INTRODUCCIÒN A LA INTELIGENCIA ARTIFICIAL PARA CIENCIAS E INGENIERÌAS

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Análisis de rasgos de personalidad mediante comportamientos específicos

RESPONSABLES:

ANGIE VIVIANA ROSALES PORTILLA C.C 1037661697
KEVIN ALONSO RESTREPO GARCÌA C.C 1216726638
DANIEL ALEXANDER BASTO MORENO C.C 1030639051

PROFESOR:
RAÙL RAMOS POLLÀN

FACULTAD DE INGENIERÍA. INGENIERÍA DE SISTEMAS.
UNIVERSIDAD DE ANTIOQUIA
MEDELLÍN, COLOMBIA

***Resumen*— En los últimos años, el poder determinar *Rasgos* de la *Personalidad* de una persona ha generado mucho interés, no solo en personas enfocadas en la Psicologia o Psiquiatria, sino también en diversas áreas (como en la ingeniería de sistemas o ciencias de la computación). Esto se debe al creciente uso de medios virtuales para la realización de pruebas con el fin de establecer *Rasgos* de *Personalidad* que se usarán en diferentes ámbitos, como aplicar a un trabajo, determinar intereses, principales actitudes, entre otros.**

Con el fin de determinar de una mejor manera estos *Rasgos,* nos propusimos usar el *Modelo de cinco factores (FFM)* y el *Modelo oceánico* los cuales son una taxonomía de la *Personalidad,* además usamos *Redes neuronales* para determinar así qué tipo de personalidad tiene cada persona y esto cómo se relaciona con algunos de sus gustos. Para ello pensamos hacer uso de Python así como de una base de datos pública en internet con el fin que todo sea accedido de manera gratuita y así permitir a personas foráneas hacer uso de ello en un futuro.

Palabras clave*: *Modelo de cinco factores (FFM), Modelo oceánico, Personalidad, Rasgos, Redes neuronales, Perceptrón Multicapa.

I. INTRODUCCIÓN

En años recientes, el crecimiento en demanda e investigación de la inteligencia artificial ha generado que con ella se estudien muchos ámbitos que anteriormente no se podían, esto gracias a su capacidad de "aprender" y a sus grandes posibilidades a futuro, entre las cuales se encuentra el desarrollo de una IA independiente. Por tal motivo, un área de estudio que ha interesado mucho tanto a informáticos como a personal de la salud, es como describir rasgos de la personalidad.

En este proyecto se quiere hacer uso de redes neuronales con el fin de entrenarlas para que puedan categorizar los rasgos y así designar los factores de personalidad pertinente a las personas, además, de poder determinar actitudes y gustos. Existe una base de datos con más de un millón de datos que puede resultar muy útil para dicha tarea.

Es de suma importancia resaltar, que este tipo de análisis y trabajo se puede aplicar a varios ámbitos, entre los cuales están los conocidos test de personalidad, pruebas psicotécnicas y de mercadeo, resultando de gran importancia para las empresas, ya que puede ayudar a decidir que tipo de persona necesita para un determinado empleo (en caso de hacer uso de una prueba psicotécnica) o ayuda a saber qué forma de mercadeo o publicidad se debe usar para llegar a determinado grupo de personas.

Para ello haremos aplicaremos ingeniería de características, analizaremos si las columnas son relevantes, reduciremos la de dimensionalidad de la base de datos, agrupamos los datos permitiendo obtener las clases y por último clasificaremos los datos. Los resultados de este trabajo nos servirán para aprender sobre redes neuronales, métodos y/o técnicas para manejo de información, rasgos y personalidad, además, de ayudarnos como una guía o punto de referencia para trabajos futuros.

II. PLANTEAMIENTO DEL PROBLEMA

Para algunas empresas así como varias instituciones, saber qué tipo de personalidad tiene una persona antes de contratarla, qué gustos tiene un cliente o qué actitudes puede enseñar una persona es de suma importancia. El saber los rasgos de la personalidad, la actitud o gusto de una persona puede ayudar a generar un mejor entendimiento, contratar la mejor persona para un puesto o tener mejor publicidad y demás para un producto. Por ello, tener algoritmos o métodos que ayuden a descubrir la personalidad de una persona han sido una meta para muchas personas.

