Segmentación por personalidad utilizando minería de textos sobre la base de datos Twisty

Roy Yali, Julio Casas y Alexis León
Pontificia Universidad Católica del Perú, Lima, Perú

***Resumen*—En este artículo se presenta un estudio enfocado en la segmentación por personalidad utilizando técnicas de procesamiento de lenguaje natural y de aprendizaje automático para poder predecir el tipo de personalidad MBTI (Indicador Myers-Briggs) de usuarios de twitter pertenecientes a una muestra del conjunto de datos TWISTY conformado por tweets, particularmente en español. Nuestro objetivo de predicción ha sido los tipos de personalidad Introversión/Extroversión y Pensamiento/Sensorial.**

En esta investigación hemos realizado distintos experimentos para obtener los mejores resultados, en la vectorización hemos probado con técnicas como TF, TF-IDF, Word2vec, BoEW, y en machine learning hemos probado con algoritmos como SVM, RandomForest y LightGBM. El mejor resultado F1 obtenido para la dicotomía Introversión/Extroversión ha sido: 0.7143**, Vectorización TF-IDF y Random Forest con parametrización. Y, el mejor resultado F1 obtenido para la dicotomía Pensamiento/Sentimiento ha sido:** 0.7143**, usando Bag of Embedded Words y Random Forest con hiperparametrización.**

Finalmente, hemos tenido una etapa de validación probando con cuentas de Twitter de personajes públicos de Perú, teniendo buenos resultados.

***Palabras clave*—Aprendizaje Automático, Procesamiento de Lenguaje Natura, MBTI, Stemming, Dicotomía**

I. Introducción

La segmentación de clientes es muy utilizada en distintas industrias como la Banca, Telecomunicaciones, Seguros y Retail para poder brindar ofertas convenientes por segmentos. Usualmente, los datos utilizados para segmentar son demográficos y transaccionales, sin embargo, no se tiene en cuenta datos no estructurados para segmentar por personalidad. El presente artículo desarrolla un modelo utilizando procesamiento de lenguaje natural que clasifica el texto de entrada en tipos de personalidad. Para ello, hemos utilizado un dataset etiquetado con uno de los métodos principales para catalogar los tipos de personalidades que es el MBTI (Myers Briggs Type Indicator) [1]. Este método clasifica las personalidades según las siguientes categorías:

- ¿Cómo enfocan su atención? (Introversión/Extroversión, I/E).
- ¿Cómo perciben o toman la información? (Intuición/Sensación, N/S).
- ¿Cómo toman sus decisiones? (Pensamiento/Sentimiento, T/F).
- ¿Cómo se orientan hacia el mundo exterior? (Perceptor/Juzgador, P/J).

Estas categorías dan lugar a conjuntos de pares opuestos, a los que se les conoce como dicotomía, de las cuales se pueden formar 16 combinaciones, que indican el tipo de personalidad hacia el cual una persona tiende a inclinarse.
El presente documento presentará nuestros hallazgos en la creación del modelo y se organizará de la siguiente manera. En la Sección II se presenta el estado del arte de las técnicas referidas al análisis de personalidad. Posteriormente, el diseño del experimento y la metodología a seguir se detallan en la Sección III. En la Sección IV se detallan los experimentos y los resultados obtenidos. Se realiza una breve discusión de los resultados en la Sección V. Finalmente, las conclusiones son presentadas en la Sección VI.

