________________________________________________________________________

     Algorithm: Multinomial Naive Bayes
     ____________________________________________________________________________________

     Full Personality Type Classification Score (16 Categories): 0.46

         * Introversion - Extroversion Classification Score (2 Categories): 0.77
         * Intuition - Sensing Classification Score (2 Categories): 0.86
         * Thinking - Feeling Classification Score (2 Categories): 0.81
         * Judging - Perceiving Classification Score (2 Categories): 0.68
     ____________________________________________________________________________________
```

Figura 30. Algoritmo multinomial Naive Bayes.

6.2 SUPPORT VECTOR MACHINES

El modelo y su evaluación es el siguiente:

```
[ ]  model_name = "Support Vector Machines"
     model = svm.SVC()

     score_SVC = train_and_score(model, x, y)

     score_SVC_IE = train_and_score(model, x, y_IE)
     score_SVC_NS = train_and_score(model, x, y_NS)
     score_SVC_TF = train_and_score(model, x, y_TF)
     score_SVC_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_SVC, score_SVC_IE, score_SVC_NS, score_SVC_TF, score_SVC_JP)
```

```
_____________________________________________________________________________

Algorithm: Support Vector Machines
_____________________________________________________________________________

Full Personality Type Classification Score (16 Categories): 0.65

    * Introversion - Extroversion Classification Score (2 Categories): 0.85
    * Intuition - Sensing Classification Score (2 Categories): 0.89
    * Thinking - Feeling Classification Score (2 Categories): 0.85
    * Judging - Perceiving Classification Score (2 Categories): 0.80
_____________________________________________________________________________
```

Figura 31. Algoritmo SPM.

6.3 LOGISTIC REGRESION

El modelo y su evaluación es el siguiente:

```
[ ]  model_name = "Logistic Regression"
     model = LogisticRegression(max_iter = 500)

     score_LR = train_and_score(model, x, y)

     score_LR_IE = train_and_score(model, x, y_IE)
     score_LR_NS = train_and_score(model, x, y_NS)
     score_LR_TF = train_and_score(model, x, y_TF)
     score_LR_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_LR, score_LR_IE, score_LR_NS, score_LR_TF, score_LR_JP)
```

```
_____________________________________________________________________________

Algorithm: Logistic Regression
_____________________________________________________________________________

Full Personality Type Classification Score (16 Categories): 0.65

    * Introversion - Extroversion Classification Score (2 Categories): 0.85
    * Intuition - Sensing Classification Score (2 Categories): 0.89
    * Thinking - Feeling Classification Score (2 Categories): 0.85
    * Judging - Perceiving Classification Score (2 Categories): 0.79
_____________________________________________________________________________
```

Figura 32. Algoritmo Logistic Regresion.

6.4 EXTREME GRADIENT BOOSTING

El modelo y su evaluación es el siguiente:

```
[ ]  #import warnings
     #warnings.filterwarnings("ignore")

     model_name = "Extreme Gradient Boosting"

     model = XGBClassifier(use_label_encoder = True, eval_metric = "mlogloss")
     score_EGB = train_and_score(model, x, y)

     model = XGBClassifier(use_label_encoder = True, eval_metric = "mlogloss")
     score_EGB_IE = train_and_score(model, x, y_IE)
     score_EGB_NS = train_and_score(model, x, y_NS)
     score_EGB_TF = train_and_score(model, x, y_TF)
     score_EGB_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_EGB, score_EGB_IE, score_EGB_NS, score_EGB_TF, score_EGB_JP)
```

```
--------------------------------------------------------------------------------

Algorithm: Extreme Gradient Boosting

--------------------------------------------------------------------------------

Full Personality Type Classification Score (16 Categories): 0.66

    * Introversion - Extroversion Classification Score (2 Categories): 0.85
    * Intuition - Sensing Classification Score (2 Categories): 0.90
    * Thinking - Feeling Classification Score (2 Categories): 0.83
    * Judging - Perceiving Classification Score (2 Categories): 0.80