El problema con esto surge cuando no hay un buen algoritmo o método para realizarlo o la fiabilidad puede no ser la mejor, por ello se busca continuamente las mejores herramientas para ello. El realizar encuestas (en el caso del mercadeo) o entrevistas (sean personales o de manera virtual para el caso de entrevistas), puede llegar a ser costoso en tiempo y dinero. Además, con el continuo desarrollo de la tecnología, el uso constante de redes sociales e internet por parte de las personas y la necesidad de estar a la vanguardia para no quedarse atrás en el mercado hace que sea indispensable solucionar dicha problemática. Es conveniente entonces, crear una manera (en su medida confiable y rápida) que realice este análisis utilizando el estado del arte en el procesamiento de la información respecto a la personalidad, intentando garantizar con rapidez, resultados pertinentes, concisos y con altos niveles de confianza, como aquellos que entregan algunos modelos de aprendizaje automático y sistemas expertos.

III. METODOLOGÍA Y DESCRIPCIÓN DEL MODELO

Para abordar el proyecto primero que todo se debe sintetizar la base de datos, procediendo de la siguiente manera: aplicar ingeniería de

características con el fin de verificar si hay datos faltantes, realizar limpieza de datos, luego proceder a un escalamiento de datos, finalizando con la detección y eliminación de outliers. Posteriormente debemos analizar si las columnas son relevantes, para esto haremos una reducción de dimensionalidad, verificando si todas las columnas aportan información relevante o si son datos superfluos como es el caso de las fechas, que se verifican para ver si se pueden descartar o si por el contrarío influyen en la clasificación de los datos.
Seguidamente se procede con la creación de dos modelos, uno con reducción y otro sin la reducción, esto para verificar si mejora o desmejora el modelo. Después, agrupamos los datos para poder obtener las clases. Además, usaremos el método de agrupamiento Minibatch K-means para la clasificación de las tuplas, bien conocido como 'clustering'.
Paso siguiente realizamos la clasificación de los datos: tomaremos un 70% para la creación y entrenamiento del modelo, un 10% para validación y para la prueba un 20% (tuplas sin etiquetar o clasificar). Para el manejo de los datos usaremos de pandas en python, y para el proceso de clasificación de los datos haremos uso de la librería scikit-learn.

Para la creación del modelo se implementó una red neuronal MLP (Perceptrón multicapa), haciendo uso de la librería sklearn.neural_network debido a que en esta es donde se encuentran dichas redes neuronales. Luego, se define un batch de tamaño 64 para hacer uso de los optimizadores estocásticos, la red neuronal se diseño con 3 capas ocultas y cada una de estas capas con 50 neuronas. Para el número de interacciones, será un máximo de 1000 iteraciones para obtener la convergencia, o si es el caso llegar a una aproximación.

También se envio como parametro de activación "logistic" que equivale a una función sigmoide logística, para procesar los datos aplicando una función, esto se usa para que la red neuronal haga su proceso de "aprendizaje". El parámetro de "solver" se envía como "sgd" (Stochastic Gradient Descent) que se refiere al descenso de gradiente estocástico para la optimización de pesos, esto también usado para la enseñanza y optimización de la red neuronal. Luego, se le determina el 10 como número de partida para la generación de números aleatorios en la inicialización de pesos y sesgos que serán usados por la red neuronal.

Al momento de crear el modelo de aprendizaje, se decidió crear dos redes neuronales ya que debido al alto volumen de datos, el clustering dio como resultado que dos tamaños de batch tienen un buen comportamiento, por lo tanto, se decidió hacer uso de ambos ("Minibatch-Kmeans" batch_size : 32,64) y en base a estos se crean los dos modelos MLP.

Por último, se entrena con el 70% de los datos correspondientes al clustering hecho, luego de esto se hace uso del 10% para validar su correcto funcionamiento y por último se usa el 20% restante para probar su funcionamiento. En caso de que no pase las pruebas, se debe verificar cuál es el problema:

- Necesidad de más datos.
- Verificar los parámetros de entrada.
- Aumentar o disminuir más capaz.
- Aumentar o disminuir el número de neuronas por capa.

IV. ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS

Después de haber aplicado la metodología descrita con anterioridad y haber entrenado hecho el entrenamiento de las redes neuronales (RN), se realiza las pruebas con el 20% de los datos y dio como resultado:

Red neuronal de batch 32:

RN MLP (con Clustering de Batch 32) presenta una coincidencia entre datos actuales y la predicción del 94.22708418306487 %, considerando esto como un buen resultado. En la figura 1, se aprecia el porcentaje de precisión de predecir cada personalidad, el porcentaje de recuperación, el porcentaje de puntaje f1 y la cantidad de datos que hay en cada tipo de personalidad.