II. Estado del Arte

Uno de los primeros artículos que se dedicó a la investigación en la identificación de la personalidad utilizando machine learning fue Plank (2015) [2]. Si bien MBTI fue inicialmente creado teniendo como instrumento de recolección de datos un cuestionario, Plank (2015) diseñó una metodología para aplicarlo sobre tweets y creó un dataset de tweets asignados a usuarios que fueron etiquetados según Myers Briggs, sin embargo, solo para el idioma inglés. Posteriormente, Verhoeven (2016) [3] siguió la misma metodología de Plank para etiquetar tweets en distintos idiomas entre ellos el español, este trabajo fue realizado con personal calificado de la University of Antwerp (Belgium) y University of Groningen (The Netherlands), y este será el data set que utilizaremos.
Seguidamente, Verhoeven utilizó un modelo LinearSVC para realizar la predicción de clases MBTI de los autores de los tweets. Así como este, existen varios métodos para realizar el análisis de personalidad de manera tradicional, como el estudio de árboles de dependencia, correspondencia textual, morfología, estructura de la oración, entre otros, que han sido trabajados en diversos estudios, sin embargo, las técnicas de aprendizaje supervisado han demostrado que pueden lograr muy buenos resultados al momento de identificar un tipo de personalidad bajo algun esquema como la dicotomía de los tipos MBTI [4]. En el trabajo realizado por Carlos Basto, se experimenta tanto con modelos estadísticos como Naive Bayes, como tambien con distintos modelos de aprendizaje supervizado como Random Forest, Regresión Logística, K-Nearest Neighbors, Support Vector Machine y Perceptrón Multicapa, LSTM y Bidirectional Encoder Representations from Transformers (BERT) logrando predecir las clases con

un nivel de precisión en promedio entre 85 a 86%. De manera similar. En el artículo presentado por Brandon Cui, de Stanford University, realizan la identificación de clases con 3 modelos: Naive Bayes, SVM y un modelo de Deep Learning que consiste en una LSTM que funciona como un Encoder y una red neuronal feed forward de 3 capas utilizando ReLU como función de activación, la cual sería el Decoder del modelo. Se crearon 4 clasificadores binarios con esta arquitectura para identificar individualmente cada dicotomía [5].

III. Diseño del Experimento

A. Descripción del conjunto de datos

En este proyecto se trabaja con el dataset Twisty [3], que contiene información de personalidad segun la categoría MBTI para un total de 18168 usuarios en diferentes idiomas (Holandes, Aleman, Español, Frances, Italiano y Portugues) con un promedio de 2000 tweets por usuario, distribuidos como se muestra en la Tabla I:

De este dataset se ha elegido el idioma español que son 18 millones de 10 772 usuarios, para el experimento se descargaron 82 mil tweets de 79 usuarios seleccionando muestras de forma estratificada a las clases que se van a predecir. Se eligieron las clases de representan las personalidades de acuerdo a las características de personalidades basadas en el enfoque de atención (Introvertido/Extrovertido) y cómo toman sus decisiones (Pensamiento / Sentimiento), esto debido a que en investigaciones previas se había evidenciado que las demás dicotomías no podían ser bien representadas por un aprendizaje de machine learning.

Idioma	Usuarios	Tweets	Promedio	idioma
Aleman	411	952,549	2,318	74.9
Italiano	490	932,785	1,904	70.6
Holandes	1000	2,083,484	2,083	74.0
Frances	1405	2,786,589	1,983	71.6
Portuges	4090	8,833,132	2,160	71.9
Español	10772	18,547,622	1,722	72.8

TABLA I: Descripción del conjunto de datos

B. Preprocesamiento de datos

1) ***Descarga de bases de datos****:* Esta base de datos se ha separado de tal forma que se tenga cada clase de manera balanceada, obteniendo 25 usuarios para cada una de las dicotomías encontradas, las cuales han sido descargadas por medio del api de Twitter, obteniendo finalmente un total de 79 usuarios y un total de 82468 tweets, los cuales están separados en confirmados y otros, los primeros han sido confirmados del idioma español y utilizados en el modelo Twisty [3] para identificar la personalidad de las personas, y los demás son otros tweets del mismo usuario sin confirmar el idioma.

Una vez obtenida la base de datos de los tweets de cada usuario, se ha reordenado la información para que pueda ser limpiada y procesada por separado.

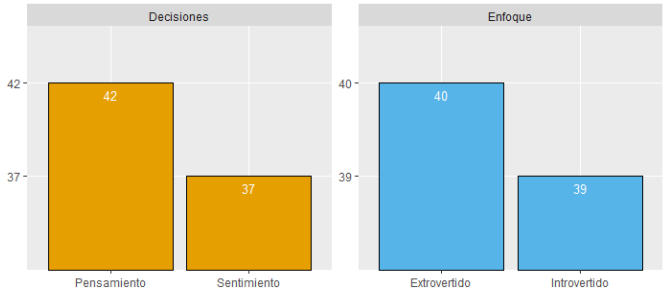

Fig. 1: Cantidad de usuarios por característica de personalidad

2) ***Limpieza de datos****:* Para el procesamiento de datos textuales se han realizado las siguientes acciones:

- Convertir texto a minúsculas, lo que permite estandarizar una codificación única cuando las palabras se repitan.
- Eliminar citas y etiquetas para evitar palabras que sean poco relevantes para analizar las emociones de los usuarios. Esta técnica es válida para encontrar tópicos, sin embargo tiene limitaciones para clasificar sentimientos.
- Eliminar signos de puntuación, símbolos, caracteres especiales.
- Eliminar dígitos debido a que son poco relevantes en el análisis de sentimientos.
- Procedimiento de lematización, permite llevar las palabras a su estado base sin perder información de la misma.
- Eliminar ”stop words”: Los stopwords que se han empleado ha recogido una base existente de stopwords en español para análisis de sentimiento, además se ha recogido la base de stopwords de NLTK y cada letra del abecedario por separado.

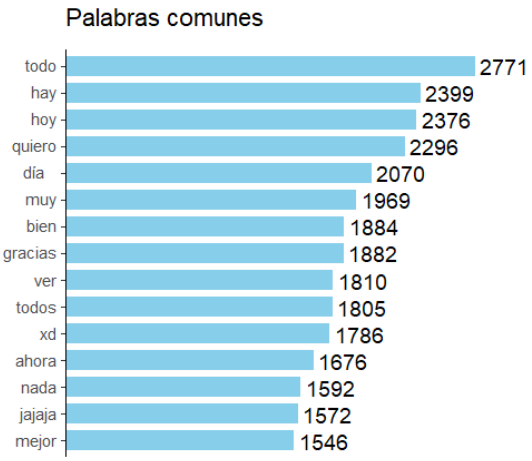

Fig. 2: Palabras más comunes del dataset

Las palabras más comunes del dataset luego de haber realizado la limpieza de datos se muestran en la Figura 2 .

*3) **Creación del documento**:* Cabe resaltar que el clasificador utilizado funciona a nivel de usuario, debido a ello el documento consistió en la concatenación de todos los tweets confirmados de ese usuario.

C. Vectorización de documentos

En esta etapa se han representado los textos en vectores característicos para posteriormente aplicar algoritmos de machine learning. Se aplicaron 4 tipos de vectorizadores que luego empleando distintos algoritmos de machine learning han servido para clasificar las dicotomías de sentimientos.

*1) **Frecuencia de términos - TF (CountVectorizer)**:* TF asigna una puntuación de la frecuencia de la palabra en el documento. Aquí se ha empleado la función CountVectorizer de la librería SciKit-Learn [6]. Esta función cuenta el número de veces que una palabra aparece en un documento dividida por el total de números de palabras en el documento.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

*2) **Frecuencia inversa de documentos - TF-IDF (TfidfVectorizer)**:* Es una puntuación de cuán representativa es la palabra en los documentos. Con TFIDF se puede determinar el peso de las palabras sobre todos los documentos en el corpus.

$$idf(w) = log(\frac{N}{df_t})$$

$$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

Se interpreta que la puntuación cuanto más cerca de 1, más informativo es ese término para ese documento. Cuanto más cerca de cero, menos informativo será ese término. Aquí hemos empleado la función TfidfVectorizer de la librería SciKit-Learn [6].

*3) **Word2vec**:* Este modelo se utiliza para representaciones de palabras en el espacio vectorial, es un modelo de red neuronal que intenta explicar las representaciones de palabras basándose en un corpus de texto. Esto implica que para aprender la representación, se busca en palabras cercanas; si un grupo de palabras siempre se encuentra cerca de las mismas palabras, terminarán teniendo representaciones similares. Aquí hemos empleado la función Word2Vec de la librería Gensim [7].

El objetivo de este modelo es maximizar la probabilidad logarítmica promedio donde $w_1, w_2, \ldots, w_T$.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=-k}^{j=k} logp(w_{t+j}|w_t)$$

Donde K es el tamaño de la ventana de entrenamiento.

Como resultante se obtiene un vector característico por cada palabra, para tener la representación del documento se utilizó el promedio de los vectores de cada palabra que lo conforman.