--------------------------------------------------------------------------------
```

Figura 33. Algoritmo Extreme Gradient Boosting.

6.5 K-NEAREST NEIGHBORS

El modelo y su evaluación es el siguiente:

```
[ ]  model_name = "K-Nearest Neighbors"

     model = KNeighborsClassifier(n_neighbors = 100)

     score_KNN = train_and_score(model, x, y)

     score_KNN_IE = train_and_score(model, x, y_IE)
     score_KNN_NS = train_and_score(model, x, y_NS)
     score_KNN_TF = train_and_score(model, x, y_TF)
     score_KNN_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_KNN, score_KNN_IE, score_KNN_NS, score_KNN_TF, score_KNN_JP)
```

```
_____________________________________________________________________________

Algorithm: K-Nearest Neighbors
_____________________________________________________________________________

Full Personality Type Classification Score (16 Categories): 0.57

    * Introversion - Extroversion Classification Score (2 Categories): 0.82
    * Intuition - Sensing Classification Score (2 Categories): 0.87
    * Thinking - Feeling Classification Score (2 Categories): 0.79
    * Judging - Perceiving Classification Score (2 Categories): 0.76
_____________________________________________________________________________
```

Figura 34. Algoritmo K-Nearest Neighbors.

6.6 RANDOM FOREST

El modelo y su evaluación es el siguiente:

```
[ ]  model_name = "Random Forest"

     model = RandomForestClassifier(max_depth = 100)

     score_RF = train_and_score(model, x, y)

     score_RF_IE = train_and_score(model, x, y_IE)
     score_RF_NS = train_and_score(model, x, y_NS)
     score_RF_TF = train_and_score(model, x, y_TF)
     score_RF_JP = train_and_score(model, x, y_JP)

     print_score(model_name, score_RF, score_RF_IE, score_RF_NS, score_RF_TF, score_RF_JP)
```

```
_____________________________________________________________________________

Algorithm: Random Forest
_____________________________________________________________________________