	precision	recall	f1-score	support
0	0.78	0.20	0.32	5811
1	0.95	0.99	0.97	241488
2	0.00	0.00	0.00	123
3	0.80	0.80	0.80	1386
4	0.64	0.21	0.32	11078
accuracy			0.94	259886
macro avg	0.63	0.44	0.48	259886
weighted avg	0.93	0.94	0.93	259886

Fig. 1. Resultados arrojados por la RN

También, se verifica el valor de pérdida de pesos en el aprendizaje (figura 3), lo cual implica que tan bien o mal se comporta el modelo en términos optimización de después de cada iteración. Idealmente, se espera que la reducción de la pérdida de peso se de después de cada iteración o

varias iteraciones. Se aprecia que al inicio del aprendizaje hay un peso elevado, pero a medida que el aprendizaje se desarrolla este peso va bajando.

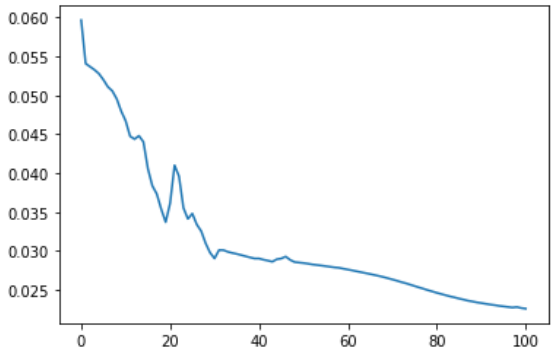

Fig. 2. Gráfica del valor de pérdida

Luego, la matriz de confusión (figura 3) nos muestra cómo se agrupan los datos dependiendo del tipo de personalidad. Se puede apreciar que con el batch de tamaño 32 la mayoría de los datos se agrupan o se clasifican como personalidad del tipo 1.

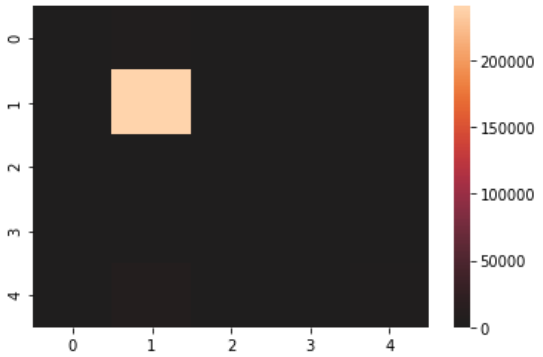

Fig. 3. Matriz de confusión

Por último, usamos una gráfica de distribución (figura 4) para visualizar el resultado de la clasificación. Se puede notar que con el que con el batch de tamaño 32 la mayoría de los datos se agrupan o se clasifican como personalidad del tipo 1 (Responsable) y el tipo 2 (Extrovertido) queda sin datos.

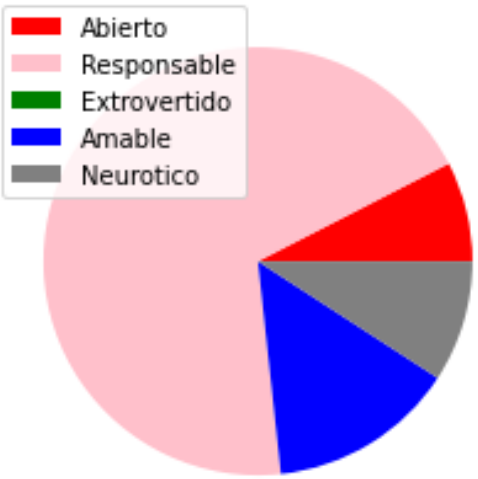

Fig. 4. Gráfica de distribución

Red neuronal de batch 64:

RN MLP (con Clustering de Batch 64) presenta una coincidencia entre datos actuales y la predicción del 99.47400013852229 %, considerando esto como un muy buen resultado. En la figura 5, se aprecia el porcentaje de precesión de predecir cada personalidad, el porcentaje de recuperación, el porcentaje de puntaje y la cantidad de datos que hay en cada tipo de personalidad.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	226
1	0.47	0.48	0.47	595
2	0.95	0.61	0.74	866
3	0.00	0.00	0.00	372
4	1.00	1.00	1.00	257827
accuracy			0.99	259886
macro avg	0.48	0.42	0.44	259886
weighted avg	0.99	0.99	0.99	259886

Fig. 5. Resultados arrojados por la RN

También, se verifica el valor de pérdida de pesos en el aprendizaje(figura 6), lo cual implica que tan bien o mal se comporta cierto modelo después de cada iteración de optimización. Idealmente, se espera que la reducción de la pérdida de peso se de después de cada iteración o varias iteraciones. Se aprecia que al inicio del aprendizaje hay un peso elevado, pero a medida que el aprendizaje se desarrolla este peso va bajando.