*4) **Bag of Embedded Words - BOEW**:* Este modelo incorpora un área de ponderación que permite alterar la importancia de cada palabra aprendida. Además, propone una función objetivo de entropía esférica para optimizar la representación aprendida para la recuperación utilizando la métrica de similitud de coseno o distancia euclidiana. [8] Este método da la posibilidad de encontrar vectores más representativos para los documentos, haciendo que en lugar del promedio usado en el paso anterior, se use un vector que indica la frecuencia en la que una representación se encuentra en un cluster (o tópico) asignado.

D. Experimentos con Machine Learning

Una vez obtenidas las distintas representaciones vectoriales de los tweets con distintas técnicas, se realizaron experimentos con algoritmos de Machine Learning, para lo cual, se utilizaron 2 modelos, el primero para predecir la dicotomía Introversión/Extroversión y el segundo para predecir la dicotomía Pensamiento/Sentimiento. El dataset fue dividido en entrenamiento y pruebas en porcentajes 80 y 20. Seguidamente se entrenaron 3 algoritmos por cada modelo, Support Vector Machine [9], LightGBM (Boosting) [10] y Random Forest (Bagging) [11] utilizando cada una de las 4 representaciones vectorizadas obtenidas en el paso anterior. Cabe resaltar que los experimentos realizados se hicieron en dos fases, la primera con los algoritmos con parámetros por defecto y la segunda, utilizando validación cruzada e hiperpametrización. La métrica empleada para identificar la eficiencia de los modelos fue el f-score, que según la bibliografía revisada [12] se adecúa bien en problemas con clasificación binaria.

E. Validación con casos reales

Finalmente, con los mejores modelos obtenidos, se realizaron pruebas con cuentas de usuarios reales de personajes peruanos para validar si se presentaban resultados predecibles.

IV. Experimentación y resultados

La experimentación se realizó de forma secuencial de acuerdo a los modelos mencionados. Se realiza un preprocesamiento del conjunto de datos, luego la vectorización de cada tweet procesado; en el caso de TF y TF-IDF se utilizaon los vectorizadores provistos por la librería SkiKit-Learn, para Word2Vec se obtiene el vector promedio de todas las palabras del tweet y para BoEW se calcula un vector con la frecuencia de cada cluster calculado mediante la aplicación del algoritmo k-means. Para las 4 técnicas se utilizó un vector de 100 características.

Una vez obtenidos los vectores, se aplicaron los modelos de machine learning escogidos. Se realizó un entrenamiento de cada modelo para cada una de las 2 dicotomías MBTI escogidas, es decir, un entrenamiento para poder clasificar Introversión vs Extroversión (I/E) y otro para Pensamiento vs Sensorial (T/F). Se utilizó como métrica de evaluación Valor-F o F-score y los resultados obtenidos se visualizan en la tabla II

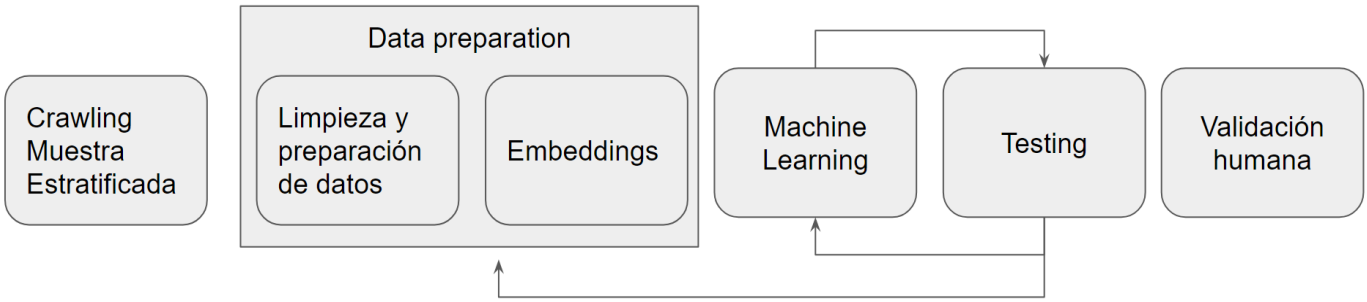

Fig. 3: Flujo de experimento.