Full Personality Type Classification Score (16 Categories): 0.60

    * Introversion - Extroversion Classification Score (2 Categories): 0.81
    * Intuition - Sensing Classification Score (2 Categories): 0.87
    * Thinking - Feeling Classification Score (2 Categories): 0.82
    * Judging - Perceiving Classification Score (2 Categories): 0.77
_____________________________________________________________________________
```

Figura 35. Algoritmo Random Forest.

7 RESULTADOS

Como resultados del presente trabajo, se presentan los diferentes algoritmos utilizados exponiendo un comparativo respecto de cuáles de ellos se ajustan mejor a la solución planteada. A continuación, se presenta la tabla de resultados obtenidos por cada algoritmo:

	Full Type	I - E	N - S	T - F	J - P
1. Multinomial Naive Bayes	0.46	0.77	0.86	0.81	0.68
2. Support Vector Machines	0.65	**0.85**	0.89	**0.85**	**0.80**
3. Logistic Regression	0.65	**0.85**	0.89	**0.85**	0.79
4. Extreme Gradient Boosting	**0.66**	**0.85**	**0.90**	0.83	**0.80**
5. K-Nearest Neighbors	0.57	0.82	0.87	0.79	0.76
6. Random Forest	0.60	0.81	0.87	0.82	0.77

Tabla 1. Porcentaje de comparación de los algoritmos utilizados. Full type es el porcentaje de acierto a la personalidad.

Lo primero que podemos observar es que, a nivel de rendimiento, existen dos grupos claramente diferenciados con métricas parejas entre sí. Por un lado, tendríamos 3 algoritmos que se ajustan peor a los datos: Multinomial Naive Bayes, K-Nearest Neighbors y Random Forest, donde el nivel de precisión no supera el 60%. Por otro lado, tendríamos Support Vector Machines, Logistic Regression y Extreme Gradient Boosting, siendo este último el que ofrece resultados ligeramente más óptimos.

Analizando las distintas pruebas realizadas podemos obtener diversas conclusiones. En la clasificación en base al tipo completo de personalidad obtenemos, en el peor de los casos, un 46% de efectividad, para pasar a un 66% en el algoritmo que proporciona mayor acierto. Teniendo en cuenta que esta clasificación responde a 16 categorías, lo que conlleva un 6.25% de probabilidad natural de acierto, elevar este índice hasta un 66% es, a nuestro parecer, una buena métrica. También debemos considerar que dada la naturaleza de este proceso únicamente se dan por válidas las predicciones que coinciden en sus 4 componentes al completo, pero que parte de ese 34% restante puede corresponderse con casos que hayan fallado de forma parcial, acertando en parte de la composición del tipo

de personalidad, casuística que actualmente el desarrollo realizado no comprende pero que no podrían considerarse como fallos completamente categóricos.

Pasando al análisis de clasificación binaria, entre los ejes de cada componente del tipo de personalidad, los resultados naturalmente son más satisfactorios. En ese sentido, en nuestra mejor aproximación a través de Extreme Gradient Boosting obtenemos una media del 85% de efectividad, con un índice de clasificación del 80% como más bajo, y un 90% en la componente que mejor responde al análisis. En la predicción binaria podemos ver que de forma transversal al algoritmo empleado los resultados siguen un mismo patrón. La componente de personalidad que mejor responde a las técnicas de predicción por medio del aprendizaje automático es la que se corresponde con la intuición (Intuition) y la racionalidad (Sensing), seguida de la componente referente a introversión (Introversión) y extroversión (Extroversión) y la asociada a la lógica (Thinking) y el sentimiento (Feeling). Un escalón por debajo encontramos la efectividad de clasificación obtenida en el componente de juicio (judging) y comprensión (Perceiving). Lo que podemos deducir de este comportamiento es que puede resultar más sencillo extraer ciertas componentes de personalidad de un individuo a través de su redacción que otras, donde quizás no responden a una manifestación tan explícita en este ámbito, y precisaron si así se requiriese de un análisis paralelo por medio de un mecanismo adicional. En cualquier caso, aunque existen ligeras diferencias, no son tan notables, y los resultados de forma general son óptimos.

De esta forma, podemos concluir que, a pesar de ser una primera aproximación, susceptible de tomar vías de mayor complejidad, y con configuraciones de algoritmos predeterminadas, donde igualmente se puede seguir indagando en una parametrización de los mismos más avanzada, los resultados son notables y prometedores. Creemos que ha quedado expuesto que efectivamente existe un vínculo entre la forma de expresión de los diferentes individuos a través de las redes sociales y la tipología de personalidad con la que se corresponden, permitiendo de esta forma el desarrollo de aplicaciones que se implementen esta extracción de conocimiento y se aprovechen del mismo.