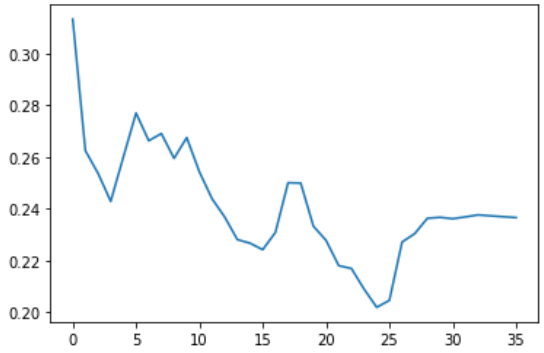

Fig. 7.Fig. 2. Gráfica del valor de pérdida

Luego, la matriz de confusión (figura 7) nos muestra cómo se agrupan los datos dependiendo del tipo de personalidad. Se puede apreciar que con el batch de tamaño 64 la mayoría de los datos se agrupan o se clasifican como personalidad del tipo 4.

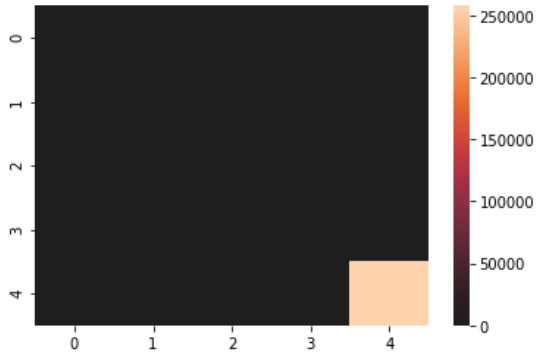

Fig. 7. Matriz de confusión

Por último, usamos una gráfica de distribución (figura 8) para visualizar el resultado de la clasificación. Se puede notar que con el que con el batch de tamaño 32 la mayoría de los datos se agrupan o se clasifican como personalidad del tipo 4 (neurótico) y el tipo 3 (Amable) queda sin datos.

Fig. 8. Gráfica de distribución

De estos resultados (tanto de la RN de batch de 32 como del de 64) podemos decir que ambas redes tienen un buen desempeño, teniendo un porcentaje de aceptación mayor al 93%, por lo tanto consideramos que ambas redes funcionan bien. Cabe resaltar que surge la duda del por qué en ambas redes se pierde un tipo de personalidad o los datos en este son muy bajos, por lo cual, en trabajos futuros se plantea verificar si el problema se presenta por el batch (de la manera en que fue creado), por el tamaño del batch o por alguno de los parámetros de entrada en la RN que se dejaron por defecto.

V. CONCLUSIONES

La realización de este trabajo nos permitió comprender sobre redes neuronales y cómo éstas llegan a ser muy efectivas a la hora de aplicarlas, además de enfatizar en las simulaciones y en el uso de librerías de código abierto que se encuentran disponibles para el lenguaje de programación Python.

Nos permitió conocer herramientas como Minibatch-Kmeans, debido a que durante el proceso de la construcción presentamos dificultades por el grandes volúmenes de datos (1’115,000 filas y 110 columnas en nuestro caso) y por ello se optó por hacer uso de Minibatch-Kmeans, ya que el Kmeans normal pierde eficiencia y puede llevar a realizar clasificaciones erróneas.

Por último, concluimos que este tipo de desarrollos pueden ayudar en gran medida a empresas o negocios, como por ejemplo:

- A la hora de construir trabajos en equipo, verificando la compatibilidad entre los diferentes integrantes.
- Ayudando al mejoramiento de métodos para llamar la atención del posible usuario.
- Brindar soporte a la hora de la contratación de personal.

VI. REFERENCIAS

[1]"Modelo de los cinco grandes - Big Five personality traits - qwe.wiki", Es.qwe.wiki, 2020. [Online]. Available: https://es.qwe.wiki/wiki/Big_Five_personality_traits. [Accessed: 28- May- 2020].

[2]"Análisis factorial - Factor analysis - qwe.wiki", *Es.qwe.wiki*, 2020. [Online]. Available: https://es.qwe.wiki/wiki/Factor_analysis. [Accessed: 28- May- 2020].

[3]"sklearn.neural_network.MLPClassifier — scikit-learn 0.23.1 documentation", *Scikit-learn.org*, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html. [Accessed: 28- May- 2020].