Vectorización	MBTI	Modelo de Machine Learning		
		SVM	Random Forest	LightGBM
TF	**I/E**	0.5882	0.4615	0.5714
	T/F	0.5333	0.4615	0.5000
TF-IDF	**I/E**	0.5556	**0.7143**	0.5333
	T/F	**0.7143**	0.4615	0.5333
W2V	**I/E**	0.5263	0.4286	0.2857
	T/F	0.0000	0.4615	0.4615
BoEW	**I/E**	0.5556	0.6250	0.5333
	T/F	**0.7143**	0.3636	**0.6154**

TABLA II: Resultados modelos I/E y T/F con parámetros por defecto

Luego de los primeros entrenamientos, se realizó una optimización de hiperparámetros mediante una búsqueda con GridSearchCV para poder encontrar la mejor configuración dentro de todas las posibles combinaciones. Dentro de la grilla de parámetros de cada modelo se incluyeron los parámetros por defecto de cada uno de ellos y se utilizó 5-fold cross-validation. Los resultados se muestran en la Tabla III.

Vectorización	MBTI	Modelo de Machine Learning		
		SVM	Random Forest	LightGBM
TF	**I/E**	0.5882	0.4286	0.5333
	T/F	0.3333	0.4615	0.4615
TF-IDF	**I/E**	0.5556	0.6250	0.4286
	T/F	0.0000	0.2857	0.3333
W2V	**I/E**	0.5263	0.5333	0.2857
	T/F	0.5000	0.5000	0.4615
BoEW	**I/E**	0.5556	0.5000	**0.6250**
	T/F	**0.6667**	**0.7143**	0.5333

TABLA III: Resultados modelos I/E y T/F con hiperparametrización

De todos los entrenamientos anteriores, los mejores resultados fueron obtenidos con Vectorización TF-IDF y Random Forest con parametrización por defecto como modelo de Machine Learning para I/E (0.7143) y para T/F la vectorización con Bag of Embedded Words y Random Forest con hiperparametrización dieron el mejor puntaje F1 (0.7143). La similitud de los valores obtenidos se debe al tamaño del conjunto de datos, debido a que los tweets se agruparon por usuarios, se cuenta con un conjunto de datos de 79 usuarios, en donde todos sus tweets fueron concatenados. De esta manera, el conjunto de pruebas se conforma con el 20% del total, es decir, 16 usuarios.

Para evaluar el rendimiento de los mejores modelos obtenidos se realizó una predicción de las dicotomías MBTI de usuarios de personas famosas a través de sus tweets. Se concatenaron 100 tweets por cada usuario y se aplicó el mismo pipeline del preprocesamiento del conjunto de datos. Los usuarios evaluados fueron: el músico peruano Gian Marco (@gianmarcomusica), Mariano Morikawa (@morikawaphd), quien es un científico de ascendencia peruano-japonesa especializado en temas de descontaminación utilizando biotecnología y nanotecnología, Modesto Montoya (@modestomontoya), renombrado físico nuclear peruano y actual asesor presidencial en materia científica y María Antonieta Alva (@tonialval), ex ministra de Economía y Finanzas.

Nombre	Usuario Twitter	Dicotomia	
		I/E	T/F
Gian Marco	@gianmarcomusica	Extroversión	Sensorial
Mariano Morikawa	@morikawaphd	Introversión	Sensorial
Modesto Montoya	@modestomontoya	Introversión	Sensorial
María A. Alva	@tonialval	Introversión	Pensamiento

TABLA IV: Resultados luego de la optimización de hiperparámetros con GridSearch

Como se puede apreciar en la tabla IV, Gian Marco obtuvo como dicotomía I/E: Extroversión y como T/F: Sensorial, que encaja con el perfil de un artista, mientras que los demás usuarios, quienes tienen un perfil académico, obtuvieron la dicotomía I/E de Introversión. Por otro lado, ambos científicos obtuvieron una dicotomía T/F de Sensorial, a diferencia de María Antonieta Alva, quien obtuvo Pensamiento. Para poder interpretar esta diferencia, se analizará un extracto de los tweets observados en las imágenes 1 y 2