8 TRABAJOS FUTUROS

En cuanto a los puntos de mejora de trabajos futuros que han surgido del desarrollo de nuestro caso de estudio, consideraremos los siguientes:

Balanceo del dataset: recopilación de ejemplos para aquellos tipos de personalidad que se encuentran poco representados para la obtención de un conjunto de datos equilibrado.

Seguir trabajando sobre el pre-procesado de los datos, depurando las técnicas actuales e incluyendo nuevas, de cara a la obtención del conjunto más óptimo posible para ser consumido por los algoritmos.

Profundizar sobre la configuración de los algoritmos más prometedores para incrementar la efectividad de los mismos sobre nuestro conjunto de datos.

Estudiar un enfoque de Deep Learning: un ejemplo podrían ser las redes neuronales convolucionales ya que cuentan con una arquitectura óptima para la detección de patrones y para el reconocimiento de imágenes y campos de clasificación definidos como serían las etiquetas de las diferentes personalidades.

Implementar el trabajo realizado sobre una herramienta software que materialice el desarrollo como un mecanismo de apoyo en el proceso de contratación de una organización.

Cabe mencionar que los datos fueron recopilados a través del foro PersonalityCafe (Canadá), ya que proporciona una gran selección de personas y su tipo de personalidad MBTI. Por lo tanto, en el set de datos puede existir un sesgo, ya que las personalidad y validación del test varían según los países. Por lo tanto, cabe la posibilidad que estos algoritmos sólo sirvan para Canadá. Como trabajos futuros se debería trabajar con set de datos más uniformes en culturas y países diferentes, para poder obtener valores o algoritmos más universales y con aplicabilidad global.

9 CONCLUSIONES

9.1 CONCLUSIONES DEL PROYECTO

Con el siguiente trabajo nos propusimos discutir la importancia del capital humano dentro de las empresas, considerando que este pudiera ser realmente bien percibido tanto por la empresa como por el personal, destacando la importancia de los procesos propios del área de recursos humanos, haciendo foco en los aquellos vinculados a la selección y al desarrollo personal del individuo.

Dejamos de manifiesto que una técnica utilizada para los procesos mencionados en el párrafo anterior es el test MBTI mediante el cual, puede identificarse el tipo de personalidad y comportamiento de una persona, destacando que dicha información podría utilizarse con el fin de seleccionar al candidato que más se ajuste a los requerimientos del ente, así como también lograr que este conozca sus fortalezas y debilidades con miras a la mejora diaria en su trabajo, empoderamiento y liderazgo.

Entendemos el avance que la ciencia de datos está teniendo prácticamente en todos los ámbitos, no siendo los procesos de selección, reclutamiento y capacitación, la excepción. Por este motivo, nos propusimos analizar la capacidad predictiva que diferentes algoritmos de inteligencia artificial tienen respecto al indicador MBTI de cara a aplicarlos en los citados procesos proponiendo una nueva forma de mirar y encarar los mismos, a través de la recolección de comentarios y apreciaciones del sujeto, sea mediante sus mails enviados, comentarios en redes sociales o aquellos construidos a partir de la apreciación de un video o imagen con fines de someterlos al algoritmo elegido y obtener de manera rápida su tipo de personalidad y comportamiento MBTI.

De esta forma, con un set de datos que consta de un poco más de 8.000 comentarios de personas con su respectiva clasificación de MBTI, se ha procedido a construir seis algoritmos de machine learning y a evaluar la precisión de cada uno en dos frentes diferentes:

Por un lado, evaluamos el nivel de acierto del tipo completo de personalidad y, por otro, la clasificación binaria de la misma.

El algoritmo con mejor nivel predictivo en este caso fue Extreme Gradient Boosting, logrando un nivel de aciertos en el tipo completo de personalidad alrededor del 66%, precisión que asciende al 80% en promedio, si evaluamos la clasificación binaria.

Consideramos que estos son buenos resultados, sobre todo si tenemos en cuenta que los tipos de personalidad según el test analizado, son de 16.

Llegamos a la conclusión que la personalidad y comportamiento puede ser definida y pronosticada a partir de la manera en la que un sujeto se manifiesta por escrito, sin embargo, hay que tener en cuenta que la mencionada manifestación escrita debe ser realizada de la manera más natural posible, sin ningún tipo de prejuicio o sesgo, situaciones que se generan, lógicamente, en los procesos de selección.