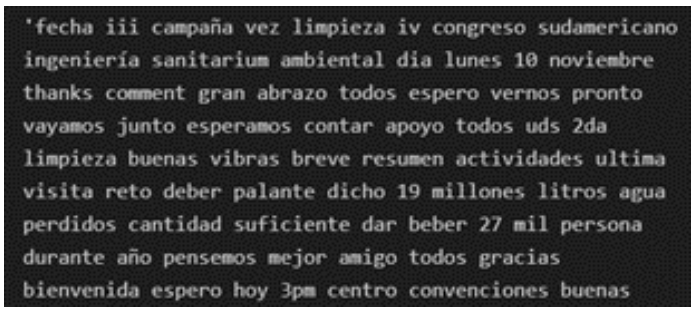

Fig. 4: Extracto de los tweets de Mariano Morikawa

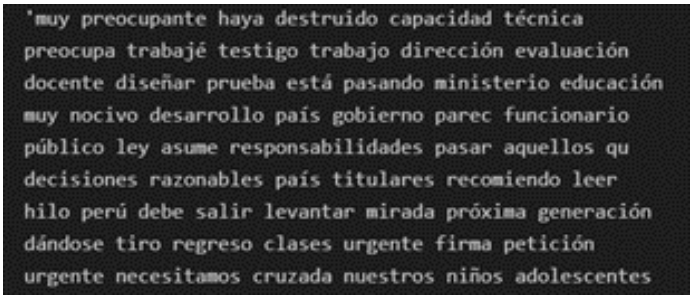

Fig. 5: Extracto de los tweets de María Antonieta Alva

En los tweets de la Mariano Morikawa se puede apreciar una mayor interacción con la audiencia con palabras y frases como “gracias”, “abrazo”, “espero vernos pronto”, que caracteriza a la dicotomía Sensorial, que basa sus decisiones en función a como sus acciones afectan a los demás, a diferencia de los tweets de María Antonieta Alva, que tienen una naturaleza más informativa.

V. Discusion

Las métricas F1 obtenidas en los experimentos tanto con parámetros fijos como en la optimización con GridSearch muestran que los resultados utilizando una vectorización TF-IDF fueron más precisos comparados a los de la vectorización TF.

Dado que un tweet suele tener una naturaleza más informal que un trabajo académico, debido a que es una red social, TF-IDF reduce el peso de las palabras comunes, dando énfasis a palabras únicas en los tweets, lo cual se traduce en mejores resultados. De manera similar Bag of Embedded Words superó a Word2Vec, dado que es una mejora de dicha técnica. El vector Word2Vec se calculó utilizando los promedios de los valores asignados a cada palabra a diferencia de BoEW que toma en cuenta la frecuencia de las palabras por clases asignadas mediante k-means, dándole un énfasis al dominio del contenido de los tweets.

Por otro lado, una limitación fue la cantidad de tweets que se pudieron obtener para el experimento. El trabajo de línea base utilizó un concatenado de 200 tweets por usuario para un total de 10000 usuarios en español. Debido a las limitaciones del máximo de consultas que permite el API de Twitter (tiempos de espera de aproximadamente 10 minutos para obtener alrededor de 500 tweets), se obtuvieron un total de 82468 tweets de 79 usuarios luego del filtrado de tweets inválidos y vacíos. Se compensó la poca cantidad de usuarios concatenando todos los tweets posibles por cada usuario para agregar la mayor cantidad de información posible a los modelos. Esta limitación también generó que los resultados sean números similares, debido a que el conjunto de prueba está compuesto por un 20% del total de usuarios.

De manera similar al trabajo base, se realizó la clasificación para 2 de las 4 dicotomías existentes I/E y T/F, por lo que cada uno de los modelos buscará encontrar las diferencias entre estos tipos de personalidades. Las figuras 3 y 4 muestran un wordcloud de las palabras más comunes en los tweets de los usuarios que fueron previamente etiquetados con cada una de las dicotomías.

Fig. 6: Palabras más representativas para la dicotomía I/E

Fig. 7: Palabras más representativas para la dicotomía T/F

Si bien hay palabras en común entre cada uno de los 2 grupos, esto se debe a que son palabras que no estarían asociadas a una personalidad en particular, las palabras únicas en cada grupo son las que aportan información a los modelos, por ejemplo, en la figura 3, dentro del conjunto de palabras para Introversión destacan “casa”, “vida”, a diferencia del grupo Extroversión, en donde “feliz”, “bueno”, “mall” no se ubican entre las palabras más comunes del otro grupo. De manera adicional, en el grupo Extroversión hay más palabras que indican un tipo de emoción, como distintos tipos de risas (“jajaja”, “jaja”, “xd”). Con respecto a la segunda dicotomía, no se encontraron palabras que caractericen a una en particular, ya que la mayoría se repite en ambos grupos, en este caso, los clasificadores, principalmente los que utilizaron BoEW se basaron en el contexto de cada tweet, definido por las clases encontradas mediante k-means para poder hacer la distinción de la personalidad de esta dicotomía.

VI. Conclusiones

Los resultados obtenidos evidenciar que si es posible la segmentación en base a la personalidad y podría ser utilizado para estrategia de segmentación de clientes y así poder brindar ofertas personalizables no solo por la condición sociodemográfica, transaccional e historial crediticio, si no también por la personalidad.

Por otro lado, debido a que nuestro modelo tiene como input un documento (hemos probado con 100 a 200 tweets

concatenados), podría extenderse las pruebas a otras fuentes de los clientes como por ejemplo, chats y transcripciones de audio.

En cuanto a la evaluación de métricas del modelo, el mejor resultado F1 obtenido para la dicotomía Introversión/Extroversión ha sido: 0.7143, Vectorización TF-IDF y Random Forest con parametrización por defecto como modelo de Machine Learning.

Y así mismo, el mejor resultado F1 obtenido para la dicotomía Pensamiento/Sentimiento ha sido: 0.7143, Bag of Embedded Words y Random Forest con hiperparametrización.

Acerca de las representaciones, los vectorizadores TF-IDF y BoEW fueron los que obtuvieron mejores resultados, por un lado porque representan mejor las palabras más importantes y por otro lado, ya que en su proceso incluye la segmentación de distintos tópicos, y esto resulta ser importante para la discriminación de la clase binaria.

Cabe mencionar también que el vectorizador BoEW en todas las combinaciones ha obtenido una métrica superior a 0.5, por lo que se concluye que, en general, caracteriza mejor las dicotomías de personalidad.

También, los resultados obtenidos en la evaluación externa con figuras públicas de Perú fueron buenos, las personalidades obtenidas fueron en su mayoría las esperadas en las pruebas con los dos modelos.

En las oportunidades de mejora, consideramos que se debe trabajar en incluir más usuarios en el dataset.

Inclusive, construir un dataset con usuarios peruanos y etiquetarlos también podría ser un buen ejercicio por particularidades del lenguaje y contexto cultural que pueda afectar la predicción de la personalidad.

Y finalmente, se podrían realizar más experimentos con distintos tamaños para los vectorizadores e intentar llegar a una mejor representación.

REFERENCIAS

[1] P. Briggs Myers, I. Myers, *Gifts Differing: Understanding Personality Type*, 2nd ed. Nicholas Brealey, 2010.

[2] B. Plank and D. Hovy, "Personality traits on twitter—or—how to get 1,500 personality tests in a week," 2015.

[3] D. W. . P. B. Verhoeven, B, "TWISTY: a Multilingual Twitter Stylometry Corpus for Gender and Personality Profiling," *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 1632–1637, 05 2016.

[4] C. Basto, "Extending the Abstraction of Personality Types based on MBTI with Machine Learning and Natural Language Processing," *arXiv preprint arXiv:2105.11798*, 2021.

[5] . Q. C. Cui, B, "Survey analysis of machine learning methods for natural language processing for MBTI Personality Type Prediction," 2017.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[8] N. Passalis and A. Tefas, "Learning bag-of-embedded-words representations for textual information retrieval," *Pattern Recognition*, vol. 81, pp. 254–267, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320318301420

[9] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. New York, NY, USA: Association for Computing Machinery, 1992, p. 144–152. [Online]. Available: https://doi.org/10.1145/130385.130401

[10] Microsoft. Lightgbm documentation. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/index.html

[11] Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, 1995, pp. 278–282 vol.1.

[12] Y. Sasaki, "The truth of the f-measure," *Teach Tutor Mater*, 01 2007.