Sin perjuicio de lo expuesto, creemos que un algoritmo predictivo puede ser utilizado como apoyo a estos procesos y, sin dudas aplicado a dinámicas de grupo con el objetivo de lograr el desarrollo del capital humano, donde seguramente cada colaborador estará dispuesto a manifestarse con sinceridad de forma de que se le diga los puntos de mejora que propicien su crecimiento.

Un punto a considerar, es que estos algoritmos funcionan mucho mejor si el lenguaje en el que es utilizado es el inglés, siendo un punto de mejora, en un futuro, adaptarlo a otros idiomas, considerando, inclusive, las características propias que marcan a una persona, en relación a su país de origen y cultura.

En definitiva, en este ensayo, el lector no solamente puede entender la manera de desarrollar algoritmos, las librerías utilizadas, y los procesos específicos dentro de la ciencia de datos, sino que también obtiene una mirada práctica de su aplicación, dejando marcados, claramente, aquellos aspectos en donde todavía falta mayor investigación y desarrollo de cara a lograr un algoritmo con mayor nivel de precisión.

9.2 APRECIACIONES PERSONALES

9.2.1 Jairo Diego Cuesta

Desde una perspectiva académica y personal, la experiencia ha sido completamente satisfactoria. A lo largo del desarrollo del proyecto he podido consolidar y aplicar parte de los conocimientos adquiridos a lo largo de la titulación, los que sin duda conforman un conjunto de competencias extra que construirán una sólida base y servirán de ayuda de cara a orientar, o al menos tener la posibilidad de hacerlo, mi futuro laboral hacia la ciencia de datos.

Muestra de estos conocimientos serían, por ejemplo, el estudio exploratorio de un conjunto de datos de cara a la identificación del trabajo necesario a realizar sobre el mismo y las posibilidades que este nos ofrece, el análisis y transformación de este a través de mecanismos y librerías específicas, o la ejecución de diferentes tipos de algoritmos de aprendizaje automático y posterior examen de los resultados obtenidos, conformando en

su conjunto una parte del flujo abarcado en un desarrollo real de una herramienta de inteligencia artificial.

En definitiva, y haciendo referencia a mi formación como ingeniero, nuestra tarea primordial es, a groso modo, la de dibujar una solución para un problema concreto con un propósito específico, siendo capaz de resolverlo, o en su defecto, generar un impacto positivo. En este sentido, la realización de este máster me permitirá ampliar el abanico de problemas sobre los que poder trabajar, haciendo uso de nuevas soluciones, y por tanto, propiciando dicho impacto positivo sobre áreas que hasta este momento se encontraban fuera de mi alcance.

9.2.2 Marc Faravelli Rodríguez

Con este trabajo final de máster, he podido afianzar la mayoría de los conocimientos tratados durante el curso. Mediante este programa de máster nos han equipado con las habilidades clave que necesitamos para desarrollar sistemas de información de próxima generación utilizados para describir y administrar datos, descubrir nuevos hechos y relaciones en ellos, hacer predicciones y asesorar a quien corresponda la toma de decisiones.

Ha sido una buena manera de poder poner en práctica muchos conceptos vistos y entendidos por parte de nuestros profesores. Por otro lado, quisiera destacar también el conocimiento adquirido al escuchar las opiniones, la visión e ideas de mis compañeros de proyecto. Creo que hemos podido crear un grupo en el cual hemos sabido complementarnos ya que cada uno de nosotros dominaba más algunas áreas respecto que otras y viceversa.

En cuanto al proyecto en sí, quisimos resaltar la importancia de utilizar herramientas informáticas y digitales que puedan jugar un papel alternativo o sustitutivo para las fases mayoritariamente complejas de la realidad empresarial. En particular, se profundizó en la creación de herramientas para el descubrimiento más eficiente de la personalidad basadas en la gamificación (uso de elementos provenientes de los videojuegos, en contextos no lúdicos) y su uso en procesos de selección de personal en el ámbito de la Gestión de Recursos Humanos.

9.2.3 Francisco Olivera Maturana

Este trabajo de fin de máster personalmente fue un desafío, ya que es un área completamente diferente a la que me desenvuelvo (Biologo Marino), pero si rescato lo relevante de trabajar en un ambiente diferente, me refiero a los datos, pero que a su vez es similar en la forma de trabajar y resolver las problemáticas. Es decir, los datos son datos, ahora bien, que vengan en palabras, números, signos, etc, siempre es información y se debe buscar la forma de cómo procesarlos. Esto último creo que fue el aporte del Máster, tener datos y buscar la forma de cómo procesarlos en función del problema y qué queremos observar o analizar.

En un principio cuando supe que esto se debía realizar en grupo, la verdad tuve una reacción negativa debido principalmente que somos de profesionales diferentes, pero luego cuando comenzamos a realizar este proyecto me di cuenta que la diferencia de profesiones es la fortaleza, por lo tanto, me quedo muy contento con el desempeño de este grupo para lograr este trabajo.

9.2.4 Gustavo Martín González

Este trabajo de fin de máster no solo me ha servido para fijar conocimientos en competencias propias de la ciencia de datos, sino que también pude visualizar la importancia y lo productivo que puede ser el trabajo multidisciplinar.

Particularmente, cuando formamos grupo y vi que los cuatro integrantes éramos de profesiones muy diferentes, tuve dudas de si podríamos llegar a ponernos de acuerdo en un tema a desarrollar y, posteriormente, armar un buen trabajo final. La realidad me ha demostrado que tener diferentes visiones realmente ha enriquecido la experiencia.

Considero que hemos logrado un ensayo de muy buen nivel, con mucho esfuerzo y coordinación, abordando un tema en donde ninguno de nosotros es especialista, sin embargo, gracias a nuestra capacidad de análisis, síntesis y a los conocimientos que hemos obtenido a lo largo del programa, estamos en condiciones de presentar un muy buen resultado final.

Particularmente me quedo con ganas de seguir estudiando y formándome en lenguajes de programación, Machine Learning, Deep Learning, etc, con lo cual, este es el primer paso a los fines de ser un científico de datos, donde seguramente, combinando mi formación en el área de las ciencias económicas, podré marcar la diferencia.

10 REFERENCIAS

Armstrong, M. B., Ferrell, J., Collmus, A., & Landers, R. (2016). *Correcting misconceptions about gamification of assessment: More than SJTs and badges. Industrial and Organizattional Psychology.*

Armstrong, M. B., Landers, R. N., & Collmus, A. B. (2017). *Gamifying recruitment, selection, training, and performance management: Game thinking in human resource management, Emerging research and trends in gamification.*

Blog de Human Development Solutions. (29 de 11 de 2011). Obtenido de https://hdsdesarrollomexico.wordpress.com/2011/11/29/testdepersonalidadmbti/#:~:text=El%20Myers%2DBriggs%20Type%20Indicator,sus%20preferencias%20personales%20m%C3%A1s%20importantes.

Briggs Myers, I. B., & McCaulley, M. H. (1962). *Manual: A guide to the development and use of the Myers-Briggs type indicator.*

Cerkez, N., Vrdoljak, B., & Skansi, S. (s.f.). *The Present Situation and the Prospect of Determining the Personality Type of Text Author with Machine Learning.*

Cui, B., & Qi, C. (s.f.). *Survey Analysis of Machine Learning Methods for Natural Language Processing for MBTI Personality Type Prediction.* Universidad de Stanford.

Deloitte University Press. (2016). *Tendencias globales en capital humano 2016 - La nueva organización: Un diseño diferente.*

Fetzer, M., McNamara, J., & Geimer, J. L. (2017). *Fetzer, M., McNamara, J., & Geimer, J. L. (2017). Gamification, serious games and personnel selection. The Wiley Blackwell handbook of the psychology of recruitment, selection and employee retention.*

Hawkes, B., Cek, I., & Handler, C. (2017). *The gamification of employee selection tools: An exploration of viabidity, utility, and future directions, Next generation technology enhanced assessment: Global perspectives on occupational and workplace testing.*

Kestenbaum, J. (2016). *Society of Humans Resources Management, Key Elements of an Effective Talent Acquisition Strategy.*

La Vanguardia. (30 de Agosto de 2017). Obtenido de https://www.lavanguardia.com/vivo/psicologia/20170830/43890425278/test-personalidad.html

Liu, G., & Ma, A. (s.f.). *Neural Networks in Predicting Myers Brigg Personality Type From Writing Style.*

Mohammad, H. A., & Kazemian, H. (2020). *Machine Learning Approach to Personality - Type Prediction Based on the Myers–Briggs Type Indicator.* Londres.

The Myers & Briggs Fundation. (s.f.). *The Myers & Briggs Fundation*. Obtenido de https://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/

Winterhalter, B. (31 de Agosto de 2014). *The Boston Globe*. Obtenido de https://www.bostonglobe.com/opinion/2014/08/30/istj-enfp-careers-hinge-dubious-personality-test/8ptUGXhu6DndFdjCngcxSN/story.html